

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ**

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Обґрунтування вибору апаратного та програмного  
забезпечення для побудови системи вивчення технологій з веб-  
розробки»

на здобуття освітнього ступеня магістра  
зі спеціальності 123 Комп'ютерна інженерія  
(код, найменування спеціальності)  
освітньо-професійної програми Комп'ютерні системи та мережі  
(назва)

*Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело*

\_\_\_\_\_ (підпис)

Микита КУЦЕНКО  
Ім'я, ПРІЗВИЩЕ здобувача

Виконав:  
здобувач вищої освіти  
група КСДМ-62

Микита КУЦЕНКО

Керівник:  
науковий ступінь,  
вчене звання

Ярослав ТОРОШАНКО  
К.Т.Н., С.Н.С.

Рецензент:  
науковий ступінь,  
вчене звання

\_\_\_\_\_ Ім'я, ПРІЗВИЩЕ

**Київ 2023**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Комп'ютерної інженерії  
Ступінь вищої освіти Магістр  
Спеціальність 123 Комп'ютерна інженерія  
Освітньо-професійна програма Комп'ютерні системи та мережі

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри Комп'ютерної інженерії  
\_\_\_\_\_ Наталія ЛАЩЕВСЬКА  
« \_\_\_\_\_ » \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Куценко Микиті Олександровичу

1. Тема кваліфікаційної роботи: «Обґрунтування вибору апаратного та програмного забезпечення для побудови системи вивчення технологій з веб-розробки», керівник кваліфікаційної роботи Ярослав ТОРОШАНКО, к.т.н., доцент, затверджені наказом Державного університету інформаційно-комунікаційних технологій від 19.10.2023 р. №145
2. Строк подання кваліфікаційної роботи 26 грудня 2023 р.
3. Вихідні дані до кваліфікаційної роботи: підручники, навчальні посібники, статті та тези конференцій вітчизняних і зарубіжних авторів, Інтернет-ресурси з питань рекомендаційних систем, тестування, Item Response Theory та технологій з веб-розробки.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
  - 1) Теоретичні основи та методичні підходи до системи вивчення технологій веб-розробки;
  - 2) Аналіз та оцінка ефективності системи;
  - 3) Визначення оптимального апаратного та програмного забезпечення для системи вивчення технологій веб-розробки.
5. Перелік графічного матеріалу: *презентація.*
6. Дата видачі завдання 10 вересня 2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір технічної літератури	10.09.2023-24.09.2023	Виконано
2	Теоретичні основи та методичні підходи до системи вивчення технологій веб-розробки	25.09.2023-15.10.2023	Виконано
3	Аналіз та оцінка ефективності системи	16.10.2023-05.11.2023	Виконано
4	Визначення оптимального апаратного та програмного забезпечення для системи вивчення технологій веб-розробки	06.11.2023-19.11.2023	Виконано
5	Оформлення роботи, висновки	20.11.2023-17.12.2023	Виконано
6	Розробка демонстраційного матеріалу, доповідь	18.12.2023-25.12.2023	Виконано
7	Отримання рецензії		
8	Затвердження пояснювальної записки роботи завідувачем кафедри		
9	Захист кваліфікаційної роботи		

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Микита КУЦЕНКО

(Ім'я, ПРІЗВИЩЕ)

Керівник  
кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Ярослав ТОРОШАНКО

(Ім'я, ПРІЗВИЩЕ)

## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра:  
115 стор., 34 рис., 5 табл., 50 джерел.

*Мета роботи* – Покращити продуктивність роботи системи вивчення технологій з веб-розробки.

*Об'єкт дослідження* – Система вивчення технологій з веб-розробки.

*Предмет дослідження* – Аналіз апаратного та програмного забезпечення.

*Короткий зміст роботи:* У роботі досліджено класифікацію апаратних та програмних забезпечень враховуючи статистичний аналіз використання у 2023 році. Показано результати виконання тестування системи вивчення технологій з веб-розробки. Визначено оцінку продуктивності системи. Виявлено поточні проблеми системи на для покращення. Обґрунтовано мінімально оптимальне апаратне та програмне забезпечення.

КЛЮЧОВІ СЛОВА: СИСТЕМА, РЕКОМЕНДАЦІЇ, ТЕСТУВАННЯ, СУЧАСНА ТЕОРІЯ ТЕСТУВАННЯ, НАВЧАННЯ, ТЕХНОЛОГІЇ, ВЕБ-РОЗРОБКА, РИНОК ІТ, ВЕБ-САЙТ.

## ABSTRACT

Text part of the master's qualification work: 115 pages, 34 pictures, 5 tables, 50 sources.

*The purpose of the work* – Improve the performance of the web development technology learning system.

*Object of research* – A system for studying web development technologies.

*Subject of research* – Analysis of hardware and software.

*Summary of the work:* The work examines the classification of hardware and software, taking into account the statistical analysis of use in 2023. The results of testing the web development technology learning system are shown. The assessment of system performance is determined. Current system issues identified for improvement. The minimum optimal hardware and software is justified.

KEYWORDS: SYSTEM, RECOMMENDATIONS, TESTING, ITEM RESPONSE THEORY, EDUCATION, TECHNOLOGIES, WEB DEVELOPMENT, IT MARKET, WEBSITE





## ЗМІСТ

<b>ВСТУП .....</b>	<b>11</b>
<b>1 ТЕОРЕТИЧНІ ОСНОВИ ТА МЕТОДИЧНІ ПІДХОДИ ДО СИСТЕМИ ВИВЧЕННЯ ТЕХНОЛОГІЙ ВЕБ-РОЗРОБКИ .....</b>	<b>15</b>
1.1 Класифікація апаратних та програмних забезпечень .....	15
1.1.1 Апаратне забезпечення .....	16
1.1.2 Програмне забезпечення .....	31
1.2 Визначення ключових показників ефективності системи .....	40
1.2.1 Загальні функціональні можливості Dev Tools .....	41
1.2.2 Функціональні можливості вкладки Performance у Dev Tools .....	43
1.2.3 Ключові показники ефективності для системи вивчення технологій з веб-розробки .....	45
1.3 Архітектура системи вивчення технологій з веб-розробки .....	46
1.3.1 Базові та сучасні технології з веб-розробки .....	46
1.3.2 Математичний огляд рішення .....	50
1.3.3 Сучасна теорія тестування .....	62
1.3.4 Інтерфейс системи .....	78
Висновки до розділу 1 .....	79
<b>2 АНАЛІЗ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ .....</b>	<b>80</b>
2.1 Статистичний аналіз використання апаратного та програмного забезпечення .....	80
2.1.1 Статистика використання апаратного забезпечення у 2023 році .....	81
2.1.2 Статистика використання програмного забезпечення у 2023 році .....	85
2.2 Проведення тестування системи на різному апаратному та програмному забезпеченні .....	89



	9
2.3 Оцінка продуктивності системи .....	91
Висновки до розділу 2 .....	92
<b>3 ВИЗНАЧЕННЯ ОПТИМАЛЬНОГО АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ ВИВЧЕННЯ ТЕХНОЛОГІЙ ВЕБ-РОЗРОБКИ .....</b>	<b>93</b>
3.1. Оцінка мінімальних вимог .....	93
3.2 Вибір програмного забезпечення .....	94
3.3 Оптимізація та ефективність системи .....	95
Висновки до розділу 3 .....	96
<b>ВИСНОВКИ .....</b>	<b>97</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>98</b>
Додаток А. Результат перевірки тестувань використовуючи Item Response Theory .....	103
Додаток Б. Інтерфейс системи .....	104
Додаток В. Проведення тестування системи на різному апаратному та програмному забезпеченні .....	108
Додаток Г. Оцінка продуктивності системи .....	114
<b>ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) .....</b>	<b>116</b>

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ІТ – інформаційні технології

ООП – об'єктно-орієнтоване програмування

СНР – системи надання рекомендацій

СТТ – сучасна теорія тестування

ПК – персональний комп'ютер

HTTP – Hypertext Transfer Protocol

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

SASS – Syntactically Awesome Style Sheet

LESS – Leaner Style Sheets

## ВСТУП

В епоху Інтернету [1], який став ключовою частиною повсякденного життя, багато аспектів нашого існування перейшли в онлайн-формат, у тому числі освіта у школах та університетах, робочі процеси, спілкування з близькими та різноманітні види розваг. Це сприяло зростанню інтересу до розуміння процесу створення онлайн-сервісів, якими люди користуються щодня. Щоб створити власні веб-системи, багато хто обирає навчальні курси, де ознайомлюються з основами веб-розробки.

### **Актуальність роботи**

Однією з актуальних проблем сьогодні є труднощі, з якими стикаються люди після завершення навчальних курсів, у оцінці своїх реальних знань та навичок у галузі веб-розробки, які вони здобули. Є важлива потреба у розробці системи, яка була б орієнтована на потреби користувача та вимоги ІТ-ринку [2], сумісна з різними сучасними пристроями. Така система мала б забезпечити можливість оцінки рівня знань користувача, а також надати рекомендації щодо матеріалів для повторення або вивчення, які допоможуть користувачеві досягти кваліфікацій, необхідних для посади розробника.

### **Порівняння роботи з відомими розв'язаннями проблеми**

Хоча існує чимало сервісів для тестування знань у сфері веб-розробки, вони часто не надають те, що найбільш потрібно користувачам: реальну допомогу в становленні як розробника. Ці системи зазвичай обмежуються наданням можливості пройти тест та отримати загальну оцінку, але не вказують на конкретний рівень знань та не дають порад, які допоможуть користувачеві розвинути свої навички до рівня, необхідного для професіонала у веб-розробці.

### **Мета і завдання дослідження**

**Мета роботи** – Покращити продуктивність роботи системи вивчення технологій з веб-розробки.

В процесі дослідження вирішувалися наступні завдання:

- 1. Визначення ключових показників, що є важливими під час тестування та впливають на ефективність роботи системи.*
- 2. Розглянути архітектуру побудови системи вивчення технологій з веб-розробки, для розуміння процесів та оцінки їх ефективності роботи.*
- 3. Виконати статистичний аналіз використання апаратного та програмного забезпечення за 2023 рік, що дозволить провести тестування системи на популярних та важливих забезпеченнях.*
- 4. Враховуючи результати з проведеного статистичного аналізу, провести тестування системи на необхідному апаратному та програмному забезпеченні.*
- 5. Провести оцінку продуктивності системи завдяки ключовим показникам ефективності.*
- 6. Визначити та обґрунтувати мінімально оптимальне апаратне та програмне забезпечення для системи вивчення технологій з веб-розробки.*

**Об'єкт дослідження** є система вивчення технологій з веб-розробки.

**Предметом дослідження** є аналіз апаратного та програмного забезпечення.

#### **Методи дослідження**

1. Запуск та проходження тестування на різному апаратному і програмному забезпеченні, з фіксацією результатів роботи системи завдяки вбудованим інструментам розробника у браузерях;
2. Аналіз отриманих результат з перевіркою ключових показників для пошуку оптимально мінімальних вимог, а також недоліків системи.

#### **Новизна одержаних результатів**

Було досліджено вплив використання різного апаратного та програмного забезпечення для оптимізації роботи системи вивчення технології з веб-розробки; зокрема було винайдено оптимальне апаратне та програмне забезпечення, яке зменшує час обробки результатів тестування та пришвидшую рендеринг системи.

## **Практичне значення одержаних результатів**

Розроблена система володіє комерційною цінністю як інструмент, що надає рекомендації для вивчення технологій з веб-розробки, та може використовуватися як:

- Доповнення до існуючих систем навчання веб-розробки;
- Окремий сервіс для отримання спеціалізованих рекомендацій у галузі веб-розробки.

Оскільки система має комерційну цінність, її оптимізація відіграє ключову роль. Покращення часу обробки результатів тестування та прискорення рендерингу системи значно збільшує її вартість, роблячи її більш ефективною та привабливою для потенційних користувачів.

## **Апробація результатів та публікації**

За даною роботою було взято участь у конференціях:

1. *I Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»*
2. *Науково-практична конференція «Проблеми комп'ютерної інженерії»*
3. *V Міжнародна науково-практична конференція молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології» (SoftTech-2023)*

Дана робота була опублікована у журналі категорії В:

*«Поперешняк С.В., Куценко М.О., Антоненко А.В. (2023). Система надання рекомендацій щодо вивчення технологій з веб-розробки. Таврійський науковий вісник. Серія: Технічні науки, Випуск 6 (2), 11-23.»*

Дана робота була опублікована у журналі *«Lecture Notes in Computational Intelligence and Decision Making»*, який входить у *Web of Science*.

*«Yurchuk I., Kutsenko M. (2022) An Intelligent System for Providing Recommendations on the Web Development Learning. In: Babichev S., Lytvynenko V. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2021. Lecture Notes on Data Engineering and Communications Technologies, vol 77. Springer, Cham.»*

Посилання: [https://link.springer.com/chapter/10.1007/978-3-030-82014-5\\_38](https://link.springer.com/chapter/10.1007/978-3-030-82014-5_38)

### **Структура роботи**

Робота викладена на 105 сторінках друкованого тексту, який складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків (А, Б, В, Г.). Робота містить 5 таблиць, 34 рисунки та 4 додатки, обсягом 13 стор.

# 1 ТЕОРЕТИЧНІ ОСНОВИ ТА МЕТОДИЧНІ ПІДХОДИ ДО СИСТЕМИ ВИВЧЕННЯ ТЕХНОЛОГІЙ ВЕБ-РОЗРОБКИ

У наш час існує велика кількість пристроїв, якими люди користуються для повсякденних задач, або роботи і навчання. Кожен з цих пристроїв є унікальним, оскільки вони відрізняються як зі сторони апаратного забезпечення, так і зі сторони програмного та операційного забезпечення. Оскільки всі люди різні, то кожен сам обирає яке апаратне та програмне забезпечення йому подобається. Сучасний бізнес та системи, мають бути до цього готові [3], та підтримувати усі сучасні апаратні та програмні забезпечення, щоб якомога більше користувачів могли легко та комфортно використовувати сервіс. Саме тому, розуміння сучасних потреб є головною частиною під час розробки будь-якої системи. Але не дивлячись на просто розробку, система має бути оптимізованою, оскільки не кожен має потужні девайси.

## 1.1 Класифікація апаратних та програмних забезпечень

У сучасному світі веб-розробки, важливість правильного вибору апаратного та програмного забезпечення не може бути недооцінена. Цей розділ присвячений детальному аналізу та класифікації ключових компонентів, які формують основу ефективного та продуктивного процесу веб-розробки [4]. Він охоплює широкий спектр апаратного забезпечення, від персональних комп'ютерів до серверного обладнання, та аналізує різноманітні програмні засоби, такі як операційні системи та веб-браузери, що відіграють ключову роль у розробці та впровадженні веб-проектів.

З розвитком технологій, вибір апаратного та програмного забезпечення стає дедалі складнішим завданням. Тому, цей розділ також надає інсайти щодо критеріїв

вибору та інтеграції цих елементів для досягнення оптимальної продуктивності та ефективності у веб-розробці. Зрозуміння того, як різні апаратні та програмні компоненти впливають на розробку та впровадження веб-додатків, є критично важливим для кожного веб-розробника, що прагне досягти успіху в цій швидкозмінній індустрії.

### **1.1.1 Апаратне Забезпечення**

У цьому розділі розглядається апаратне забезпечення як ключовий елемент у підтримці різноманітних користувацьких пристроїв у сфері інформаційних технологій. Розуміння того, як різні типи апаратного забезпечення взаємодіють і впливають на користувацький досвід, є критично важливим для створення систем, які є доступними та зручними для користувачів, незалежно від їхнього обраного пристрою.

З огляду на широкий спектр пристроїв, якими сьогодні користуються люди — від мобільних телефонів до настільних комп'ютерів, важливо розуміти, як апаратне забезпечення може впливати на доступність та ефективність систем. В цьому розділі я зосереджую увагу на аналізі різноманітних типів апаратного забезпечення, оцінюючи їх важливість та вплив на створення універсальних та доступних систем для широкого спектра користувачів.

#### **1.1.1.1 Персональні Комп'ютери**

Персональні комп'ютери, які охоплюють як настільні, так і портативні системи, є незамінними у великій кількості професійних сфер, особливо у веб-розробці. Вони забезпечують базову платформу для розробки, тестування та впровадження веб-додатків, поєднуючи в собі потужність, надійність та гнучкість. У цьому розділі я зосереджую увагу на ролі та важливості ПК в контексті веб-



розробки, розглядаючи ключові аспекти, які роблять їх незамінними інструментами для будь-якого розробника.

Настільні комп'ютери, зокрема, відіграють вирішальну роль у веб-розробці. Завдяки своїй модульності та легкості у апгрейді, вони дозволяють розробникам налаштовувати систему під свої конкретні потреби, чи то збільшення об'єму оперативної пам'яті, встановлення потужніших процесорів чи графічних карт. Також, настільні ПК часто мають краще охолодження та здатні ефективніше працювати з великими об'ємами даних і складними додатками, що є важливим для розробки високопродуктивних веб-систем.

Крім технічних характеристик, важливою є також ергономіка та зручність роботи з настільними ПК. Розробники часто проводять довгі години за комп'ютером, і можливість налаштувати робоче місце з урахуванням індивідуальних потреб та комфорту є критично важливою. Ергономічні монітори, клавіатури та інші периферійні пристрої допомагають зберегти продуктивність та запобігають втомі.

### **Настільні комп'ютери**

У сучасному бізнес-середовищі, успіх часто вимірюється швидкістю отримання, обробки та аналізу важливих даних. В цьому контексті ключову роль відіграють мобільні технології. Переносні комп'ютери, смартфони та планшети стали невід'ємною частиною ділового світу, оскільки вони забезпечують постійний доступ до інформації, незалежно від місця перебування користувача. Мобільність і гнучкість, які надають ці пристрої, є критично важливими для підтримки динамічного темпу сучасного бізнесу.

Стандартний настільний персональний комп'ютер, який є основним інструментом в офісах, традиційно складається з наступних компонентів:

- *Системний блок* – Це основний компонент, який містить основну плату, процесор, оперативну пам'ять, жорсткий диск, джерело живлення та інші внутрішні компоненти.

- *Клавіатура і миша* – Ці пристрої введення дозволяють користувачам взаємодіяти з комп'ютером, вводячи текст та вказуючи команди.
- *Монітор* – Візуальний інтерфейс, що відображає графічну та текстову інформацію.

Додаткові периферійні пристрої часто включають:

- *Принтер* – Пристрій для виведення документів та графіки на папір.
- *Сканер* – Використовується для цифрового перетворення фізичних документів у електронні формати.
- *Зовнішні накопичувачі* – Жорсткі диски, SSD, флеш-накопичувачі для додаткового зберігання даних.
- *Мережні пристрої* – Такі як Wi-Fi-адаптери або Ethernet-порти для підключення до мережі Інтернет або корпоративної мережі.

У сучасному світі, зі зростанням популярності хмарних технологій та послуг, багато традиційних функцій настільних комп'ютерів переміщуються в онлайн-середовище. Це означає, що фізичне обладнання вже не обов'язково має бути потужним, оскільки значну частину обчислень та зберігання даних можна виконувати в хмарі.

Оскільки ключовим компонентом настільного комп'ютера є системний блок, розглянемо детальніше його основні компоненти:

### *Материнська плата*

Є основою системного блоку, до якої підключаються усі інші компоненти. Включає в себе роз'єми для процесора, слоти для оперативної пам'яті, роз'єми розширення (наприклад, PCI, PCIe), порти для підключення внутрішніх і зовнішніх пристроїв (USB, HDMI, Ethernet тощо).

### *Процесор (Центральний процесорний блок - CPU)*

Процесор це мозок комп'ютера, що відповідає за обробку інструкцій та керування іншими компонентами. Розміщується в спеціальному роз'ємі на материнській платі.

### Оперативна пам'ять (RAM)

Використовується для тимчасового зберігання даних, які активно використовуються процесором. Встановлюється в спеціальні слоти на материнській платі.

### Жорсткий диск (HDD) або твердотілий накопичувач (SSD)

Використовується для постійного зберігання даних і програм. Підключається до материнської плати через SATA-інтерфейс або через M.2 слот (у випадку деяких SSD).

### Блок живлення

Забезпечує електроенергією всі компоненти системного блоку. Підключається до материнської плати, жорсткого диска та інших компонентів, які потребують живлення.

### Відеокарта (GPU)

Відповідає за обробку графічної інформації та виведення зображення на монітор. Може бути інтегрованою в материнську плату або процесор, або встановлюватися як окремий компонент у слот розширення.

### Звукова карта

Відповідає за обробку звукових сигналів. Часто інтегрована в материнську плату, але також може бути окремим компонентом.

### Оптичні приводи (наприклад, DVD, Blu-ray)

Використовуються для читання та запису оптичних дисків, але в ході сучасних змін стає менш актуальним. Підключаються до материнської плати через SATA-інтерфейс.

### Кулери і система охолодження

Забезпечують охолодження компонентів особливо важливо для процесора та відеокарти. Можуть включати в себе кулери, теплові трубки, радіатори та в деяких випадках системи водяного охолодження.

## Портативні комп'ютери (Ноутбуки)

Портативний персональний комп'ютер, відомий також як лептоп, є компактним варіантом персонального комп'ютера, зазвичай важить до 4 кг, поєднуючи в собі функціонал системного блоку та пристроїв вводу/виводу. Ці комп'ютери оснащені власними джерелами живлення для автономної роботи та відрізняються за розмірами, вагою, продуктивністю та використовуваними операційними системами.

Ноутбук, який також можна називати мобільним мікрокомп'ютером, має одночастинний корпус з вагою від 1 до 4 кг. Його електроживлення може здійснюватися від акумуляторних батарей або через електромережу. Особливістю ноутбуків є їх оснащеність рідкокристалічними дисплеями, енергонезалежною пам'яттю, різноманітними накопичувачами, CD-ROM приводами, часто - вбудованими модемами, а також можливостями підключення до зовнішніх пристроїв. Ноутбуки пропонуються на ринку багатьма компаніями в різних моделях, деякі з них мають характеристики, порівнянні з настільними ПК.

Серед провідних виробників ноутбуків [5] - такі компанії, як IBM, Apple, Compaq, Dell, Toshiba. Класифікація ноутбуків відбувається за критеріями якості та розміру екрана, типу процесора, обсягу оперативної та дискової пам'яті, потужності батарей, типу вбудованого дисководу та наявності модему, а також за масо-габаритними характеристиками.

Ноутбуки класифікують за масо-габаритними параметрами на середньоформатні, тонкі (Slim) та надтонкі (Ultraslim). Більшість ноутбуків мають засоби для підключення до Інтернету, включно з інтегрованими мережними адаптерами, модемами та бездротовими рішеннями. Однак основним недоліком ноутбуків часто є їх порівняно нижча продуктивність порівняно з настільними ПК.

### 1. Структурні компоненти ноутбука.

У ноутбуків всі компоненти зібрані в одному корпусі, який складається з двох основних частин. Перша, тонша частина, яка зазвичай розгортається під кутом 90 градусів, містить в собі рідкокристалічний монітор. Цей монітор

складається з матриці, до якої під'єднані гнучкі шлейфи для передачі зображення.

Щодо РК-екранів, існують два основних типи – глянсові та матові. Глянсові екрани мають оптичне покриття, що зменшує відблиски і розсіювання світла, забезпечуючи яскравіше та чіткіше зображення, порівняно з матовими екранами. Матові екрани, у свою чергу, краще підходять для роботи в умовах з високим рівнем освітленості.

На тій же частині корпусу, залежно від моделі, можуть бути встановлені мікрофон, web-камера, інвертор для підсвічування матриці, а також антени для модулів Bluetooth та WI-FI.

У другій частині корпусу розташовані клавіатура, кнопка включення, світлові індикатори, тачпад, динаміки, дисковод, різноманітні інтерфейси, включаючи USB-порти, роз'єми для зарядки, micro-SD, HDMI, VGA та інші.

Майже кожен ноутбук має адаптер для підключення до локальної мережі. Сучасні моделі зазвичай оснащені Ethernet-картами з підтримкою швидкості до 1 Гбіт/с.

Корпус ноутбука зазвичай виготовлений з міцного пластику та зсередини обшитий металеву фольгою для захисту від зовнішніх електромагнітних полів, що забезпечує надійний захист внутрішніх компонентів.

Якщо зняти задню кришку корпусу, можна буде побачити внутрішню побудову ноутбуку, як зображено на Рисунку 1.1.



Рисунок 1.1 - Внутрішня побудова ноутбуку

Розглянемо усі запчастини по окремої:

### Оперативна пам'ять

Оперативна пам'ять у ноутбуках відіграє ключову роль у загальній продуктивності системи, незважаючи на її компактні розміри. Оперативна пам'ять необхідна для тимчасового зберігання даних та команд, які використовує процесор під час виконання завдань. У ноутбуках, як правило, використовуються пам'яті типу DDR2 SO DIMM та DDR3 SO DIMM. Ці типи пам'яті є адаптацією стандартної DDR SDRAM для потреб мініатюризації, характерної для ноутбуків. Останнім часом у продажу з'являються все більше моделей з DDR3 SO DIMM, оскільки цей тип пам'яті забезпечує кращу продуктивність і енергоефективність.

### Оптичний привод

Завдяки відсутності традиційного механічного висувного лотка, який займав би додатковий простір, оптичний привод у ноутбуках здатен зберігати свою компактність. На ринку можна зустріти як стандартні DVD приводи, так і більш сучасні Blu-ray приводи в дорожчих моделях ноутбуків.

### Система охолодження

Система охолодження в ноутбуках включає вентилятор та систему теплових трубок для забезпечення циркуляції повітря. Холодне повітря засмоктується знизу корпусу та проходить через гарячі компоненти, ефективно охолоджуючи їх.

### Центральний процесор

Центральний процесор в ноутбуках відрізняється від настільних аналогів меншими розмірами та зниженим енергоспоживанням. Окрім того, в них впроваджені спеціальні технології оптимізації споживання енергії. Технології, як Intel SpeedStep [6] та AMD PowerNow [7], дозволяють знижувати тактову частоту процесора в залежності від поточних завдань, що сприяє збільшенню часу автономної роботи. Сьогодні на ринку лідирують дві компанії-виробники процесорів - Intel і AMD, кожна з яких має своїх прихильників та опонентів.

### Відеоадаптер

Відеоадаптер у ноутбуках буває двох видів: *інтегрований у чипсет* або *дискретний*, представлений окремими мікросхемами. Дискретні відеокарти відзначаються більшою потужністю та гнучкістю, оскільки їх можна замінити без заміни материнської плати. Провідними виробниками відеоадаптерів є AMD та NVidia.

### Жорсткий диск

Жорсткий диск (вінчестер) у ноутбуках має стандартний форм-фактор 2,5 дюйма, проте відрізняється від настільних аналогів меншою швидкістю обертання та зменшеними габаритами. Сучасні ноутбуки в основному використовують SATA інтерфейс для підключення жорстких дисків.

### Материнська плата

Центральним елементом ноутбука, що об'єднує всі функціональні компоненти є материнська плата. Її якість і

характеристики залежать від загальної конфігурації та потужності ноутбука.

### Акумулятор

Для забезпечення автономної роботи ноутбука, акумулятор є ключовим компонентом. Він контролює температуру, кількість циклів заряду-розряду, напругу і часто виготовляється на основі літій-іонних технологій. Важливо дотримуватися правил експлуатації акумулятора для забезпечення його тривалого терміну служби.

## 2. Класифікація ноутбуків.

Розглянемо різні типи ноутбуків за діагоналлю дисплея:

### Desktop Replacement

Ці ноутбуки мають діагональ дисплея 17 дюймів і більше. Вони характеризуються значними габаритами та вагою, що робить їх незручними для частого перенесення. Проте великий розмір дисплея забезпечує більший комфорт при роботі, а просторий корпус дозволяє встановити потужніші компоненти і забезпечити ефективне охолодження.

### Масові ноутбуки

Ця категорія охоплює моделі з діагоналлю дисплея від 14 до 16 дюймів. Вони балансують між габаритами, вагою та продуктивністю, пропонуючи задовільний рівень продуктивності при прийнятних розмірах і вазі.

### Субноутбуки

З діагоналлю екрану від 11 до 13,3 дюймів, субноутбуки відрізняються своєю компактністю та легкістю, але маленький розмір екрана може знижувати зручність користування. Через обмежений простір для компонентів, у цих ноутбуках часто використовуються



мобільні процесори з низьким енергоспоживанням, що також впливає на загальну продуктивність.

### Нетбуки

З діагоналлю дисплея від 7 до 12,1 дюйма, нетбуки призначені для базових завдань, таких як веб-серфінг, робота з електронною поштою та офісними програмами. Незважаючи на обмежену продуктивність, малі габарити та вагу, нетбуки приваблюють користувачів своєю доступністю та тривалішим часом автономної роботи. Через компактні розміри вони зазвичай не мають оптичних дисків, але оснащені Wi-Fi-адаптерами.

### Handheld PC

Ноутбуки з діагоналлю дисплея 7 дюймів і менше виокремлюються у спеціальну категорію. Вони є надзвичайно портативними, але зазвичай обмежені у функціональності порівняно з більшими моделями.

Окрім того, розглянемо детальніше категорії ноутбуків за їх призначенням та технічними характеристиками:

### Бюджетні ноутбуки

Ці ноутбуки пропонуються за низькою ціною та мають обмежені можливості. Вони часто мають скромний час автономної роботи. Діагональ екрану бюджетних моделей зазвичай становить 14-15 дюймів, але існують варіанти і з більшою діагоналлю. Нетбуки, з діагоналлю екрана 7 – 10,2 дюйма, також входять у цю категорію.

### Ноутбуки середнього класу

Це найбільша категорія ноутбуків, що включає моделі з різною діагоналлю екрана та середнім рівнем продуктивності. Корпус цих ноутбуків зазвичай виготовлений з пластику. Моделі середнього класу

часто класифікуються як офісні, мультимедійні ноутбуки економ-класу або навіть як ігрові ноутбуки економ-класу.

#### Бізнес-ноутбуки

Ці ноутбуки схожі на моделі середнього класу, але відрізняються строгим дизайном та використанням дорожчих матеріалів. Вони часто відносяться до категорії субноутбуків, рідше – до Desktop Replacement.

#### Мультимедійні ноутбуки

Ці ноутбуки оснащені відеокартами та процесорами середнього класу, що робить їх придатними для більшості завдань, у тому числі і для ігор. Діагональ екрану зазвичай становить 15 – 17 дюймів.

#### Ігрові ноутбуки

Призначені спеціально для ігор, оснащені потужними процесорами та відеокартами. Деякі моделі навіть мають два графічних адаптери, що працюють у режимі SLI/Crossfire.

#### Мобільна робоча станція

Це потужні ноутбуки, призначені для професійних завдань, які вимагають високої обчислювальної потужності.

#### Іміджеві ноутбуки

Відрізняються унікальним дизайном та використанням незвичайних матеріалів. Часто належать до класу субноутбуків, але можуть включати і моделі ігрових чи бізнес-ноутбуків.

#### Захищені ноутбуки

Ці ноутбуки призначені для роботи в складних умовах. Вони стійкі до ударів, вологи, екстремальних температур і інших агресивних факторів.

#### Ноутбуки із сенсорним дисплеєм

Це гібридні пристрої, що поєднують функціональність планшета і ноутбука. Мають поворотний сенсорний екран, що дозволяє використовувати їх як в якості звичайного ноутбука, так і як планшет.

### **1.1.1.2 Серверне Обладнання**

Розділ присвячений серверному обладнанню, яке відіграє надзвичайно важливу роль у світі інформаційних технологій та веб-розробки. Сервери забезпечують функціонування веб-сайтів, додатків та баз даних, є основою для хмарних обчислень та надають необхідні ресурси для обробки великих обсягів даних. Вони відіграють критичну роль в підтримці стабільності, швидкості та безперервності веб-послуг, що є життєва важливими аспектами в сучасному цифровому світі.

У цьому розділі розглядаються різні типи серверного обладнання, включаючи фізичні сервери, віртуальні машини та хмарні рішення. Кожен тип має свої переваги та особливості використання, залежно від потреб користувачів та вимог до проекту. Окрім того, аналізуються критерії вибору серверного обладнання, включаючи обчислювальну потужність, обсяг пам'яті, можливості масштабування та безпеки. Важливо також розглянути аспекти управління серверами, включаючи системне адміністрування, моніторинг ресурсів та забезпечення високої доступності послуг. Серверне обладнання є не лише фундаментом для стабільної роботи веб-додатків, але й ключовим елементом у забезпеченні гнучкості та масштабованості веб-проектів, що має вирішальне значення для забезпечення високої якості обслуговування користувачів.

### **Фізичні сервери**

Прикладні комерційні та бізнесові системи, які обслуговують значну кількість користувачів, включаючи системи управління базами даних, обробки транзакцій, великі видавничі та мережеві системи, а також системи розробки ПЗ і обробки зображень, вимагають використання моделі обчислень "клієнт-сервер" [8]

та розподіленої обробки. У такій моделі, сервер виконує значну частину обробки даних, в той час як інша частина роботи виконується на комп'ютері користувача. Інколи, як сервер, так і клієнт можуть функціонувати на одному і тому ж комп'ютері.

Існує кілька типів серверів для різних застосувань, зокрема файловий сервер, сервер бази даних, принт-сервер, обчислювальний сервер та сервер застосувань. Кожен тип сервера спеціалізується на певному ресурсі – це може бути файлова система, база даних, принтери, процесори або програмні пакети.

Сервери також можна класифікувати за масштабом мережі, в якій вони використовуються, наприклад, сервери для робочих груп, відділів або корпоративні сервери. Проте така класифікація є досить умовною, оскільки розмір групи може варіюватися від декількох осіб до сотень, а сервер відділу може обслуговувати від 20 до 150 користувачів. Залежно від кількості користувачів і виду їхніх завдань, вимоги до апаратного і програмного забезпечення сервера, його надійності та продуктивності можуть суттєво відрізнятися.

Файлові сервери для невеликих робочих груп (до 20-30 осіб) зазвичай базуються на платформі персональних комп'ютерів і використовують програмне забезпечення Novell NetWare. У цьому випадку файл-сервер служить центральним сховищем даних. Типовий набір компонентів для таких серверів включає процесор, основну і зовнішню пам'ять, а також адаптер Ethernet. Часто до складу сервера входять дисковод гнучких дисків і дисковод компакт-дисків. Оскільки графічні потреби для більшості серверів не мають принципового значення, достатньо використовувати стандартний монохромний монітор з невисокою роздільною здатністю. Рекомендується використання пристрою безперебійного живлення для забезпечення стабільності роботи.

Для файл-серверів загального доступу, які можуть одночасно обслуговувати десятки або навіть сотні користувачів, використання одно процесорної платформи та програмного забезпечення Novell може бути недостатнім. У таких випадках застосовуються більш потужні багато процесорні сервери, які мають наступні характеристики:

- *Збільшена Оперативна Пам'ять:* Можливість нарощування оперативної пам'яті до декількох гігабайтів, що забезпечує високу продуктивність при роботі з великими обсягами даних.
- *Розширений Дисковий Простір:* Великий обсяг дискового простору, що може досягати сотень гігабайтів, дозволяє зберігати великі масиви даних.
- *Швидкісні Інтерфейси Обміну Данями:* Використання інтерфейсів типу Fast SCSI-2, Fast&Wide SCSI-2, і Fiber Channel забезпечує швидкий доступ до даних.
- *Множинні Мережні Інтерфейси:* Наявність кількох мережних інтерфейсів підвищує гнучкість і надійність мережевих з'єднань.
- *Операційна Система:* Ці сервери зазвичай працюють на операційній системі UNIX, що підтримує мережеві протоколи TCP/IP і NFS.
- *Сервери Баз Даних:* Багатопроцесорні UNIX-сервери часто використовуються як сервери баз даних у великих інформаційних системах, так як вони здатні впоратися з великим навантаженням, що пов'язане з обробкою запитів до бази даних.
- *Надійність та Продуктивність:* Супер сервери мають високу загальносистемну продуктивність, функціональні можливості компонентів, стійкість до відмов, а також підтримують багатопроцесорну обробку, системне адміністрування та дискові масиви великої ємності.
- *Архітектура та Надійність:* Сучасні супер сервери часто мають два або більше центральних процесорів, багатоштинну структуру, достатні можливості нарощування обчислювальної потужності та дискового простору, а також засоби для забезпечення надійності зберігання даних та захисту від несанкціонованого доступу.

Такі супер сервери на сьогоднішній день можуть конкурувати з мейнфреймами за рівнем продуктивності та надійності, що робить їх відмінним вибором для великих і складних обчислювальних завдань.

## Віртуальні сервери та хмарні рішення

Хмарні обчислення Google надають різноманітні сервіси, які можна класифікувати в основні категорії хмарних сервісів, зазначені вище. Ось приклади сервісів Google Cloud Platform (GCP), що належать до кожної з цих категорій:

### 1) Інфраструктура як сервіс (IaaS)

- *Google Compute Engine*: Провідний IaaS-сервіс від Google, який надає віртуальні машини в хмарі.
- *Google Cloud Storage*: Послуга зберігання даних, яка пропонує надійне та масштабоване зберігання даних.

### 2) Платформа як сервіс (PaaS):

- *Google App Engine*: Платформа для розробки та хостингу веб-додатків, яка автоматично масштабується відповідно до потреб користувачів.
- *Google Kubernetes Engine*: Сервіс для оркестрації контейнеризованих додатків з використанням Kubernetes.

### 3) Програмне забезпечення як сервіс (SaaS):

- *Google Workspace (раніше G Suite)*: Набір хмарних офісних інструментів, що включає Gmail, Docs, Drive, Calendar та інше.
- *Google Cloud's AI та ML сервіси*: Набір інструментів машинного навчання та штучного інтелекту, доступних як SaaS.

### 4) Хмарні зберігальні системи:

- *Google Cloud SQL*: Хмарна база даних для MySQL, PostgreSQL та SQL Server.
- *Google Cloud Bigtable*: Хмарне NoSQL-зберігання для великих обсягів даних.

### 5) Хмарні послуги обробки даних:

- *Google BigQuery*: Хмарний сервіс для швидкого аналізу великих даних.

- Google Cloud Dataflow: Хмарний сервіс для обробки поточкових та пакетних даних.

б) Хмарні послуги для розробників (DevOps):

- *Google Cloud Build*: Сервіс для автоматизації збірки, тестування та розгортання додатків.
- *Google Cloud Source Repositories*: Хмарний сервіс управління кодом та колаборації для розробників.

Google Cloud Platform надає інші послуги, які охоплюють широкий спектр хмарних потреб, від мережевих технологій до штучного інтелекту, безпеки, управління API та іншого.

### 1.1.2 Програмне Забезпечення

Аналіз та класифікацію програмного забезпечення, є невід'ємною частиною будь-якої IT-інфраструктури, зокрема в контексті веб-розробки. Програмне забезпечення включає в себе різноманітні системи та додатки, які забезпечують функціонування апаратного обладнання та виконують специфічні завдання, необхідні для розробки, тестування та підтримки веб-додатків.

У цьому розділі особлива увага приділяється операційним системам, які слугують фундаментом для всіх інших типів програмного забезпечення. Операційні системи визначають базові принципи взаємодії між апаратним забезпеченням та різноманітними програмами, впливаючи на продуктивність, безпеку та зручність користування системою. Також розглядаються веб-браузери, які є основним інтерфейсом для доступу до веб-додатків та веб-ресурсів, і відіграють ключову роль у забезпеченні високоякісного користувацького досвіду.

Окрім основних категорій, у розділі також висвітлюються додаткові аспекти програмного забезпечення, такі як різноманітні утиліти, інструменти для розробки та тестування, а також системи управління базами даних. Кожен з цих компонентів

має своє специфічне застосування та важливість у створенні та підтримці ефективних веб-додатків.

Цей розділ підкреслює важливість правильного вибору та використання програмного забезпечення, що відіграє вирішальну роль у забезпеченні надійності, безпеки та ефективності веб-систем. Вибір програмного забезпечення впливає не тільки на технічні характеристики проекту, але й на його здатність задовольняти потреби та очікування кінцевих користувачів.

### **1.1.2.1 Операційні Системи**

Операційні системи (ОС) становлять основу для роботи будь-якого комп'ютерного обладнання, від персональних комп'ютерів до мобільних пристроїв та серверів. Цей розділ присвячений аналізу різних типів операційних систем, їх характеристикам та впливу на процес веб-розробки. Операційні системи є важливим чинником, який визначає як загальну продуктивність і стабільність системи, так і можливості для розробки та тестування веб-додатків.

Розглядаються найпоширеніші операційні системи, такі як Windows, macOS, Linux та їхні різні дистрибутиви. Кожна ОС має свої унікальні особливості, переваги та недоліки, що впливають на вибір розробників та адміністраторів. Наприклад, Linux є відомим своєю гнучкістю та безпекою, а Windows — своєю широкою сумісністю з різноманітним програмним забезпеченням.

Важливим аспектом є також адміністрування та налаштування операційних систем. Правильно налаштована ОС може значно підвищити продуктивність роботи, забезпечити стабільність та надійність. Розглядаються інструменти та практики для оптимізації ОС, що включають керування ресурсами, налаштування безпеки та моніторинг системи.

Окрім того, аналізуються тенденції та нововведення в сфері операційних систем, їх вплив на майбутнє веб-розробки та адаптація до зростаючих вимог сучасного цифрового світу. Операційні системи є динамічною та постійно розвиваючою сферою, а їх вибір та використання мають стратегічне значення для успішної реалізації веб-проектів.



## **Windows**

Windows [9] - це сімейство операційних систем, розроблене компанією Microsoft, яке стало фундаментом для значної частини комп'ютерного світу. Ці системи базуються на графічному інтерфейсі користувача і відіграли вирішальну роль у популяризації інноваційних способів взаємодії між людиною та машиною, а також між різними комп'ютерними системами. Windows створило середовище, дружнє як для взаємодії користувачів з комп'ютерними програмами, так і для взаємодії між апаратними компонентами в межах одного комп'ютера.

На сьогоднішній день більшість комп'ютерів у світі працюють під управлінням операційних систем сімейства Windows, які займають більше 90% ринку комп'ютерних систем загалом і приблизно 95% ринку персональних комп'ютерів. Це сімейство операційних систем включає різні версії, кожна з яких орієнтована на певний сегмент ринку, від домашніх користувачів до бізнесу і навіть серверних систем.

Одним з ключових аспектів успіху Windows є її здатність підтримувати широкий спектр апаратного забезпечення та програмних додатків. Система забезпечує широку сумісність з різними комп'ютерними компонентами та периферійними пристроями, що робить її вибором для багатьох виробників ПК.

Windows також відома своїм інтуїтивно зрозумілим та користувацьким привітним інтерфейсом, який постійно розвивається і вдосконалюється. З кожною новою версією Microsoft прагне підвищити продуктивність, безпеку та зручність користування, інтегруючи такі функції, як віртуальні асистенти, покращену підтримку сенсорних екранів та інноваційні хмарні технології. Крім того, Microsoft активно працює над інтеграцією хмарних технологій у свої продукти, роблячи Windows більш гнучкою і адаптованою до сучасних потреб користувачів.

## **macOS**

За останні роки компанія Microsoft, яка є лідером на ринку операційних систем, поступово відчуває конкуренцію з боку інших виробників. ОС Windows,

хоча і залишається монополістом у цій сфері, стикається з активним розвитком інших систем, таких як Mac OS від Apple.

Mac OS (Macintosh Operating System) [10] - це пропріетарне сімейство операційних систем з графічним інтерфейсом, розроблених спеціально для комп'ютерів Apple Macintosh. Цікаво, що багато IT-експертів вважають Mac OS першою з сучасних операційних систем, яка використала графічний інтерфейс користувача, на відміну від традиційної командної строки. Це означає, що тепер користувачі могли взаємодіяти з усіма системними об'єктами та функціями через графічні компоненти екрану, такі як вікна, значки, меню, кнопки, списки тощо. У той час деякі програмісти порівнювали роботу в графічному інтерфейсі з управлінням власним світом, наскільки ця технологія була зручна.

Поява Mac OS відбулася у 1984 році, коли компанія Apple Computer представила персональний комп'ютер Macintosh. Новинка мала унікальні на той час можливості: користувачі керували своїм комп'ютером не тільки за допомогою команд, введених з клавіатури, але й за допомогою нового пристрою, званого "миша". Миша контролювала курсор, який у свою чергу на моніторі управляв графічними об'єктами, такими як папки, ярлики файлів тощо. Крім того, Mac OS вперше використовувала нині звичний нам віконний інтерфейс для представлення та організації інформації.

Порівнюючи Mac OS з її головним конкурентом - ОС Windows від Microsoft, можна виділити кілька основних відмінностей.

По-перше, надійність і стабільність роботи операційної системи: у цьому відношенні лідерство належить Apple.

По-друге, Mac OS відрізняється більш цікавим і практичним дизайном, який можна охарактеризувати як "нічого зайвого".

По-третє, Комп'ютери під управлінням Mac OS майже не схильні до інфекцій вірусами і стійкі до несанкціонованого доступу до особистої інформації.

Деякі експерти також виділяють іміджеві відмінності між двома системами. Macintosh з її Mac OS орієнтована переважно на графічні та мультимедійні функції і справляється з цими завданнями краще, ніж Windows. Натомість Windows

ефективніша у роботі з офісними додатками та статистикою. Рідко хто грає в комп'ютерні ігри на Macintosh, в той час як на ринку існує цілий сегмент ігрових ПК та спеціальних периферійних пристроїв під управлінням Windows.

## **Linux**

Linux [11], яка є популярною UNIX-системою кінця XX століття, виникла як доступна альтернатива більш дорогим рішенням, таким як HP-UNIX, SUN Solaris, Digital UNIX. На сьогоднішній день ця операційна система стрімко завойовує ринок, випереджаючи інші відомі ОС за темпами розвитку.

Однією з ключових переваг Linux є її універсальність: система придатна для широкого спектра застосувань, від персональних комп'ютерів до високопродуктивних многопроцесорних серверів і кластерів. Наразі Linux стабільно функціонує на багатьох платформах, включаючи PC, Alpha, PowerPC, Macintosh, SGI MIPS, Strong ARM, SGI Visual Workstations, VAX, 8086, PC-98, Palm Pilot, HP PA-RISC, Sparc & UltraSparc64 systems, m68k.

Як операційна система, Linux виконує багато функцій, які характерні для DOS і Windows, але відрізняється особливою потужністю і гнучкістю. Більшість операційних систем для персональних комп'ютерів, таких як DOS, були створені для ПК з обмеженими можливостями, які лише нещодавно перетворилися на універсальні машини. Такі ОС постійно модернізуються, щоб відповідати розвитку апаратних можливостей ПК. Натомість Linux розроблялася в іншому контексті: як версія Unix для ПК, система, яка протягом десятиліть використовувалася на мейнфреймах і міні-ЕОМ і є основною ОС робочих станцій. Linux пропонує швидкість, ефективність і гнучкість Unix, використовуючи при цьому переваги сучасних персональних комп'ютерів.

З фінансової точки зору, Linux має значну перевагу, оскільки це не комерційна ОС. На відміну від Unix, Linux поширюється безкоштовно за загальною відкритою ліцензією GNU в рамках Фонду вільного програмного забезпечення (Free Software Foundation), роблячи цю ОС доступною всім бажаючим. Незважаючи на те, що Linux захищена авторським правом, вона доступна завдяки відкритій

ліцензії GNU, яка дозволяє їй залишатися безкоштовною і стандартизованою системою. Існує лише один офіційний варіант Linux.

Спочатку операційні системи розроблялися для підвищення ефективності апаратних засобів. Вони були адаптовані до обмежених можливостей ранніх комп'ютерів, зосереджуючись не на користувачеві, а на апаратному забезпеченні. З часом ОС стали більш гнучкими, адаптуючись під потреби користувачів, а не тільки апаратного забезпечення.

На сьогоднішній день Linux є повноцінною операційною системою сімейства UNIX. Практично всі важливі програмні пакети доступні на Linux, що робить її універсальним інструментом для широкого спектру застосувань.

### **1.1.2.2 Веб-Браузери**

Веб-браузери є центральними інструментами в екосистемі веб-розробки, слугуючи мостом між користувачами та веб-додатками. Цей розділ розглядає різні веб-браузери, їхні особливості та значення у процесі веб-розробки. Веб-браузери не тільки забезпечують доступ до веб-ресурсів, але й впливають на сприйняття та взаємодію користувачів з веб-додатками, тому їх вибір та оптимізація мають важливе значення.

У цьому розділі аналізуються найпопулярніші веб-браузери, такі як Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, та інші. Кожен браузер має свої унікальні характеристики, від забезпечення високої продуктивності до різноманітних можливостей налаштування та розширень. Особливу увагу приділяється сумісності браузерів з різними веб-технологіями, їх впливу на відображення веб-контенту та взаємодію з користувачами.

Також важливим є розгляд безпеки та конфіденційності в веб-браузерах, оскільки вони є критичними аспектами у забезпеченні захисту даних користувачів та надійності веб-додатків. Обговорюються різні механізми безпеки, вбудовані у браузері, та їх роль у захисті від загроз, таких як шкідливе програмне забезпечення та фішинг [12].

Окрім того, є висвітлення тенденцій та інновацій у світі веб-браузерів, включаючи розвиток нових стандартів веб-технологій [13], реакцію на зміни у поведінці користувачів та адаптацію до нових вимог цифрового середовища. Веб-браузери є динамічною частиною веб-екосистеми, які кожна людина використовує кожен день за для власних потреб, таких як робота, навчання, досліди, і їх розвиток безпосередньо впливає на можливості та майбутнє веб-розробки.

## **Google Chrome**

Google Chrome [14], розроблений компанією Google, є безкоштовним веб-браузером, побудованим на основі проекту з відкритим кодом Chromium та іншого відкритого програмного забезпечення, включаючи компоненти WebKit та Mozilla. Цей браузер був спеціально створений для задоволення сучасних потреб користувачів, з особливим акцентом на веб-сайти, що функціонують як веб-додатки. Однією з ключових особливостей Google Chrome є його архітектура, яка була розроблена, щоб забезпечити підвищену стабільність, швидкість і безпеку, разом з чистим, простим і ефективним інтерфейсом користувача.

В основу функціонування Chrome закладено систему вкладок, де кожна вкладка працює як незалежний процес, що сприяє стабільності і безпеці роботи браузера. Ця особливість забезпечує краще відокремлення вмісту і запобігає збоєм однієї вкладки на інші. Крім того, Chrome знаний своєю швидкістю, як у завантаженні сторінок, так і в обробці JavaScript, завдяки використанню ефективного движка V8 [15].

Безпека є ще однією важливою характеристикою Google Chrome, оскільки браузер постійно оновлюється, щоб захистити користувачів від новітніх загроз і вразливостей. Вбудовані функції, а саме такі як захист від фішингу і шкідливого програмного забезпечення, допомагають користувачу у забезпеченні безпечного серфінгу в Інтернеті.

Google Chrome також відрізняється своїм простим і чистим інтерфейсом, що робить його зручним у використанні для користувачів різного рівня досвіду. Браузер підтримує різноманітні розширення та теми, що дозволяє користувачам

налаштувати його під свої персональні потреби і смаки. Синхронізація з Google акаунтом забезпечує зручний доступ до закладок, історії перегляду та інших налаштувань на будь-якому пристрої, де використовується Chrome.

## **Mozilla Firefox**

Mozilla Firefox [16], відомий також як Вогнелис чи Вогняна Лисиця українською, є вільним веб-браузером з відкритим кодом. Використовуючи удосконалене ядро Quantum, Firefox став важливим гравцем на ринку веб-браузерів. З моменту запуску версії 1.0, Firefox отримав позитивні відгуки та широку популярність серед користувачів і фахівців.

Firefox, який був другим за популярністю браузером після Internet Explorer у 2009 році, з часом здобув суттєвий приріст користувачів, незважаючи на зростаючу конкуренцію від Google Chrome.

У липні 2023 року Mozilla випустила останню версію Firefox 115.0 для Windows 7, Windows 8 та деяких версій macOS. Користувачі цих систем будуть переведені на версію Firefox 115 ESR, яка забезпечуватиме обмежену підтримку до вересня 2024 року, включаючи випуск патчів безпеки та оновлень, які не додаватимуть нових функцій.

Firefox доступний для широкого спектру операційних систем, включаючи Windows, Mac OS X, Linux, OS/2, FreeBSD, Solaris та інші, що робить його універсальним інструментом для веб-серфінгу.

## **Safari**

Safari [17], розроблений корпорацією Apple, є веб-браузером, що став частиною операційних систем macOS, iOS та iPadOS. У грудні 2003 року, після тривалого періоду використання Internet Explorer for Mac на комп'ютерах Mac, Apple анонсувала бета-версію свого нового браузера на MacWorld Expo. Існували очікування, що Apple використає движок Mozilla Gecko для свого браузера, але вони вирішили використовувати KHTML - движок веб-браузера Konqueror з KDE.

Цікавим є той факт, що одним з ключових учасників розробки Safari був Дейв Хаятт, який раніше працював над Netscape Navigator і був одним з розробників Firefox, використовуючи движок Gecko. Apple, однак, зробила ставку на KHTML, оскільки він був швидшим, легшим і краще підтримував стандарти, ніж Gecko.

Apple переробила KHTML, видаливши залежність від набору Qt і замінивши його на адаптер, який був сумісний з Cocoa API на мові Objective-C. Це призвело до створення відкритої бібліотеки WebCore, що базувалася на KHTML. Використовуючи JavaScript движок KDE kjs у поєднанні з власним Apple JavaScriptCore, вони створили WebKit - програмний каркас, який використовується в багатьох програмах Mac OS, включаючи Safari, для відображення HTML.

Safari розширив інтерфейс WebKit, додавши користувацькі функції, подібно до того, як Firefox розширив движок Mozilla Gecko. Як і Mozilla, Apple отримувала прибуток від партнерства з Google. Основною причиною розробки Safari була потреба платформи Mac OS в сучасному та ефективному веб-браузері.

## **Edge**

Microsoft Edge [18], спочатку відомий як Проєкт Spartan, є сучасним веб-браузером, розробленим компанією Microsoft. Цей браузер було випущено у 2015 році разом з операційною системою Windows 10. Edge функціонує незалежно від Internet Explorer і не є його заміною. Microsoft Edge, здатний працювати в системах, починаючи від Windows 7 та Windows Server 2008 R2, але він недоступний для Windows Vista та більш ранніх версій.

На початку 2023 року Microsoft оголосила про припинення підтримки Edge в Windows 7, 8 та 8.1. Це означає, що Edge 109 став останнім значущим оновленням браузера для цих операційних систем. Проте, компанія вирішила продовжувати випускати періодичні оновлення безпеки та критичні виправлення, щоб підтримувати безпеку користувачів, які продовжують використовувати ці старіші версії Windows.

## 1.2 Визначення ключових показників ефективності системи

Сучасні веб-додатки, завдяки своїм розширеним можливостям, можуть конкурувати з традиційними програмами, пропонуючи при цьому доступність з будь-якого пристрою та місцеположення - будь то комп'ютер, планшет чи мобільний телефон. Окрім того, вони часто виявляються економніше у вартості володіння та експлуатації. Ці характеристики роблять веб-технології ідеальними для вирішення різноманітних бізнес-задач. У зв'язку з цим, до них пред'являються високі вимоги до продуктивності. Для оптимізації продуктивності веб-додатків важливо зменшити обсяг переданих даних та кількість запитів до сервера. З ростом складності веб-додатків збільшується й ризик порушення їхньої функціональності [19]. В розділі розглядаються різні способи оптимізації швидкості завантаження веб-сторінок, що сприяють підвищенню продуктивності веб-додатків.

Продуктивність веб-додатків обмежується технічними параметрами середовища, в якому вони працюють: об'ємом оперативної пам'яті, швидкістю жорстких дисків і процесорів, пропускну здатністю комунікаційних каналів. Важливими показниками ефективності веб-додатків є кількість одночасно обслуговуваних користувачів і швидкість реакції системи на їх запити. Вплив цих факторів на продуктивність веб-додатків є очевидним.

Ефективність будь-якої інформаційної системи оцінюється шляхом зіставлення досягнутих результатів від її використання із загальними витратами на створення, експлуатацію та розвиток системи.

Показником, що характеризує підвищення якості управлінських процесів унаслідок впровадження нової інформаційної системи, є коефіцієнт поліпшення якості управління. Цей коефіцієнт вимірюється через зменшення часових затрат на збір, обробку, передачу інформації та розробку управлінських рішень порівняно з попереднім періодом. Важливо зауважити, що компетентність менеджерів і їхня здатність до аналізу залишаються незмінними до і після впровадження системи. Ефективна робота такої системи на об'єкті управління сприяє вдосконаленню



існуючих форм і методів управління, забезпечуючи більш ефективне виконання управлінських функцій.

### 1.2.1 Загальні функціональні можливості Dev Tools

Chrome DevTools [20] є незамінним набором інструментів для веб-розробників, інтегрованим у браузер Google Chrome. Вони дозволяють розробникам ефективно відлагоджувати код JavaScript, ставлячи точки зупинки, переглядаючи стек викликів, а також виконуючи код у реальному часі для виявлення і усунення помилок. Також можна перевіряти і редагувати HTML і CSS безпосередньо у браузері, що дозволяє відразу бачити зміни. Завдяки цьому можна експериментувати з оформленням сторінки і швидко знаходити оптимальні стилі.

Chrome DevTools також надає можливість аналізувати продуктивність веб-сторінки, виявляючи частини коду, які сповільнюють завантаження або виконання. Інструмент моніторингу мережі дозволяє перевірити, як швидко завантажуються ресурси сторінки, що є критично важливим для забезпечення кращого користувацького досвіду.

За допомогою інструменту для симуляції мобільних пристроїв розробники можуть перевірити, як їх веб-сайт або додаток виглядає та працює на різних пристроях і роздільних здатностях, що важливо в сучасному світі, де люди використовують безліч різних гаджетів для доступу до Інтернету. Крім того, Chrome DevTools надає доступ до консолі, де можна виконувати JavaScript-команди в реальному часі та переглядати повідомлення про помилки та логи, що значно спрощує процес розробки і відладки.

Розглянемо кожен вкладку Dev Tools для пошуку потреб під час тестування системи:

- *Elements* – Ця вкладка дозволяє інспектувати та змінювати HTML і CSS в реальному часі. Розробники можуть переглядати структуру DOM

(Document Object Model) сторінки, змінювати атрибути, контент елементів, а також експериментувати зі стилями CSS. Це допомагає швидко відлагоджувати проблеми з відображенням та макетуванням.

- *Console* – Ця вкладка використовується для перегляду діагностичних повідомлень та виконання JavaScript в реальному часі. Розробники можуть використовувати її для виконання коду, тестування функцій і перегляду помилок або попереджень.
- *Sources* – В цій вкладці можна переглядати, відлагоджувати та редагувати код JavaScript, що виконується на сторінці. Також можна встановлювати точки зупинки, відстежувати стек викликів та переглядати змінні. Це важлива вкладка для глибокого аналізу і відладки коду JavaScript.
- *Network* – Ця вкладка використовується для аналізу мережевої активності веб-сайту. Можна переглядати час завантаження кожного ресурсу (наприклад, зображень, скриптів, CSS-файлів), а також відслідковувати запити та відповіді HTTP. Це дозволяє ідентифікувати проблеми, пов'язані з продуктивністю, та оптимізувати завантаження ресурсів.
- *Performance* – У цій вкладці розробники можуть записувати та аналізувати продуктивність сторінки під час завантаження або виконання. Це дозволяє виявити уповільнення та оптимізувати час виконання скриптів та рендеринг сторінки.
- *Memory* – Тут можна перевіряти використання пам'яті веб-сторінкою та виявляти витіки пам'яті. Розробники можуть проаналізувати знімки пам'яті та профілі витрат пам'яті для виявлення неефективного використання ресурсів.
- *Application* – В цій вкладці можна переглядати інформацію про веб-сторінку, таку як збережені файли cookie, дані локального сховища, бази даних IndexedDB та інші ресурси веб-додатків. Це корисно для роботи з даними, які зберігаються локально в браузері.

- *Security* – Ця вкладка надає інформацію про безпеку веб-сайту, включаючи сертифікати SSL/TLS та інші параметри безпеки. Вона дозволяє перевіряти, чи є з'єднання з сайтом безпечним.
- *Lighthouse* – Це інтегрований інструмент для оцінки якості веб-сторінок з точки зору продуктивності, доступності, виконання кращих практик та SEO. Lighthouse генерує звіти з рекомендаціями щодо поліпшення веб-сторінки.

Оскільки в подальшому ми будемо працювати з ефективністю та продуктивністю роботи системи, нам підходить тестування завдяки вкладці Performance, яка надасть усю необхідну інформацію про систему.

### 1.2.2 Функціональні можливості вкладки Performance у Dev Tools

Вкладка Performance у Chrome DevTools дозволяє веб-розробникам профілювати веб-сторінку, виявляючи причини її низької продуктивності. Для ініціації процесу профілювання можна скористатися кнопками на верхній панелі (Рисунок 1.2): перша з них відразу розпочинає збір даних, тоді як друга спочатку перезавантажує сторінку, а потім починає збір. Зупинити збір інформації можна в будь-який момент, натиснувши кнопку Stop.

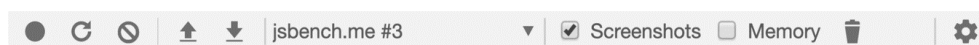


Рисунок 1.2 - кнопки верхньої панелі.

Крім того, на панелі інструментів є можливості для очищення зібраних даних та їх імпорту/експорту. Також можна вибирати зі списку доступних профілів для аналізу. Додаткові налаштування дозволяють візуалізувати чи приховати скриншоти сторінки й графік використання пам'яті, провести очищення пам'яті й налаштувати параметри збору інформації.

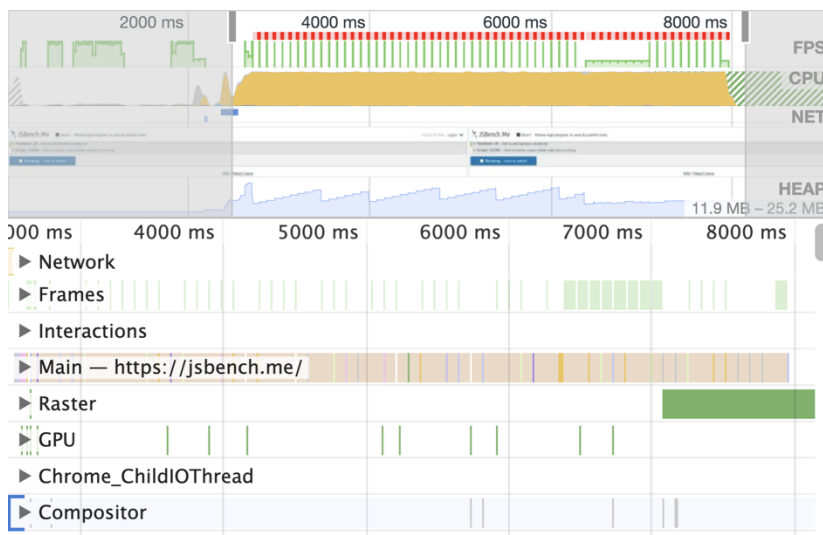


Рисунок 1.3 – Центральна частина вкладки

Після збору даних у центральній частині вкладки з'являється графік, на якому можна виділити певні часові проміжки для детальнішого аналізу (Рисунок 1.3). Інформація про виділені проміжки відображається під графіком у вигляді блоків, що містять детальну інформацію про процеси на сторінці, такі як діаграми виконання коду, записи взаємодії користувача зі сторінкою, мережні запити та інше.

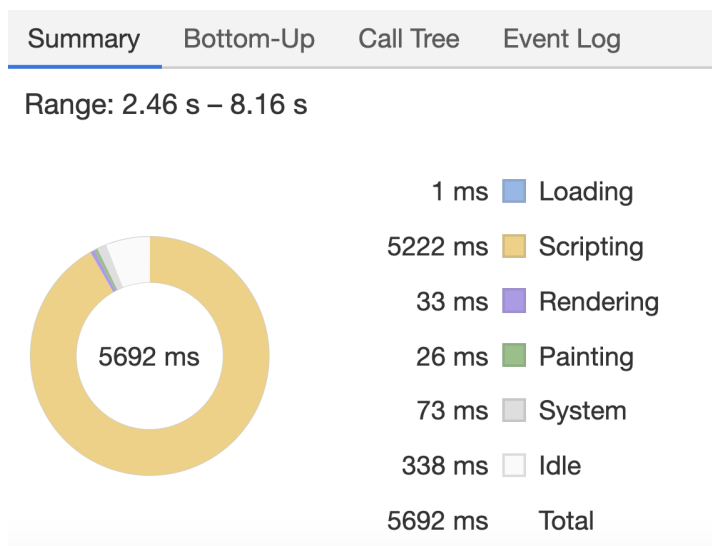


Рисунок 1.4 – Додаткові вкладки

В нижній частині вікна інструментів розробника розташовані додаткові вкладки (Рисунок 1.4): Summary, Bottom-up, Call Tree і Event Log. Кожна з них пропонує свій погляд на зібрані дані. Наприклад, вкладка Summary показує легенду до кольорів графіка та секторну діаграму часу виконання за вибраний проміжок часу. Вкладка Bottom-up сортує викликані функції за часом виконання, дозволяючи

швидко знайти код, який виконується повільно. Вкладка Call Tree відображає загальний час виконання всіх дій та дозволяє аналізувати ціле дерево викликів. Вкладка Event Log містить хронологічний список всіх подій, зафіксованих під час вибраного проміжку часу.

### 1.2.3 Ключові показники ефективності для системи вивчення технологій з веб-розробки

Виходячи з розглянутого функціоналу вкладки Performance, ключовими показниками ефективності для системи вивчення технологій з веб-розробки будуть наступні показники:

- *Швидкість обробки даних* – швидкість обробки даних залежить від багатьох факторів, починаючи від апаратного забезпечення пристрою, закінчуючи складністю обробки коду системи. Саме тому, необхідно враховувати швидкість обробки даних та не ускладнювати роботу на стороні клієнта, тобто на стороні Front-End. Це дозволить мати кращою продуктивність системи на будь якому пристрої.
- *Швидкість рендерингу і відрисовки* – немало важно враховувати, що кожна зміна на сторінці змушує браузер вирисовувати сторінку заново та рендерити її. Саме тому, код системи має бути оптимізованим та не перенавантажувати сторінку великою кількістю задач на рендеринг.
- *Навантаження CPU* – головне навантаження на обробку даних йде на CPU, та оскільки не усі девайси мають велику потужність на обробку великої кількості даних, то краще за все по максимуму перенести обробку даних на сторону back-end, щоб взаємодія користувача з сервісом була легкою та комфортною.
- *Блокери коду* – це часта проблема розробників, оскільки випадково можна написати код який буде працювати синхронно та блокувати

подальші рухи, оскільки перший ще не виконався. У разі будь якої помилки, це може привести до незбереженню даних користувача та змушенню проходження тестування ще раз. Але результат від 2-го разу вже буде не той як спочатку, тому тестування буде не точним.

### **1.3 Архітектура системи вивчення технологій з веб-розробки**

Для кращого тестування та винаходження мінімально оптимального апаратного та програмного забезпечення, необхідно у першу чергу ознайомитись із системою вивчення технологій з веб-розробки. Це дозволить краще розуміти послідовність дій системи при кожному кроці, та винайти недоліки у поточній реалізації системи.

#### **1.3.1 Базові та сучасні технології з веб-розробки**

Глобальна історія веб-розробки розпочалася близько тридцяти років тому, конкретно 6 серпня 1991 року. В цей день відбулася прем'єра першого веб-сайту, який був запущений організацією «World Wide Web Consortium» [21] і який досі можна відвідати за його оригінальною адресою:

<http://info.cern.ch/hypertext/WWW/TheProject.html>

При аналізі коду цього першого веб-сайту стає зрозуміло, що він суттєво відрізняється від сучасних стандартів веб-розробки. Незважаючи на це, він коректно відображається у всіх сучасних браузерів. Це демонструє одночасно і сильну сторону, і слабкість веб-розробки: код має виглядати однаково в будь-якому браузері, і це правило залишається актуальним незалежно від того, коли веб-сайт був створений.

## 1) Базові технології

Початок ери веб-розробки бере свій старт із "основоположних" технологій, які і досі широко використовуються мільйонами розробників по всьому світу: HTML, CSS та JavaScript..

### *HyperText Markup Language*

*HyperText Markup Language* (аббревіатура *HTML*) [22] - це стандартизована мова розмітки документів, призначена для перегляду веб-сторінок у браузерях. HTML дозволяє браузерам отримувати документи з сервера або локального сховища за допомогою протоколів HTTP/HTTPS та інтерпретувати їх код у веб-інтерфейс, який користувач може переглядати на веб-сайті.

HTML був вперше використаний та представлений публічно в першому веб-сайті, згаданому в інтродукції цього розділу, і з тих пір пройшов п'ять головних версій. Остання версія, відома як "HTML5", була опублікована 5 квітня 2008 року.

*HTML* - це мова розмітки документів, яка ґрунтується на використанні тегів. HTML-документи будуються з елементів, які відзначаються початковими та кінцевими тегами. Крім того, ці елементи можуть мати атрибути, які визначають їхні властивості..

### *Cascading Style Sheets*

*Cascading Style Sheets* (*CSS*) [23] - це формальна мова, яка використовується для опису зовнішнього вигляду документа, або, іншими словами, веб-сторінки, яка створена з використанням мови розмітки HTML.

Термін CSS був запропонований Хоконом Лі в 1994 році, і в середині 1990-х років він викликав інтерес у Консорціумі Всесвітньої павутини "World Wide Web". Це призвело до видання рекомендації "CSS1" у грудні 1996 року. Протягом свого існування CSS пройшов три основні версії, а остання з них відома як "CSS3". З 29 вересня 2011 року розробляється версія "CSS4", яка має внести багато нововведень.

CSS дозволяє змінювати зовнішній вигляд елементів HTML за допомогою селекторів, які є описом елемента або групи елементів, і надавати їм властивості та власні значення для надання стилів цим елементам.

### **JavaScript**

*JavaScript* [24] - це мультипарадигмова мова програмування, яка підтримує об'єктно-орієнтований (ООП), імперативний і функціональний стилі програмування. Вона є реалізацією специфікації ECMAScript.

Мова JavaScript була створена Бренданом Ейхом за десять днів, і спочатку її назвали Mocha, а потім LiveScript. Однак 4 грудня 1995 року вона була перейменована в JavaScript.

Спочатку JavaScript був розроблений як мова програмування для веб-браузерів і використовувався для надання інтерактивності веб-сторінкам. Проте в сучасному світі JavaScript набув величезних можливостей і дозволяє програмувати не лише на стороні клієнта (Front-End), але і на стороні сервера (Back-End), в мобільних та десктопних додатках.

## **2) Сучасні технології**

Протягом більше ніж 30 років розвитку веб-розробки з'явилася значна кількість сучасних технологій, які сприяють прискоренню та полегшенню процесу написання коду, роблять його більш зручним та дозволяють компілювати його в стандартні мови - HTML, CSS та JavaScript. З цією метою були розроблені різні інструменти та підходи.

Наприклад, препроцесори, такі як Pug і Jade, дозволяють писати HTML-код у більш компактній та зручній формі, яка потім перетворюється в стандартний HTML. Аналогічно, препроцесори стилів, такі як SASS, SCSS і LESS, дозволяють розробникам писати CSS з використанням змінних та інших зручних функцій, які після компіляції перетворюються в звичайний CSS.

У додаток до цього, існують різні CSS-фреймворки [25], такі як Bootstrap, Materialize і Tailwind CSS, які надають готові компоненти та стилізацію, що



дозволяє розробникам швидко створювати веб-сторінки без великого обсягу власного коду.

Ці інструменти та технології роблять веб-розробку більш продуктивною та зручною для розробників, допомагають зекономити час та ресурси при створенні веб-додатків і сайтів.

Несперечно, JavaScript розвинувся в широкий спектр технологій та інструментів, що використовуються для різних цілей у веб-розробці. Ось кілька з них:

- 1) *jQuery*: jQuery був створений для полегшення взаємодії з HTML-документами та обробки подій. Він надає простий та зручний інтерфейс для виконання багатьох завдань.
- 2) *Фреймворки Front-End*: Різні фреймворки, такі як React, Vue та Angular, розроблені для створення високоефективних та динамічних веб-додатків, включаючи односторінкові додатки (SPA). Вони спрощують розробку та підтримку великих веб-проектів.
- 3) *Node.js*: Node.js є середовищем виконання JavaScript на стороні сервера. Він дозволяє розробникам створювати Back-End для веб-сайтів та додатків за допомогою JavaScript.
- 4) *React Native*: Це фреймворк для створення мобільних додатків за допомогою JavaScript та React. Він дозволяє розробникам створювати мобільні додатки для різних платформ з використанням загального коду.
- 5) *Electron*: Electron - це платформа для створення десктопних додатків за допомогою веб-технологій, включаючи HTML, CSS та JavaScript. Вона дозволяє створювати додатки для різних операційних систем.

Ці технології розширюють можливості JavaScript та дозволяють розробникам створювати різноманітні веб-додатки та додатки для різних платформ, збільшуючи продуктивність та швидкість розробки.

Так, виправно вказано, що базові технології веб-розробки, такі як HTML, CSS та JavaScript, є фундаментальними та ключовими складовими для будь-якого веб-розробника. Незважаючи на появу нових технологій та інструментів, ці базові мови

залишаються важливими, оскільки вони є основою для створення веб-сторінок та додатків.

HTML відповідає за структуру та розмітку вмісту веб-сторінок. CSS відповідає за зовнішній вигляд та стилізацію цього вмісту. JavaScript, у свою чергу, додає інтерактивність та функціональність до веб-сторінок.

Основи цих мов є необхідними для розуміння та ефективної роботи з будь-якими іншими технологіями веб-розробки. Навіть при використанні фреймворків та інших вищих рівнів абстракції, розробник повинен знати основи, щоб ефективно працювати з кодом та вирішувати проблеми.

Отже, вивчення та засвоєння базових технологій веб-розробки завжди залишатиметься актуальним та корисним для тих, хто бажає працювати в цій галузі.

### 1.3.2 Математичний огляд рішення

#### Математичний опис задачі

Нехай,  $K$  представляє собою набір знань у певній області, а  $T$ ,  $T = \{T_1, \dots, T_n\}$  - набір завдань, які вимагають використання цих знань та навичок на практиці в певній області. Коли людина має набір знань  $K$  і розв'язує завдання  $T$ , вона може використовувати свої знання та навички для успішного вирішення цих завдань.

Зв'язок між набором знань і завданнями полягає в тому, що наявність правильних знань може сприяти успішному вирішенню завдань. Однак успіх у вирішенні завдань також залежить від того, як добре людина може застосовувати ці знання на практиці, її навичок і вмінь.

Припустимо, що є дві карти  $f$  і  $g$  такі, що:

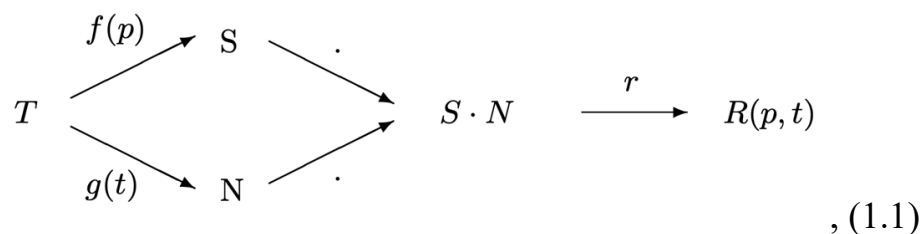
- $f(p): T \rightarrow S$ ,
- $g(t): T \rightarrow N$

Де:

- $S$  і  $N$  - множини дійсних невід'ємних та натуральних чисел відповідно;

- $p$  (людина) та  $t$  (час) - параметри карт.

Мета полягає у отриманні системи рекомендацій  $R(p, t) = r(f(T)(p) \cdot g(T)(t))$ , яка базується на параметрах  $p$  та  $t$ , де  $r$  - представляє кореляцію між  $T$  і набором даних, що включає ці знання, а також набором навичок та вмінь людини.



Формула 1.1 – Діаграма взаємозв'язку:  $T$  - це набір завдань,  $f(p)$ ,  $g(t)$  і  $r$  - карти,  $t$  і  $p$  - параметри, а  $R(p, t)$  - список рекомендацій.

У контексті цієї експертної системи, терміни, визначені раніше та описані в Формулі 1.1, мають наступні атрибути та характеристики:

- $K = \{ "html", "css", "javascript" \};$
- $T = 30$  (розмір розміру  $T$ );
- Карта  $f(p)$  - являє собою таке тестування, при якому для будь-якого параметра  $p$  та елемента  $T$  існує унікальне дійсне та не від'ємне значення;
- Карта  $g(t)$  - відображає оцінку  $T$  на основі даних, зібраних від 60 міжнародних ІТ-компаній протягом 12 років;
- Символ  $\cdot$  означає по елементне множення відповідного значення обох множин  $S$  і  $N$ . Очевидно, що ця операція є комутативною ( $S \cdot N = N \cdot S$ ).

### Вхідна інформація

Для полегшення процесу набуття знань користувачами, системи надання рекомендацій (СНР) впроваджує програмне забезпечення, яке включає тестування з використанням політомії. Це означає, що кожен тест пропонує користувачеві більше ніж два варіанти відповідей. Такий підхід сприяє економії часу користувача, оскільки він може швидко обрати один з запропонованих варіантів, а також

підвищує безпеку, запобігаючи введенню користувачем HTML-тегів, які можуть спричинити збої в системі.

### 1. Відповіді користувача

В процесі тестування користувач вибирає правильні відповіді на запитання і переходить до наступних, натискаючи кнопку "Next". Тим часом система аналізує відповідь користувача та інтегрує її до основної бази даних, що містить усі відповіді користувача.

Потім ці дані використовуються системою для розрахунку результатів тестування, формування рекомендацій та аналізу ринкових даних з огляду на рівень знань користувача.

### 2. Тести

Оскільки система рекомендацій базується на тестуванні, вона вимагає наявності тестів, які містять важливі дані, а саме:

- Використану технологію веб-розробки для створення тесту;
- Рівень складності питання;
- Формулювання питань;
- Набір варіантів відповідей;
- Визначену правильну відповідь;
- Коефіцієнти значимості кожної відповіді;
- Тематику питання;
- Посилання на рекомендовані матеріали за даною темою.

У процесі вибору тестів для системи було проведено глибокий аналіз ресурсів, що надають таку можливість. В результаті цього аналізу був виявлений канал "Задачі з Frontend" у месенджері "Telegram" [26], з яким було укладено угоду про використання їх тестів у власній СНР. Цей ресурс було обрано, оскільки він має аудиторію близько 10 тисяч фахівців у галузі

Front-End, з яких приблизно 2 тисячі активно беруть участь у тестах. Крім того, завдяки можливості аналізувати відгуки користувачів на кожне питання, цей ресурс дозволяє отримати готову аналітику кожного тесту, що сприяє проведенню власного аналізу, оцінці складності тестів та вибору тих, які найкраще підходять для системи.

### 3. Дані від «DOU»

Для аналізу даних ринку було вибрано та отримано доступ до опитувань від «DOU» [27]. Це використання дозволяє проводити детальний аналіз ринку, використовуючи інформацію від понад 20,000 спеціалістів, зібрану протягом останніх 12 років. Це дозволяє також проводити порівняння між цими даними та рівнем знань користувача, що може виявитися корисним для вдосконалення рекомендаційної системи.

Було звернено до компанії «DOU» із запитом на отримання доступу до їх даних з опитувань. Компанія «DOU» надала доступ на використання даних в ході розробки системи. Система ніяким чином не порушує права на використання даних з опитувань від компанії «DOU».

### **Вихідна інформація**

Оскільки основна мета цієї системи полягає у навчанні користувачів, вихідні дані системи, які базуються на вищезгаданих вхідних даних, повинні включати результати, отримані користувачем під час проходження тестування, а також аналіз його рівня знань у порівнянні з актуальними потребами ринку. Це дозволить не тільки оцінювати рівень знань користувача, але й забезпечити йому відповідні рекомендації для підвищення професійних компетенцій.

#### 1. Результат тестування

Після завершення тестування, користувачу важливо знати свій рівень знань. Для спрощення цього процесу вирішено подавати результати у

відсотковому виразі, оскільки це допоможе користувачу легше зрозуміти свої загальні досягнення. Результати будуть надані як за кожною окремою технологією, так і у вигляді загального підсумку тестування, що сприятиме кращому розумінню того, які аспекти потребують подальшого вивчення.

Кожна відповідь у тесті має свій коефіцієнт, який допомагає точніше визначити рівень знань користувача по кожній темі та технології. Завдяки аналізу відповідей користувача та врахуванню коефіцієнтів, система формує висновки та надає рекомендації, які поділяються на наступні категорії:

- Теми, які потрібно вивчити;
- Теми, які необхідно повторити;
- Теми, рекомендовані для вивчення.

Такий підхід дозволяє системі більш точно визначити рівень знань користувача, а самому користувачу - отримати зрозуміле уявлення про свій план подальшого навчання.

## 2. Аналіз ринку

Оскільки ця система зосереджена на оцінці рівня знань користувача та його подальшому навчанні, її завданням є допомогти користувачу зрозуміти, як його знання корелюють із потребами ринку. Використовуючи дані від «DOU»[1], система аналізує ці дані у порівнянні з рівнем знань користувача. Цей аналіз дозволяє системі визначити, чи кваліфікований користувач на посаду, таку як «Intern/Trainee Front-End» чи «Junior Front-End» [28].

Якщо рівень знань користувача недостатній для таких позицій, система надає рекомендації щодо технологій, які користувачеві потрібно додатково опанувати. Таким чином, користувач може отримати необхідні знання для претендування на ці або більш високі посади у майбутньому.

## Механізм складання рекомендацій

Для формування рекомендацій користувачам, система використовує три основних кроки, які допомагають у зрозумінні потреб користувача та аналізу його рівня знань у контексті ринкових вимог. Ці кроки включають:

- 1. Попереднє рішення частини тестування: На цьому етапі система аналізує відповіді користувача на тестові завдання, що дозволяє оцінити його знання та вміння.*
- 2. Попереднє рішення аналізу даних: Тут система здійснює аналіз зібраних даних, порівнюючи результати користувача з існуючою інформацією на ринку, щоб зрозуміти, на якому рівні знаходяться його компетенції.*
- 3. Коригування та складання рекомендацій: На останньому етапі, відомому як алгоритм складання рекомендацій, система на основі отриманих даних та аналізу формує індивідуальні рекомендації для користувача. Це включає вказівки на області, що потребують покращення, та поради щодо подальшого навчання.*

Такий підхід дозволяє системі ефективно визначити рівень знань користувача та надати йому відповідні рекомендації для підвищення його професійних навичок. Розглянемо усі кроки детальніше.

### 1. Попереднє рішення частини тестування

#### Коефіцієнти

Для визначення рівня знань, користувач проходить тестування, відповідаючи на ретельно підготовлені питання. Ці питання охоплюють ключові аспекти веб-розробки: від концепції веб-сайту до створення сторінок за допомогою HTML та CSS, а також програмування на JavaScript. Відповіді розроблені з використанням політомії, де кожна відповідь має свій коефіцієнт і класифікується за однією з чотирьох категорій: *правдива, майже правдива, майже хибна, хибна*. Така градація

допомагає краще оцінити рівень знань користувача по кожній технології.

У Формулі 1.1, набір  $S$  визначається таким чином:

- Коефіцієнт «1» відповідає «правдивій» відповіді;
- Коефіцієнт «0.8» – «майже правдивій» відповіді;
- Коефіцієнт «0.2» – «майже хибній» відповіді;
- Коефіцієнт «-1» – «хибній» відповіді;

У формулі 1.1, тестова частина позначена як  $f(p)$ , яка залежить від навичок та умінь користувача. Визначення  $f(T)(p) = \{f(T_1)(p), \dots, f(T_n)(p)\}$ , передбачає, що для будь-якого індексу  $i$ , відображення  $f(T_i)(p)$  приймає одне з чотирьох значень: 1, 0.8, 0.2, або -1.

### Варіація тестування

Система пропонує користувачам два різних формати тестування, залежно від їхніх потреб та часових обмежень:

- *Швидкий тест*: Цей тест складається з 10 запитань і розрахований на 5 хвилин. Він призначений для користувачів, які шукають швидке оцінювання своїх знань без глибокого занурення.
- *Повний тест*: У цьому варіанті тестування 30 запитань, на які потрібно відповісти протягом 15 хвилин. Повний тест дозволяє користувачам отримати більш детальний аналіз їхніх знань і навичок.

Обидва типи тестування спрямовані на ефективне визначення рівня знань користувача, але вони пропонують різні рівні глибини та деталізації оцінювання.

### Конфігурація тестування

Під час проходження повного тестування, користувач має можливість налаштувати тест, вибираючи технології та рівень



складності, які він хоче перевірити. Конфігурація тестування включає наступні категорії технологій:

- HTML;
- CSS;
- JavaScript.

Користувач також може вибрати рівень складності для тестування, який включає:

- Простий рівень;
- Середній рівень;
- Складний рівень.

Для того, щоб пройти повне тестування та отримати аналіз своїх знань у контексті ринкових вимог, користувачу необхідно вибрати усі категорії та рівні складності. В іншому випадку, тестування вважатиметься неповним, і користувач не отримає аналіз ринку після завершення тесту.

## 2. Попереднє рішення аналізу даних

Для проведення аналізу сучасного ринку були використані наступні дані, отримані від «DOU»:

- Рейтинг мов програмування;
- Запропоновані посади;
- Заробітна плата;
- Досвід роботи фахівців.

Згідно з Формулою 1.1 та зібраними даними, для кожного параметра  $T_j$ ,  $i = 1, \dots, n$  будується прогноз на наступні 12 місяців. Далі, на основі цього аналізу формується рейтинг  $\{T_1, \dots, T_n\}$ . Цей рейтинг фактично є картою  $g(T_i)(t)$ , де  $t$  є часовим параметром. Результатом цього процесу є набір  $\{g(T_1)(t), \dots, g(T_n)(t)\}$ , де кожен елемент представляє собою натуральне число, відображаючи рейтинг  $T_i$  на основі аналізу отриманих даних.

Такий підхід дозволяє системі оцінити актуальний стан ринку та надати користувачам відповідні рекомендації, засновані на реальних даних і тенденціях у сфері ІТ.

### 3. Алгоритм складання рекомендацій

На підставі результатів перших двох кроків алгоритму складання рекомендацій, система має два набори чисел:  $\{f(T_1)(p), \dots, f(T_n)(p)\}$  від тестування користувача і  $\{g(T_1)(t), \dots, g(T_n)(t)\}$  від аналізу ринку. На основі цих даних формується список рекомендацій  $R(p, t) = \{R_1, R_2, \dots, R_l\}$ , де  $l \leq n$ , за формулою 1.1:

$$R_{i+1} = r(\arg(\min\{\{f(T_1)(p) \cdot g(T_1)(t), \dots, f(T_n)(p) \cdot g(T_n)(t)\} \setminus r^{-1}(R_i)\})), \quad (1.2)$$

*Формула 1.2 – Діаграма взаємозв'язку включає  $T$  - набір завдань,  $f(p)$ ,  $g(t)$  і  $r$  - карти,  $t$  і  $p$  - параметри, а  $R(p, t)$  - список рекомендацій.*

, де: карта  $r$  визначає відповідність між множинами  $T$  та  $R$  як списком у базі знань, з  $R_0 = \emptyset$ ,  $r^{-1}(R_0) = \emptyset$ ,  $i = 1, \dots, l$  та  $l \leq n$ .

Важливо зауважити, що  $l$  може бути обрано як  $0,5n$  для оптимізації навантаження на систему рекомендацій.

Додатково, система може надавати рекомендації щодо знань, які потрібно активно вивчати, та тем, які вимагають освіження в пам'яті користувача.

Далі, ми розглянемо ключові властивості  $f(p)$ ,  $g(t)$  і  $r$ , що представлені у Формулі 1.1. Нагадуємо, що  $T$ ,  $N$ ,  $S$  і  $R$  - це кінцеві множини, які представляють собою набори завдань, рейтинговий список, результати тестування та список рекомендацій відповідно.

Для демонстрації специфіки та часових тенденцій (трендів ринку ІТ) системи будуть сформульовані та доведені деякі твердження.

**Лема 1** стверджує, що для будь-якого заданого параметра  $p$  завжди існує таке додатне значення  $\varepsilon$ , що списки рекомендацій  $R(p, t)$  і  $R(p, t + \varepsilon)$  не будуть

ідентичними. Це вказує на те, що рекомендації змінюються у часі, навіть при сталому  $p$ . Давайте розглянемо доказ цієї леми.

**Доказ:** Беремо довільне значення параметра  $p$  і припускаємо, що для двох моментів часу  $t_1$  і  $t_2$  ми маємо два різних списки рекомендацій  $R_1(p, t_1)$  та  $R_2(p, t_2)$ .

1. Якщо  $l_1 \neq l_2$ : Очевидно, списки різні, і лема доведена,  $\varepsilon = |t_2 - t_1|$ .
2. У випадку  $l_1 = l_2$ , маємо дві можливості:
  - *Випадок 1:* Існують такі  $i$  та  $j$ , що  $R_{li}^1 \neq R_{lj}^2$ . Це означає, що списки рекомендацій різні, навіть якщо їх довжина однакова, лема доведена, і  $\varepsilon = |t_2 - t_1|$ .
  - *Випадок 2:* Для всіх  $i$  виконується  $R_{li}^1 = R_{li}^2$ . Використовуючи Формулу 1.2, отримуємо новий список  $R_3(p, t_3)$ . Якщо для всіх  $i$  виконується  $R_{li}^1 = R_{li}^2 = R_{li}^3$ , продовжуємо процес, доки не знайдемо такі  $s$  та  $t_s$ , що  $R_{li}^1 = \dots = R_{li}^{s-1} \neq R_{li}^s$ . В такому випадку, лема доведена і  $\varepsilon = |t_1 - t_s|$ .

Оскільки карта  $g(t)$  не є константою і змінюється у часі, завжди можна знайти такий момент часу  $t_3$ , при якому  $g(t) \neq g(t_3), i t \neq t_3$ . Це підтверджує, що з плином часу рекомендації системи змінюються, що відображає динаміку та адаптацію системи до змін у знаннях користувача та умовах ринку.

**Лема 2** стверджує, що для фіксованого моменту часу  $t$  і двох різних параметрів  $p_1$  та  $p_2$ , або списки рекомендацій  $R(p_1, t)$  і  $R(p_2, t)$  будуть різними, або ідентичними. Ця лема демонструє, як система реагує на зміни параметрів, які визначають індивідуальні властивості та уміння користувачів. Давайте розглянемо доказ цієї леми.

**Доказ:** Розглянемо два списки рекомендацій  $R_1(p_1, t) = \{R_1, \dots, R_{l_1}^1\}$  та  $R_2(p_2, t) = \{R_1^2, \dots, R_{l_2}^2\}$  для  $p_1$  та  $p_2$  у момент часу  $t$ .

1. Припускаємо, що  $f_1(t) = f_2(t)$ , де  $f_1(t)$  і  $f_2(t)$  відповідають  $p_1$  і  $p_2$  відповідно.
2. За Формулою 1.1 ми маємо:
  - $R_0^1 = R_0^2 = \emptyset$ ;

- Для будь-якого  $i$ , де  $i = 1, \dots, \min \{l_1, l_2\}$ , або  $R_i^1 = R_i^2$ , або  $R^1 \neq R^2$ .  
Якщо  $R_i^1 = R_i^2$ , то згідно з Формулою 1.2, що  $g(T_i)(p_1) = g(T_i)(p_2)$ .
  - Якщо  $l_1 \neq l_2$ , то існує таке  $i$ , що  $\min \{l_1, l_2\} < i < \max \{l_1, l_2\}$ , і або  $R_i^1 = \emptyset$  та  $R_i^2 \neq \emptyset$  або  $R_i^1 \neq \emptyset$  та  $R_i^2 = \emptyset$ .
3. Якщо для часу  $t$   $R(p_1, t) = R(p_2, t)$ , це означає, що з точки зору IT-ринку в цей момент часу  $p_1$  і  $p_2$  є ідентичними.
  4. Якщо для часу  $t$   $R(p_1, t) \neq R(p_2, t)$ , то можливо, вони стануть однаковими у майбутньому моменті  $t + \varepsilon$ .

Ця лема важлива для розуміння того, як система відрізняє індивідуальні властивості та потреби користувачів. Якщо два користувача мають різні рекомендації при одному й тому ж часі  $t$ , це свідчить про різницю в їхніх знаннях та навичках. Цей тип ідентичності в рекомендаціях може бути розглянутий як своєрідна форма суперництва між користувачами.

### Механізм прогнозу даних

Поліноміальна регресія 2-го ступеня [29] використовується для прогнозування зміни заробітної плати фахівців на наступний рік. Цей метод є ефективним, коли звичайна лінійна регресія не може адекватно описати відносини між даними через їх нелінійність. Ось ключові аспекти поліноміальної регресії:

1. *Перетворення вихідних ознак:* У поліноміальній регресії, вихідні ознаки перетворюються на поліноми заданого ступеня (у цьому випадку, 2-го ступеня). Це дозволяє моделі краще відображати складні залежності між даними.
2. *Моделювання за допомогою лінійної моделі:* Після перетворення, дані моделюються за допомогою лінійної моделі. Це означає, що попри нелінійність вхідних даних, сама модель залишається лінійною щодо нових поліноміальних ознак.

Необхідність поліноміальної регресії впливає з таких моментів:

- *Недоліки лінійної моделі для нелінійних даних:* Лінійна модель може ефективно працювати з лінійними даними, але коли дані мають нелінійну структуру, лінійна модель часто дає невисоку точність прогнозів. У таких випадках, поліноміальна регресія може значно покращити якість моделювання.
- *Необхідність для нелінійно розташованих даних:* Коли точки даних розташовані нелінійно, поліноміальна регресія стає необхідною, оскільки вона дозволяє краще відтворити складні зв'язки між ознаками та цільовою змінною.

Застосування поліноміальної регресії у цьому контексті вказує на прагнення до більш точного та надійного прогнозування економічних показників, таких як заробітна плата.

*Основна мета регресійного аналізу* - визначити залежність змінної  $y$  від однієї або декількох незалежних змінних  $x$ .

Моделлю простої лінійної регресії є:

$$y = \beta_0 + \beta_1 x + \varepsilon, (1.3)$$

Якщо кількість елементів більше, то тоді краще підходить використання моделі квадратичної моделі

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon, (1.4)$$

Також, якщо кількість елементів збільшується, то можна отримати очікуване значення ступеню  $n$ , використовуючи загальну модель поліноміальної регресії:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \dots + \beta_n x^n + \varepsilon, (1.5)$$

Модель поліноміальної регресії представлена як:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \dots + \beta_m x_i^m + \varepsilon, (1.6)$$

Дана модель може бути також представлена у матричній формі через матрицю проектування  $X$ , вектор відповіді  $\vec{y}$ , вектор параметрів  $\vec{\beta}$  та вектор випадкових помилок  $\vec{\varepsilon}$ . Модель можна представити у вигляді:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ 1 & x_3 & x_3^2 & \cdots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}, \quad (1.7)$$

Що при використанні чисто матричної нотації записується як:

$$\vec{y} = X\vec{\beta} + \vec{\varepsilon}, \quad (1.8)$$

Вектор розрахункових коефіцієнтів поліноміальної регресії становить:

$$\hat{\vec{\beta}} = (X^T X)^{-1} X^T \vec{y}, \quad (1.9)$$

Припускаючи що  $m < n$  щоб матриця була зворотною.

Ці моделі використовуються для аналізу та прогнозування залежностей у різних областях, включаючи економіку, науку, інженерію та соціальні науки. Поліноміальна регресія зокрема корисна, коли відносини між змінними є нелінійними, дозволяючи створювати більш точні прогнози.

### 1.3.3 Сучасна теорія тестування

Сучасна теорія тестування (IRT), яка також відома як Item Response Theory, є потужним інструментом в області психометрики [30] та освітнього оцінювання [31]. Ця теорія надає більш глибокий і точний аналіз результатів тестувань порівняно з традиційними методами [32]. Ось ключові аспекти IRT:

- *Оцінка ймовірності правильної відповіді:* IRT оцінює, як імовірність правильної відповіді залежить від рівня здібностей або рис особистості піддослідного.
- *Виключення неінформативних питань:* Одним з застосувань IRT є ідентифікація та видалення питань з тесту, які не надають значущої інформації про здібності або навички особи.

- *Аналіз латентних конструктів:* IRT дозволяє оцінити взаємозв'язки між латентними (непостережуваними) характеристиками, такими як здібності або вміння, і спостережуваними відповідями на завдання.
- *Оптимізація пред'явлення завдань:* За допомогою IRT можна оптимізувати набір завдань, які пред'являються респондентам, забезпечуючи, що кожне завдання є відповідним до рівня здібностей особи.
- *Встановлення зв'язку між здібностями та результатами тесту:* IRT дозволяє оцінити, як особливості піддослідного впливають на його здатність вирішувати завдання різного рівня складності.

Використання IRT у розробці та аналізі тестів є особливо корисним, оскільки ця теорія враховує індивідуальні відмінності між піддослідними та їхніми відповідями на конкретні завдання. Це дозволяє створювати більш чутливі та адаптивні оцінювальні інструменти.

Переваги Сучасної теорії тестування (СТТ) дійсно значні, особливо у контексті точності та інваріантності вимірювань. Ось детальніший огляд цих переваг:

1. *Специфічність до вибірки тестів:* В IRT кожен тестовий завдання (item) оцінюється індивідуально, а його характеристики не залежать від конкретного набору завдань у тесті. Це означає, що характеристики тестового завдання (наприклад, його складність) можуть бути застосовані універсально, незалежно від того, в якій комбінації тестових завдань вони використовуються.
2. *Інваріантність дескрипторів тестових завдань:* Параметри тестових завдань у IRT, такі як індекси складності та відмінності, не залежать від конкретної групи піддослідних, які проходять тест. Це робить результати IRT стабільними та надійними, навіть якщо склад тестуючих змінюється.
3. *Точна статистика:* IRT забезпечує високу точність оцінок здібностей та характеристик завдань. Це дозволяє отримувати більш точні та об'єктивні висновки про рівень знань та навичок піддослідних.

Мета IRT полягає у створенні інваріантної статистики завдань та точних оцінок здібностей осіб. Це означає, що IRT дозволяє розробляти тестові завдання та оцінювати реакції піддослідних на ці завдання таким чином, що ці показники залишаються стабільними та надійними незалежно від особливостей конкретної вибірки піддослідних або специфіки тестового набору.

## **Моделі IRT**

### 1. Передісторія моделей

Передісторія та основні характеристики Сучасної теорії тестування (СТТ або IRT) глибоко залучені у психометрику та освітнє оцінювання. Ось ключові аспекти цієї теорії:

1. *Зв'язок між латентними ознаками та відповідями* – СТТ описує взаємозв'язок між латентними ознаками (непостережуваними характеристиками, такими як здібності або вміння), властивостями пунктів шкали (тестових завдань) та відповідями респондентів.
2. *Прогнозування ефективності респондента* – Це можливо зробити, визначивши характеристики респондентів (риси чи здібності), оцінивши їх за цими характеристиками (бали здібностей), і використовуючи ці бали для прогнозування або пояснення результатів виконання завдань та тестів.
3. *Характеристики IRT:*
  - Поведінка респондента може бути пояснена або передбачена на основі однієї чи декількох характеристик, що називаються рисами.
  - Існує взаємозв'язок між виконанням завдань респондентами та латентними рисами або здібностями, які лежать в основі виконання тестування.
  - Ефективні моделі IRT забезпечують засоби для оцінювання респондентів за цими характеристиками.



- Риси повинні оцінюватися на основі спостережуваних результатів респондента за набором тестових завдань.
4. *Застосування СТТ у навчальному тестуванні* – Основне застосування СТТ - у сфері освіти, де вона використовується для аналітичного порівняння здібностей респондента із їхніми результатами тестування. Це дозволяє більш точно оцінити здібності учнів та студентів, а також ефективно відстежувати їхній прогрес.

Таким чином, IRT є важливим інструментом в освітньому оцінюванні, дозволяючи вчителям, лекторам та дослідникам більш точно зрозуміти та оцінити здібності учасників тестування.

## 2. Припущення моделі IRT

Основні припущення Сучасної теорії тестування (СТТ) включають одномірність, локальну незалежність та адекватну специфікацію моделі.

### Одномірність

Усі завдання у тесті вимірюють лише одну певну характеристику або здібність. Важливо, що в моделі СТТ для кожного респондента існує тільки одна змінна  $Y$ , що відображає цей вимір. Всі інші фактори, які можуть вплинути на відповіді на завдання, розглядаються як випадкові помилки або специфічні для завдання збурення, які є унікальними для кожного завдання і не спільними для інших.

Існує три загальноприйнятих методів для оцінювання одномірності тесту, що допомагають у визначенні, чи відповідає тест припущенню про одновимірність, яке є ключовим для валідності застосування СТТ:

- *Аналіз власних значень між елементами кореляційної матриці* – використовується для визначення, чи всі завдання тесту вимірюють одну й ту ж характеристику.

- *Критерій суттєвої одномірності Стаута* – використовує статистичні критерії для перевірки, чи є тест одновимірним.
- *Індекси, засновані на неув'язках від одновимірного рішення* – індекси допомагають визначити, наскільки добре одновимірна модель відповідає даним, порівняно з багатовимірними моделями.

### Локальна незалежність

Локальна незалежність у СТТ фокусується на залежності між різними елементами тесту. Вона передбачає, що відповіді на одне завдання тесту не впливають на відповіді на інше завдання після того, як враховано рівень здібностей респондента, позначеного як  $\theta$ . Важливим є те, що хоча відповіді на окремі завдання мають бути незалежними після урахування  $\theta$ , вони можуть демонструвати певну кореляцію на рівні всієї вибірки респондентів. Це означає, що припущення про незалежність елементів можливе лише після аналізу здібностей. Локальна незалежність є ключовою для забезпечення точності оцінювання в СТТ, оскільки вона допомагає уникнути спотворень у взаємодії між завданнями.

### Правильна специфікація моделі

Наприклад, при використанні моделі з одним параметром рівня логістичної (1PL) моделі на даних, які в дійсності краще описуються моделлю з різними нахилами або нульовою нижньою асимптотою, може виникнути проблема несумісності. Багато тестових елементів у такому випадку не будуть адекватно описані моделлю 1PL.

Крім того, якщо функція відповіді на завдання не є монотонно зростаючою, жодна з поширених моделей IRT не буде підходити. У таких випадках, важливим показником для адаптації тестових елементів є кореляція між спостережуваними відповідями та відповідями, передбаченими моделлю. Ця кореляція залишку або

різниці між спостережуваними та передбаченими пропорціями відповідей важлива для оцінювання адекватності моделі. Важливо, що цей залишок розраховується для груп респондентів з приблизно однаковим рівнем здібностей  $\theta$ , що дозволяє більш точно визначити, наскільки добре модель відповідає даним.

### 3. Моделі IRT для дихотомічних елементів

Моделі СТТ для дихотомічних елементів описують взаємодію між особами, що мають різні латентні риси, та пунктами тесту, які можуть бути призначені для вимірювання більше ніж однієї риси. В таких моделях враховується:

- $n$  – кількість респондентів;
- $i$  – індивідуальні питання;
- $j$  – відповіді кожного респондента;
- $Y_{ij}$  – відповіді респондентів, та розглядаються як випадкові змінні, що відображають реакцію респондента  $j$  на питання  $i$ ;
- $\Omega_Y$  – набір можливих значень у який входить відповіді респондентів  $Y_{ij}$  та вважається однаковим для всіх питань тесту;
- $\theta_j$  – латентні риси кожного респондента;
- $\eta_i$  – набір параметрів, які моделюють характеристики кожного питання.

Існує різноманіття моделей IRT, залежно від функціональних формул, що описують імовірність відповіді  $Y_{ij}$ , враховуючи ці значення:

$$P(Y_{ij} = y | \theta_j, \eta_i) = \phi(y | \theta_j, \eta_i); y \in \Omega_Y, (1.10)$$

Параметри питань  $\eta_i$  можуть включати в себе:

- 1)  $a_j$  – параметри дискримінативності, які показують, наскільки добре питання розрізняє між респондентами з різними рівнями латентної риси;

- 2)  $b_j$  – параметри складності, які визначають рівень здібностей, необхідний для правильної відповіді;
- 3)  $c_j$  – параметри вгадування, що вказують на імовірність вгадування правильної відповіді;
- 4)  $d_j$  – параметри похибки або необережності, які враховують можливу випадковість у відповідях.

Залежно від кількості цих параметрів, що входять до рівняння моделі, існують різні типи моделей СТТ. Кожен тип моделі використовується для різних цілей оцінювання та аналізу, в залежності від специфіки тесту та характеристик респондентів.

## Типи моделей IRT

### 1PL Модель Раша

1PL Модель Раша є базовою моделлю у IRT та характеризується наявністю одного параметра:

$$P(Y_{ij} = 1 | \theta_j, b_i) = \frac{\exp(\theta_j - b_i)}{1 + \exp(\theta_j - b_i)}, \quad (1.11)$$

Де:

$b_i$  ( $-\infty < b_i < \infty$ ) є параметром складності, який вказує на рівень складності завдання, тобто наскільки легким або складним є завдання.

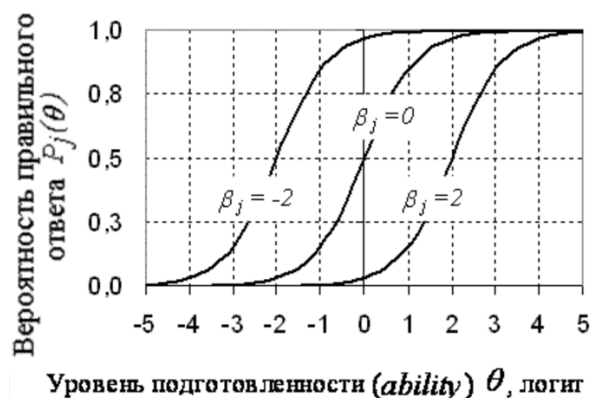


Рисунок 1.5 – Графік 1PL моделі Раша

З наведеного графіку 1PL моделі Раша на *Рисунку 1.5*, можна зробити спостереження що якщо рівень знань респондента  $\theta_j$  дорівнює 0, то імовірність правильної відповіді на

- легке завдання буде близькою до 100%,
- на завдання середньої складності - близькою до 50%,
- на складне завдання - близькою до 0%.

В рамках 1PL моделі Раша можна виділити два основні випадки:

- Якщо різниця між знаннями респондента  $\theta_j$  та складністю завдання  $b_i$  є позитивною (тобто  $\theta_j - b_i > 0$ ), респондент, швидше за все, відповідь правильно.
- Якщо ця різниця є негативною (тобто  $\theta_j - b_i < 0$ ), респондент, швидше за все, відповідь неправильно.

Ця модель є особливо корисною за своєю простотою та здатністю ефективно оцінити взаємодію між рівнем здібностей респондента та рівнем складності завдань.

### 2PL Модель Бірнбаума

2PL модель, яка також відома як Модель Бірнбаума, є другою моделлю IRT, що розширює основні концепції моделі Раша, включаючи два ключові параметри:

$$P(Y_{ij} = 1 | \theta_j, a_i, b_i) = \frac{\exp a_i(\theta_j - b_i)}{1 + \exp a_i(\theta_j - b_i)}, \quad (1.12)$$

Де:

$a_i$  ( $-\infty < a_i < \infty$ ) є параметром дискримінативності або нахилу, який показує, наскільки швидко змінюється імовірність правильної відповіді зі зміною рівня здібностей респондента  $\theta_j$ . Цей параметр визначає, наскільки ефективно завдання розрізняє між респондентами з різними рівнями здібностей.

$b_i$  ( $-\infty < b_i < \infty$ ) є параметром складності, який вказує на рівень складності завдання, тобто наскільки легким або складним є завдання.

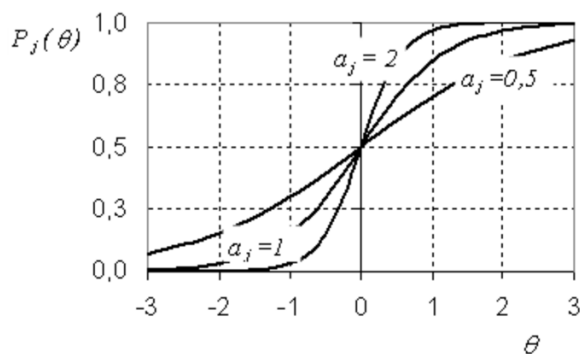


Рисунок 1.6 – Графік 2PL моделі Бірнбаума

Аналізуючи графік 2PL моделі Бірнбаума на Рисунку 1.6, можна спостерігати, що зі збільшенням складності завдання крива стає крутішою, а імовірність правильної відповіді респондента зменшується. Це вказує на те, що із зростанням складності завдання потрібен вищий рівень здібностей для його успішного вирішення.

2PL модель Бірнбаума забезпечує більш гнучке та детальне розуміння взаємодії між рівнем здібностей респондента та характеристиками завдань, оскільки вона дозволяє оцінити не тільки рівень складності, а й чутливість кожного завдання до різних рівнів здібностей респондентів.

### 3PL Модель Бірнбаума

3PL модель Бірнбаума є третьою моделлю IRT, та включає в себе три ключові параметри:

$$P(Y_{ij} = 1 | \theta_j, a_i, b_i, c_i) = c_i + (1 - c_i) \frac{\exp a_i(\theta_j - b_i)}{1 + \exp a_i(\theta_j - b_i)}, \quad (1.13)$$

Де:

$a_i$  ( $-\infty < a_i < \infty$ ) є параметром дискримінативності або нахилу, який показує, наскільки швидко змінюється імовірність правильної відповіді зі зміною рівня здібностей респондента  $\theta_j$ . Цей параметр визначає, наскільки ефективно завдання розрізняє між респондентами з різними рівнями здібностей.

$b_i$  ( $-\infty < b_i < \infty$ ) є параметром складності, який вказує на рівень складності завдання, тобто наскільки легким або складним є завдання.

$c_i$  ( $0 \leq c_i \leq 1$ ) є параметром вгадування, який вказує на імовірність вгадування правильної відповіді респондентом, навіть якщо він не володіє необхідними знаннями або навичками.

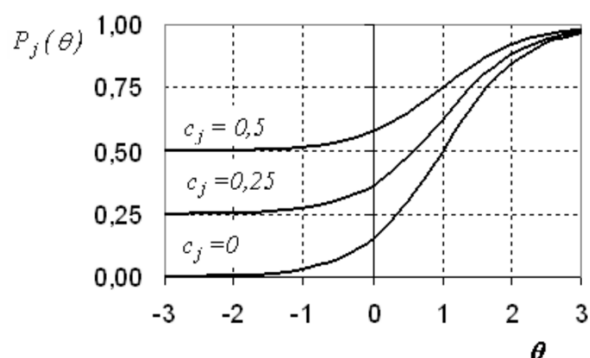


Рисунок 1.7 – Графік 3PL моделі Бірнбаума

На графіку цієї моделі (Рисунок 1.7) можна спостерігати, що із збільшенням параметра вгадування  $c_i$  зростає ймовірність того, що респондент зможе правильно вгадати відповідь, навіть не володіючи достатнім рівнем знань або навичок для цього завдання. Це особливо важливо в контексті оцінки складності завдань, оскільки надмірне вгадування може знизити надійність та валідність тесту. Наприклад, якщо параметр вгадування  $c_i$  перевищує 0.35, тестове завдання може вважатися неприйнятним, оскільки імовірність вгадування стає надто високою [33].

#### 4PL Модель Бірнбаума

4PL модель Бірнбаума, яка представляє собою ще більш розвинену модель у рамках Сучасної теорії тестування (IRT), включає чотири основні параметри. Ця модель описує ймовірність того, що респондент правильно відповість на питання, враховуючи його здібності та характеристики питання:

$$P(Y_{ij} = 1 | \theta_j, a_i, b_i, c_i, d_i) = c_i + (d_i - c_i) \frac{\exp a_i(\theta_j - b_i)}{1 + \exp a_i(\theta_j - b_i)}, \quad (1.14)$$

Де параметри моделі означають наступне:

$a_i$  ( $-\infty < a_i < \infty$ ) є параметром дискримінативності або нахилу, який показує, наскільки швидко змінюється імовірність правильної відповіді зі

зміною рівня здібностей респондента  $\theta_j$ . Цей параметр визначає, наскільки ефективно завдання розрізняє між респондентами з різними рівнями здібностей.

$b_i$  ( $-\infty < b_i < \infty$ ) є параметром складності, який вказує на рівень складності завдання, тобто наскільки легким або складним є завдання.

$c_i$  ( $0 \leq c_i \leq 1$ ) є параметром вгадування, який вказує на імовірність вгадування правильної відповіді респондентом, навіть якщо він не володіє необхідними знаннями або навичками.

$d_i$  ( $0 \leq c_i \leq d_i \leq 1$ ) є параметром похибки або необережності, який враховує імовірність помилкової або недбалої відповіді.

В контексті 4PL моделі Бірнбаума, ми бачимо, що якщо параметр вгадування  $c_i$  є низьким, існує підвищена ймовірність того, що респондент може зробити помилку або відповісти недбало. Цей аспект моделі важливий для оцінки якості тестових завдань, а також для точного розуміння поведінки респондентів під час тестування.

## Оцінка параметрів

### 1. Основна інформація

При використанні IRT необхідно здійснити два види оцінок:

- 1) Оцінка параметрів елемента;
- 2) Прогнозування індивідуальної латентної оцінки  $\theta$ .

Оцінка параметрів елемента включає в себе процеси:

- Оцінки граничної максимальної правдоподібності (MML) [34]
- Оцінки спільної максимальної правдоподібності (SML) [35].

У методі MML процес починається з припущення про розподіл латентної оцінки  $\theta$ , який зазвичай розглядається як стандартний нормальний розподіл.



Після оцінки параметрів елемента, наступним кроком є оцінка параметрів респондента. Це може бути зроблено за допомогою декількох підходів, включаючи:

- Максимальну правдоподібність;
- Байесовський підхід.

Байесовський підхід може використовувати різні методи для оцінки, такі як:

- Модальний апостеріорний (режим);
- Середнє значення апостеріорного розподілу;
- Модальний апостеріорний;
- Очікуваний апостеріорний.

Ці процеси оцінки є критичними для точного та ефективного застосування IRT, оскільки вони забезпечують основу для розуміння зв'язку між характеристиками тестових завдань та здібностями респондентів. Оцінка параметрів елемента дозволяє зрозуміти, як різні завдання відповідають на різні рівні здібностей, тоді як оцінка латентної оцінки  $\theta$  визначає, на якому рівні знаходиться кожен респондент щодо цих завдань.

#### Приклад, ситуація тестування навчання:

У даному прикладі ситуації тестування навчання ми маємо наступні елементи:

$n$  – Кількість респондентів;

$I$  – Кількість питань;

$i$  – Позначення конкретного питання;

$j$  – Позначення конкретного респондента;

$Y_{ij}$  – Випадкова змінна, яка відображає реакцію респондента  $j$  на питання  $i$ ;

$\Omega_Y$  – Набір можливих значень  $Y_{ij}$ , які вважаються однаковими для всіх питань тесту;

$\theta_j$  – Латентна (скрита) риса респондента  $j$ ;

$\eta_i$  – Набір параметрів для моделювання характеристик питань (елементів).

Ймовірність того, що респондент  $j$  відповість на питання  $i$ , може бути представлена як:

$$P(Y_{ij} = y_{ij} | \theta_j, \eta_i) = \phi(y_{ij} | \theta_j, \eta_i), \quad (1.15)$$

Де:

$y_j = (y_{1j}, y_{2j}, \dots, y_{ij}, \dots, y_{Ij})$  – вектор бінарних відповідей від  $j$ -го респондента, , що має здібності  $\theta_j$ .

Функція правдоподібності для окремого респондента  $j$  визначається як:

$$L(\theta_j, \vec{\eta} | y_{ij}) = P(y_{ij} | \theta_j, \vec{\eta}) = \prod_{i=1}^I P(y_{ij} | \theta_j, \vec{\eta})^{y_{ij}} [1 - P(y_{ij} | \theta_j, \vec{\eta})]^{1-y_{ij}}, \quad (1.16)$$

Де:

$\vec{\eta}$  – вектор параметрів елемента.

Функція правдоподібності для всіх респондентів:

$$L(\vec{\theta}, \vec{\eta} | y_{ij}) = \prod_{j=1}^n \prod_{i=1}^I P(y_{ij} | \theta_j, \vec{\eta})^{y_{ij}} [1 - P(y_{ij} | \theta_j, \vec{\eta})]^{1-y_{ij}}, \quad (1.17)$$

Повна логарифмічна правдоподібність для  $n$  респондентів:

$$l(\vec{\theta}, \vec{\eta} | y_{ij}) = \sum_{j=1}^n \sum_{i=1}^I y_{ij} \log(P(y_{ij} | \theta_j, \vec{\eta})) + (1 - y_{ij}) \log(1 - P(y_{ij} | \theta_j, \vec{\eta})), \quad (1.18)$$

Метод граничної максимальної правдоподібності для оцінки параметрів елемента використовує граничний розподіл повної логарифмічної ймовірності параметрів елемента, отриманого через інтеграцію за  $\theta$ .

## 2. Оцінка параметрів елемента

Логарифм правдоподібності маргінального розподілу параметрів елемента у контексті IRT із врахуванням форми розподілу  $g(\theta_j)$  для незалежних та однаково розподілених латентних ознак визначається за формулою:

$$l(\vec{\eta}) = \sum_{j=1}^n \sum_{i=1}^I \log \int \left[ y_{ij} \log \left( P(y_{ij} | \theta, \vec{\eta}) \right) + (1 - y_{ij}) \log \left( 1 - P(y_{ij} | \theta, \vec{\eta}) \right) \right] g(\theta_j) d(\theta_j), \quad (1.19)$$

Ця формула використовується для оцінки параметрів елемента, враховуючи латентні характеристики респондентів. У цьому виразі:

$\vec{\eta}$  – представляє вектор параметрів елемента.

$y_{ij}$  – відповідь респондента  $j$  на питання  $i$ .

$\theta$  – латентна ознака респондента.

$g(\theta_j)$  – функція розподілу латентних ознак, яка вказує, як часто зустрічається кожен рівень латентної ознаки серед респондентів.

Інтеграл береться за всіма можливими значеннями  $\theta$ , щоб врахувати всі рівні латентних ознак.

### 3. Оцінка індивідуальних прихованих показників $\theta_j$

Після оцінки параметрів та розподілу латентної ознаки  $\theta$ , оцінка  $\theta$  для кожного респондента в рамках СТТ може бути проведена за допомогою різних методів. Це включає в себе методи, засновані на машинному навчанні, очікуваному апостеріорному (EAP) оцінюванні та модальному апостеріорному (MAP) методі оцінювання.

Кожен респондент  $j$  характеризується власним  $\theta$  апостеріорним розподілом  $g(\theta | \vec{\eta}, \vec{y}_j)$ , який може бути використаний для оцінки  $\theta_j$ .

Розглянемо більш детально кожен з методів оцінювання:

*Метод очікуваного апостеріорного (EAP) оцінювання* використовує квадратурне правило Гаусса-Ерміта для апроксимації середнього значення розподілу:

$$\hat{\theta} = E[g(\theta | \vec{\eta}, \vec{y}_j)], \quad (1.20)$$

Цей метод оцінює  $\theta_j$ , використовуючи середнє значення розподілення як очікуване значення.

Модально апостеріорний метод оцінювання (MAP) знаходить моду розподілу за допомогою ітераційного методу оцінювання Фішера:

$$\widehat{\theta}_{t+1} = \widehat{\theta}_t - \left[ \frac{\frac{\partial l(\theta_j | \vec{\eta}, y_{ij})}{\partial \theta_j}}{\frac{\partial^2 l(\theta_j | \vec{\eta}, y_{ij})}{\partial \theta_j^2}} \right], \quad (1.21)$$

Де:

$$\left[ \frac{\frac{\partial l(\theta_j | \vec{\eta}, y_{ij})}{\partial \theta_j}}{\frac{\partial^2 l(\theta_j | \vec{\eta}, y_{ij})}{\partial \theta_j^2}} \right] - \text{відношення першої похідної логарифмічної функції}$$

правдоподібності  $\theta$  та матриці Гессе як матриці других похідних логарифмічної функції правдоподібності  $\theta_j$ . Цей метод є ітеративним і продовжується до тих пір, поки не буде досягнута достатня збіжність.

Обидва методи, EAP та MAP [36], забезпечують ефективні підходи до оцінювання латентних ознак респондентів, використовуючи інформацію, зібрану з тестових відповідей, та параметри елементів, визначені в моделях IRT.

## Використання

Інтеграція 3PL Моделі Раша у розроблювану систему для оцінювання користувачів є ключовим рішенням у створенні ефективного та корисного інструменту. Використання цієї моделі з IRT дозволяє перевірити існуючі та відібрані тести системи на можливість їх проходження середньостатистичним користувачем [37]. На поточному етапі розробки, система здійснює оцінку індивідуальних прихованих показників здібностей користувача, обозначених як  $\theta_j$ ,

після проходження тестування. Це дозволяє зрозуміти, які тести користувач здатен пройти.

Оскільки система не зберігає окремих показників здібностей для кожного користувача, наразі немає можливості повноцінно оцінити параметр дискримінативності, що дозволяє розуміти, як змінюється ймовірність відповіді на питання при зміні здібностей користувача. Ураховуючи це, для інтеграції IRT у систему було обрано 3PL Модель Раша. Вона визначається як оптимальна для потреб системи, зокрема завдяки можливості врахувати оцінку індивідуальних прихованих показників здібностей користувача, які припускається рівними  $\theta = 0$ . Такий підхід дозволяє визначити, як середньостатистичний користувач справиться з тестами. Показник дискримінативності при цьому береться як середній, що сприяє більш точному визначенню здатності тестів розрізняти користувачів за рівнем їхніх здібностей.

Завдяки цій інтеграції 3PL Моделі Раша у систему, було можливо провести глибокий аналіз тестів, що включені до системи. Аналіз зосереджувався на визначенні ефективності кожного тесту для середньостатистичного користувача, виходячи з припущення, що його рівень здібностей є нейтральним ( $\theta = 0$ ). Цей підхід дозволив виявити, які тести є достатньо збалансованими для різних рівнів користувачів, а які можуть бути надто складними або надто простими. Такий аналіз є важливим для налаштування системи таким чином, щоб вона була корисною та доступною для широкого спектру користувачів.

Використання середнього значення для параметра дискримінативності ( $a_i = 1$ ) у моделі додатково сприяло отриманню об'єктивних оцінок тестів. Цей параметр дозволив системі адекватно відображати варіативність реакцій різних користувачів на тестові завдання, що забезпечує вищу точність оцінювання.

Результати аналізу тестів, які були представлені у Додатку А, Таблиці А.1, надають цінні відомості щодо якості тестових завдань та їх придатності для оцінювання різних категорій користувачів. Це, в свою чергу, дозволяє робити інформовані рішення щодо подальшого розвитку та вдосконалення системи, щоб вона була максимально ефективною та відповідала потребам користувачів.

### 1.3.4 Інтерфейс системи

При вході в систему користувач відразу потрапляє на головну сторінку (Додаток Б, Рисунок Б.1), де представлена базова інформація про систему [38]. Тут користувач може дізнатися більше про систему, а для початку тестування слід перейти до вкладки «Test».

На сторінці «Test» (Додаток Б, Рисунок Б.2) пропонується перегляд доступних тестів, де користувач може вибрати відповідний для себе.

Після вибору тесту користувачу відкривається сторінка з детальним описом тесту. Щоб розпочати тестування, потрібно натиснути кнопку «Next» (Додаток Б, Рисунок Б.3).

Для персоналізації процесу тестування перед користувачем відкривається можливість налаштування тестів за складністю та технологіями (Додаток Б, Рисунок Б.4).

Завершивши налаштування, користувач переходить до самого тестування (Додаток Б, Рисунок Б.5). Відповідаючи на питання, користувач вибирає одну правильну відповідь і переходить до наступного завдання, натиснувши «Next task».

Після завершення тесту користувача перенаправляє на сторінку з результатами тестування (Додаток Б, Рисунок Б.6), де він бачить свій загальний результат і результати за кожною технологією у відсотках, а також отримує рекомендації щодо подальшого навчання. Також є можливість переглянути власні відповіді та порівняти їх із правильними.

Важливою особливістю системи є те, що на підставі результатів тестування вона автоматично визначає рівень компетенції користувача і, відповідно, надає рекомендації щодо матеріалів для подальшого вивчення. Також система дає уявлення про те, на які посади може претендувати користувач з таким рівнем знань, або які області вимагають подальшого вивчення для досягнення вищих кар'єрних цілей, а саме:

- У випадку коли користувач пройшов тестування на слабкий рівень знань, він отримає поради для досягнення посади “Intern/Trainee Front-end” розробника, а також прогноз заробітної плати на наступний рік.(Додаток Б, Рисунок Б.7)
- У випадку коли користувач пройшов тестування на середній рівень знань, система запропонує рівень знань як для посади “Intern/Trainee Front-end” розробника, але користувач отримає поради для досягнення посади “Junior Front-end” розробника, а також прогноз заробітної плати на наступний рік.(Додаток Б, Рисунок Б.8)
- У випадку коли користувач пройшов тестування на високий рівень знань, система запропонує рівень знань як для посади “Junior Front-end” розробника з подальшими варіантами розвитку для збільшення шансів на отримання пропозиції на роботу, а також прогноз заробітної плати на наступний рік.(Додаток Б, Рисунок Б.9)

## **Висновки до розділу 1**

У даному розділі було розглянуто розділ *«теоретичні основи та методичні підходи до системи вивчення технологій веб-розробки»*, та проведено класифікацію апаратних та програмних забезпечень для розуміння сучасних потреб. Окрім того, було визначено ключові показники ефективності системи, що дозволить краще розуміти процес тестування. А також, було розглянуто побудову системи, для розуміння її функціоналу та відповідності процесів до дій, що допоможе під час тестування продуктивності.

## **2 АНАЛІЗ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ**

У наш час людство не стоїть на місці, та сфера інформаційних технологій розвивається все більше з кожним днем. Постійно, люди створюються все краще апаратне забезпечення, яке дозволяє оброблювати потік даних у декілька разів швидше. Це у свою чергу, приводить до апдейту поточних програмних забезпечень, а також створенню нових програм та додатків, які краще та швидше вирішують сучасні проблеми. Саме тому, не можна один раз створити систему та не підтримувати її, оскільки з часом технологій стає набагато більше, та система має їх підтримувати. Інакше, користувачеві буде не зручно використовувати систему та вона стане не актуальною не через її функціонал, а через не підтримку сучасних девайсів.

### **2.1 Статистичний аналіз використання апаратного та програмного забезпечення**

Задля розуміння поточних сучасних девайсів користувачів, необхідно проаналізувати які апаратні та програмні забезпечення є актуальними та популярними у наш час, що дозволить покращити систему, та підтримувати майже усіх користувачів. Саме тому, необхідно провести статистичний аналіз використання апаратного та програмного забезпечення у 2023 році.



### 2.1.1 Статистика використання апаратного забезпечення у 2023 році.

Розгляд апаратного забезпечення можна розділити на CPU та число ядер, об'єм RAM пам'яті, відеокарта. Це дозволяє краще оцінити використання різного апаратного забезпечення.

Статистичні дані з використання апаратного забезпечення використовуються завдяки сервісам *Statista* та *Steam*.

*Statista* [39], німецька онлайн-платформа, спеціалізується на зборі та візуалізації даних. Поряд з відкритими даними третіх сторін, *Statista* також пропонує ексклюзивні дані через платформу, які збираються завдяки опитуванням та аналізу, проведеним її командою. Згідно з власними публікаціями, *Statista* пропонує понад 1,000,000 статистичних даних по понад 80,000 тем з більш ніж 22,500 джерел у понад 150 країнах, з кількістю відвідувань платформи у 31 мільйон разів на місяць (станом на грудень 2022 року).

Компанія надає статистику та результати опитувань, які представлені у вигляді діаграм та таблиць. Основними цільовими групами компанії є бізнес-клієнти, лектори та дослідники. Дані, які надає компанія, охоплюють, серед іншого, рекламу, поведінку споживачів чи конкретні галузі. Крім того, компанія пропонує звіти про рекламу, споживчу поведінку, політику та суспільство, а також дані про окремі сектори економіки та країни.

*Steam* [40] – це сервіс цифрової дистрибуції відеоігор та торгова платформа, розроблена компанією Valve Corporation. Вона була запущена у вересні 2003 року як програмний клієнт для автоматичного оновлення ігор Valve, а з кінця 2005 року розширилася до дистрибуції ігор третіх сторін. *Steam* пропонує різноманітні функції, включаючи співпадіння ігрових серверів із заходами проти нечесної гри Valve Anti-Cheat, соціальні мережі та сервіси стрімінгу ігор. Функції клієнта *Steam* включають автоматичне оновлення ігор, хмарне зберігання прогресу в іграх і спільнотні можливості, такі як пряме повідомлення, функції накладання у грі та віртуальний ринок колекційних предметів.

Steamworks, безкоштовний інтерфейс програмування додатків (API), випущений у 2008 році, використовується розробниками для інтеграції функцій Steam, включаючи управління цифровими правами (DRM), у свої ігрові продукти. Багато видавців ігор почали розповсюджувати свої титули на Steam у тому ж році. Спочатку розроблена для операційних систем Microsoft Windows, Steam була випущена для macOS у 2010 році та для Linux у 2012 році. Мобільні додатки для доступу до онлайн-функцій Steam вперше були випущені для iOS та Android у 2012 році. Магазин платформи також пропонує інший цифровий контент і апаратне забезпечення для ігор від Valve, включаючи програмне забезпечення для продуктивності, саундтреки до ігор, відео та гарнітуру віртуальної реальності Valve Index.

### 1) CPU

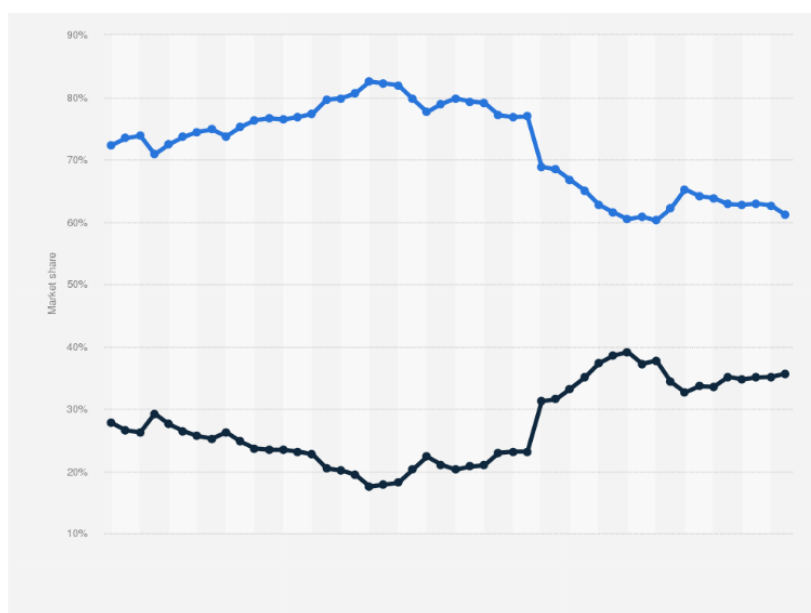


Рисунок 2.1 – Статистика використання CPU від Intel (синій) та AMD (чорний) у 2023 році

Розглянемо графік зі статистикою використання CPU від Intel та AMD у 2023 на *Рисунку 2.1*. Як можна одразу побачити, лідером по використанню CPU є Intel, який займає 61.1% від усієї кількості ядер CPU у користувачів по всьому світу. AMD відповідно займає друге місце та займає 35.6% від усієї кількості ядер CPU у користувачів по всьому світу.

## 2) Кількість ядр CPU

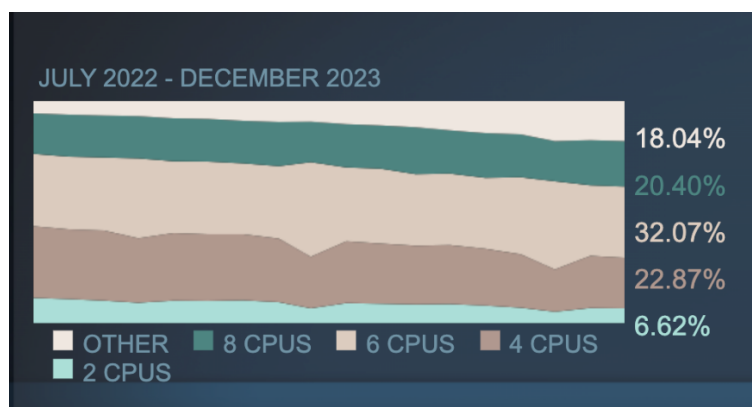


Рисунок 2.2 – Статистика кількості ядр CPU у 2023 році

Розглянемо графік зі статистикою кількості ядр CPU у 2023 на *Рисунку 2.2*. Як можна одразу побачити, лідером серед кількості ядр CPU є 6 ядр, які займають 32.07% від усієї кількості ядр CPU у користувачів по всьому світу. Друге місце, посідають 4 ядра CPU, які займають 22.87% від усієї кількості ядр CPU у користувачів по всьому світу. Третє місце, посідають 8 ядр CPU, які займають 20.40% від усієї кількості ядр CPU у користувачів по всьому світу. Четверте місце, посідають 2 ядра CPU, які займають 6.62% від усієї кількості ядр CPU у користувачів по всьому світу. Усі інші кількості ядр CPU займають 18.04% від усієї кількості ядр CPU у користувачів по всьому світу.

Саме тому, можна підкреслити що головною кількістю ядр на ринку залишаються 4, 6, 8.

## 3) Об'єм RAM пам'яті

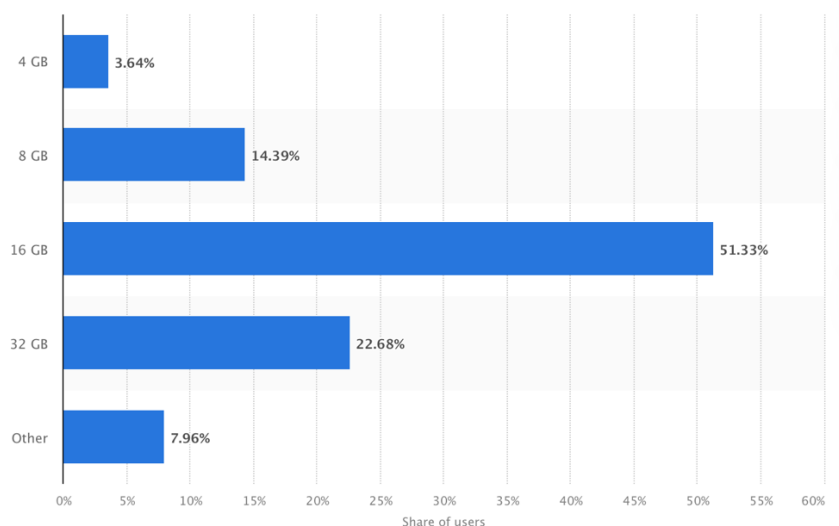


Рисунок 2.3 – Статистика використання об’єму RAM пам’яті (у GB) у 2023 році

Розглянемо графік зі статистикою використання об’єму RAM пам’яті у 2023 на *Рисунку 2.3*. Як можна одразу побачити, лідером серед об’єму RAM пам’яті є 16 GB, яке займає 51.33% від усього використання об’єму RAM пам’яті користувачами по всьому світу. Друге місце, посідає 32 GB об’єму RAM пам’яті, яке займає 22.68% від усього використання об’єму RAM пам’яті користувачами по всьому світу. Третє місце, посідає 8 GB об’єму RAM пам’яті, яке займає 14.39% від усього використання об’єму RAM пам’яті користувачами по всьому світу. Четверте місце, посідає 4 GB об’єму RAM пам’яті, яке займає 3.64% від усього використання об’єму RAM пам’яті користувачами по всьому світу. Усі інші об’єми RAM пам’яті займають 7.96% від усього використання об’єму RAM пам’яті користувачами по всьому світу.

Саме тому, можна підкреслити що головним об’ємом RAM пам’яті на ринку залишаються 8, 16, 32.

#### 4) Відеокарти

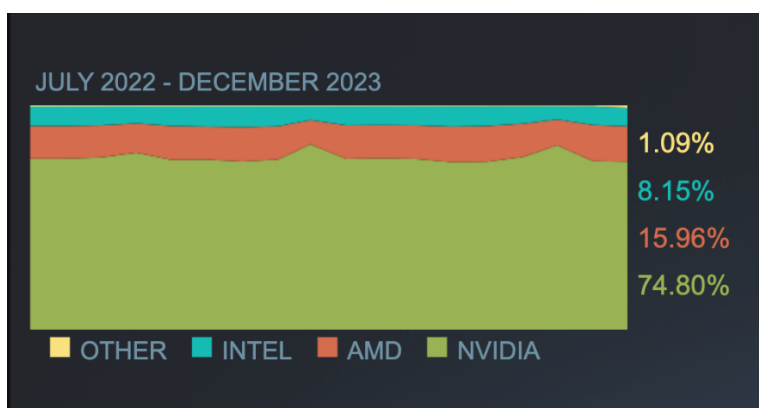


Рисунок 2.4 – Статистика використання відеокарт у 2023 році

Розглянемо графік зі статистикою використання відеокарт у 2023 на *Рисунку 2.4*. Як можна одразу побачити, лідером серед відеокарт є відеокарти від Nvidia, які займають 74.80% від усіх відеокарт користувачів по всьому світу. Друге місце, посідають відеокарти від AMD, які займають 15.96% від усіх відеокарт користувачів по всьому світу. Третє місце, посідають відеокарти від Intel, які є вбудованими графічними картами у процесори, та займають 8.15% від усіх відеокарт користувачів по всьому світу. Після них ідуть усі інші відеокарти та вони займають 1.09% від усіх відеокарт користувачів по всьому світу.

Саме тому, можна підкреслити що головними відеокартами на ринку залишаються Nvidia, AMD, та Intel.

### 2.1.2 Статистика використання програмного забезпечення у 2023 році.

Розгляд програмного забезпечення можна розділити на комп'ютерне програмне забезпечення, та веб-браузери. Це дозволяє краще оцінити використання різного програмного забезпечення.

Статистичні дані з використання програмного забезпечення використовуються завдяки сервісу *StatCounter* [41].

*StatCounter*, заснований у 1999 році, є веб-сайтом для аналізу веб-трафіку, розташованим у Дубліні, Ірландія. Дані *StatCounter* використовуються для

обчислення частки використання веб-ресурсів, наприклад. Станом на травень 2019 року, StatCounter застосовується на 0.9% усіх вебсайтів.

Статистика StatCounter базується безпосередньо на переглядах, на відміну від унікальних відвідувачів, з 3 мільйонів сайтів, які використовують цей сервіс, що в результаті дає понад 15 мільярдів переглядів на місяць. Для коригування зсуву вибірки не використовуються жодні штучні вагові коефіцієнти, отже, цифри у статистиці не можна вважати репрезентативними вибірками.

Компанію заснував Аодхан Каллен у віці 16 років. Каллен отримав нагороду "Інтернет-герой" на премії Eircom Golden Spider Awards у 2008 році. Він також був названий "Молодим європейським підприємцем року" за версією BusinessWeek у 2007 році.

### 1) Комп'ютерне програмне забезпечення

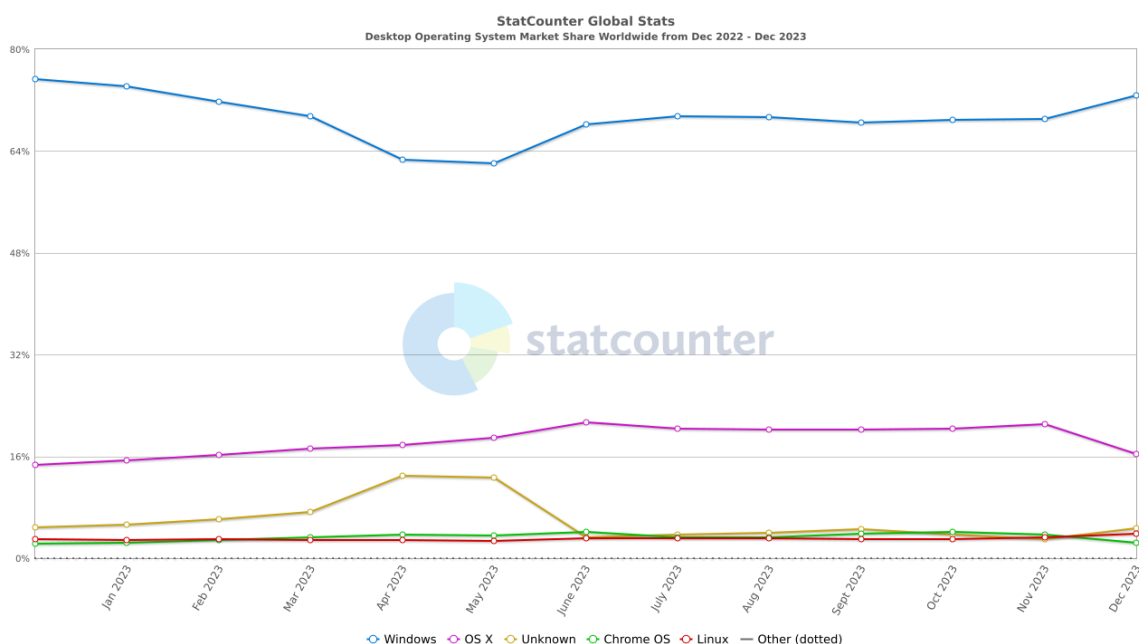


Рисунок 2.5 – Статистика використання комп'ютерних операційних систем у 2023 році

Розглянемо графік зі статистикою використання комп'ютерних операційних систем у 2023 на *Рисунку 2.5*. Як можна одразу побачити, лідером у комп'ютерних операційних системах є Windows, який займає 72.79% з усіх користувачів по всьому світу. Існує багато факторів, які



Рисунок 2.6 – Статистика використання комп'ютерних операційних систем у 2023 році

Розглянемо графік зі статистикою використання веб-браузерів у 2023 на *Рисунку 2.6*. Як можна одразу побачити, лідером серед браузерів є Chrome від Google, який займає 64.73% з усіх користувачів по всьому світу. Існує багато факторів, які дозволяють браузеру Chrome залишатись на першому місці, но ключовим на мою думку є доступність та функціонал, оскільки браузер Chrome можна встановити на будь-який пристрій, та він надає велику кількість функціоналу та кастомізації. З великим відривом друге місце займає браузер Safari від компанії Apple, який займає 18.56% з усіх користувачів по всьому світу. Це дуже очевидно, оскільки вже багато років Safari користується попитом на ринку, оскільки Apple постійно оновлює свій браузер, та працює на кастомізацією та безпекою користувача. Але на мою думку, така велика різниця у % між Chrome та Safari є через те ще, браузер Safari встановлена тільки на комп'ютерах від компанії Apple, оскільки він іде разом з операційною системою OS X, та його не можна просто встановити на будь-який комп'ютер. Трійку фіналістів посідає браузер Edge від Microsoft та встановлений одразу в операційну систему Windows. Не усі люди хочуть качати якийсь новий браузер, тому можуть використовувати нативний від операційної системи, тому Edge займає свої 4.97% з усіх користувачів по всьому світу. Четверте місце у рейтингу посідає браузер Firefox, який вже багато років на ринку, та багато користувачів до нього звикли і продовжують використовувати, саме тому Firefox займає свої 3.36% з усіх користувачів по всьому світу. Після них ідуть такі браузери, як Opera – 2.86%, Samsung Internet – 2.59%, та усі інші у сумі – 2.93% [42].

Саме тому, можна підкреслити що головними браузерами на ринку залишаються Chrome, Safari, Edge, Firefox.



## 2.2 Проведення тестування системи на різному апаратному та програмному забезпеченні

Завдяки виконаному статистичному аналізу у попередньому підрозділі, та виходячи з поточних можливостей було обрано використано наступні системи для проведення тестування системи вивчення технологій з веб-розробки на різному апаратному та програмному забезпеченні:

### Перелік використаних апаратних забезпечень:

#### 1) Апаратне забезпечення 1 (Надалі «A31»):

- CPU: *intel i5*
- RAM: *4GB*
- Відеокарта: *intel UHD Graphics*
- Операційна система: Windows

#### 2) Апаратне забезпечення 2 (Надалі «A32»):

- CPU: *intel M*
- RAM: *8GB*
- Відеокарта: *intel HD Graphics*
- Операційна система: Mac OS

#### 3) Апаратне забезпечення 3 (Надалі «A33»):

- CPU: *intel i9*
- RAM: *16GB*
- Відеокарта: *AMD Radeon Pro 5500M*
- Операційна система: Mac OS

#### 4) Апаратне забезпечення 4 (Надалі «A34»):

- CPU: *Apple M2*
- RAM: *8GB*
- Відеокарта: *Apple M2*

- Операційна система: Mac OS

Перелік використаних програмних забезпечень (браузерів):

- *Chrome*
- *Firefox*
- *Safari*
- *Edge*

Перелік проведених тестувань системи на різному апаратному та програмному забезпеченні:

1) *Тести для А31:*

- Тест А31 та браузер Chrome. (Додаток В, Рисунок В.1)
- Тест А31 та браузер Firefox (Додаток В, Рисунок В.2)
- Тест А31 та браузер Edge (Додаток В, Рисунок В.3)

2) *Тести для А32:*

- Тест А32 та браузер Chrome. (Додаток В, Рисунок В.4)
- Тест А32 та браузер Firefox (Додаток В, Рисунок В.5)
- Тест А32 та браузер Safari (Додаток В, Рисунок В.6)

3) *Тести для А33:*

- Тест А33 та браузер Chrome. (Додаток В, Рисунок В.7)
- Тест А33 та браузер Firefox (Додаток В, Рисунок В.8)
- Тест А33 та браузер Safari (Додаток В, Рисунок В.9)

4) *Тести для А34:*

- Тест А34 та браузер Chrome. (Додаток В, Рисунок В.10)
- Тест А34 та браузер Firefox (Додаток В, Рисунок В.11)
- Тест А34 та браузер Safari (Додаток В, Рисунок В.12)

## 2.3 Оцінка продуктивності системи

Завдяки проведеним тестам на різному апаратному та програмному забезпеченні, можна зробити оцінку продуктивності системи вивчення технологій з веб-розробки.

У першу чергу, було проаналізовано кожен тест апаратного забезпечення на різному програмному забезпеченні. Під час аналізу, результати тестування були виписані у окремі таблиці за кожним апаратним забезпеченням. Таблиці базуються на чотирьох ключових показниках продуктивності роботи системи, які були зазначені у Розділі 1, а саме: *швидкість обробки даних; швидкість рендерингу та відрисовки; навантаження CPU; блокери коду.*

Відповідність таблиць до результатів тестування наведені далі:

- *Результати тестування для A31 та різних браузерів наведені у Додатку Г, Таблиці Г.1.*
- *Результати тестування для A32 та різних браузерів наведені у Додатку Г, Таблиці Г.2.*
- *Результати тестування для A33 та різних браузерів наведені у Додатку Г, Таблиці Г.3.*
- *Результати тестування для A34 та різних браузерів наведені у Додатку Г, Таблиці Г.4.*

Наступним кроком, виходячи із результатів тестування можна виділити наступні висновки:

1. Потужність процесора впливає на швидкість обробки даних системою;
2. Наявність відео карти пришвидшує швидкість рендерингу системи, але не можна сказати що система має складний дизайн та буде сповільнювати рендеринг.
3. Браузер в цілому не впливає на результати обробки даних та рендерингу, оскільки за швидкість роботи відповідає апаратне забезпечення.

4. Найкращий результат швидкості обробки даних показала A34 з найпотужним залізом, та найгірший результат показала A32 з найслабкішим апаратним забезпеченням.
5. Найшвидший рендеринг був на A34, та найдовший рендеринг був на A33.
6. Показники рендерингу дуже відрізняються між собою, та на це бажано звернути увагу, це може затягувати генерацію DOM та привести за затримок з рендерингом.
7. Системи Mac OS разом із браузером Chrome від Google показали усі результати набагато кращі ніж із нативним для них браузером Safari від Apple.
8. Навантаження CPU в середньому показує від 25% до 50%, і це дуже великі цифри, та на це необхідно звернути увагу. У всіх браузерів було майже однакове навантаження CPU, але у цифрах найменше було у A31 з Edge, а найбільше було у A32 з Safari. Синергія апаратного та програмного забезпечення у Microsoft поки що краще ніж у Apple.
9. Не усі браузери та не все тести показали блокування коду, що є гарною новиною, але тим не менш, Chrome показував блокування коду в середньому по 300 ms. На цей пункт необхідно звернути увагу.

## **Висновки до розділу 2**

У даному розділі було розглянуто розділ «аналіз та оцінка ефективності системи» та було проведено статистичний аналіз використання апаратного та програмного забезпечення у 2023 році. Враховуючи статистичний аналіз було проведено тестування системи на різному апаратному та програмному забезпечення, а також проведено оцінку результатів тестування з отримання висновків з роботи системи., та побудовані таблиці з результатами тестування

### **3 ВИЗНАЧЕННЯ ОПТИМАЛЬНОГО АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ ВИВЧЕННЯ ТЕХНОЛОГІЙ ВЕБ-РОЗРОБКИ**

Розуміння сучасних потреб апаратних та програмних забезпечень є однією з головних речей під час створенні будь якої системи, оскільки ми маємо робити рішення яке буде коректна працювати будь де, оскільки кількість сучасних пристроїв дуже велика та не існує якого єдиного. Саме тому, необхідно не забувати проводити тестування системи на різних апаратних та програмних забезпеченнях [43], для винаходження проблем до того, як їх побачить користувач системи, оскільки це може змінити його думку про ваш сервіс та ви можете втратити потенційного користувача. Надалі в цьому розділі розглядається мінімально апаратне та програмне забезпечення, яке залежить від сучасних потреб.

#### **3.1. Оцінка мінімальних вимог**

Проведений у Розділі 2 статистичний аналіз ринку апаратного забезпечення виявив ключові тенденції та переважні характеристики, які використовуються мільйонами людей щодня по всьому світу. Згідно з цим аналізом, лідером на ринку процесорів є Intel, який займає більшу частку від загальної кількості ядер CPU у порівнянні з AMD. Зазначимо, що найпоширенішою кількістю ядер в процесорах залишаються 4, 6 та 8.

Щодо оперативної пам'яті, переважними об'ємами на ринку є 8, 16 та 32 Gb, що відображає потреби сучасних програм та додатків. В області графічних процесорів головними гравцями на ринку є Nvidia, AMD та Intel.

Враховуючи ці дані, на мою думку, мінімально оптимальне апаратне забезпечення для використання системи вивчення технологій можна визначити наступним чином:

- *Процесор: Intel i3 або еквівалентний.*
- *Оперативна пам'ять: мінімум 4 Gb.*
- *Відеокарта: базовий рівень, такий як Intel HD Graphics або еквівалентний.*

Ці характеристики відображають мінімальні вимоги, які забезпечують достатню продуктивність для ефективного використання сучасних веб-технологій та програм, гарантуючи зручність та доступність навчання для широкого кола користувачів.

### **3.2 Вибір програмного забезпечення**

На сьогоднішній день, виходячи з результатів аналізу головними операційними системами, які користуються популярністю серед користувачів, є Windows та OS X. Що стосується веб-браузерів, то лідерами залишаються Chrome, Safari, Edge та Firefox. Ці продукти займають провідні позиції на ринку, але важливо зазначити, що не існує конкретного програмного забезпечення, яке було б обов'язковим до використання для всіх, оскільки кожна людина в праві сама вирішувати яке програмне забезпечення вона хоче використовувати, а ось підтримка цих програмних забезпечень це вже питання до бізнесу та систем, які надають послуги користувачам.

Система вивчення технологій веб-розробки має бути гнучкою та адаптивною [44], забезпечуючи користувачам свободу вибору та можливість використовувати те програмне забезпечення, яке їм найбільше подобається. Основна задача системи полягає у підтримці основних сучасних браузерів та забезпеченні комфортної та ефективної взаємодії з користувачем.

Такий підхід дозволяє забезпечити високий рівень сумісності та доступності системи, водночас роблячи її привабливою для широкої аудиторії користувачів, незалежно від їхніх переваг у програмному забезпеченні.

### 3.3 Оптимізація та ефективність системи

У наш час будь який бізнес або система має бути готовим до нових технологій та їх підтримка буде дозволяти вашій системі залишатись сучасною та не ускладнювати user experience [45]. Завдяки проведеним тестам на ефективність та продуктивність системи вивчення технологій з веб-розробки, було проведено оцінку продуктивності системи, що показало як апаратне та програмне забезпечення себе поводить при роботі із системою.

Відповідаючи на питання «чи потребує система покращень для оптимізації та продуктивності?» я відповім що «так». Не дивлячись на доволі гарні результати роботи системи, все одно існують речі, які необхідно покращити, оскільки при подальшому розвитку та масштабуванні системи ці недоліки можуть блокувати ефективність іншого функціоналу та це буде вимагати повного рефакторингу системи, задля досягнення кращої продуктивності. Як нам відомо, рефакторинг потребує час, мотивацію та енергію для його виконання, та не завжди усі три пункти можуть бути в один час [46]. Саме тому, було б краще покращити систему на поточному етапі, та продовжити подальшу розробку після.

Виходячи з отриманих висновків оцінки продуктивності системи, хочу відмітити що саме в системі потребує оптимізації задля досягнення кращої ефективності роботи системи вивчення технологій з веб-розробки, а саме:

- Наразі багато логіки системи лежить на стороні клієнта, тобто Front-end частині, що змушує апаратне забезпечення користувача виконувати усі обрахунки та в кінці рендерити сторінку виходячи із отриманих результатів. Цього не має бути, та необхідно перенести усю логіку

обрахунків результатів тестування на сторону сервера, тобто Back-end, що девайс користувача не перенавантажувався та використовував багато потужності CPU на обрахунки [47]. Це у свою чергу дозволить слабким девайсам швидше взаємодіяти з сервісом.

- UI частина сервісу не є складною [48], але тим не менш вона займає багато часу на рендеринг після обрахунків та вони рендеряться по декілька разів ніби після кожного обрахунку код заново рендерить DOM дерево [49]. Необхідно зробити рефакторинг коду для рендерингу сторінки, та спочатку виконувати усі дії з даними та обрахунками, а вже після виконувати рендеринг сторінки.
- Необхідно передивитись реалізацію побудови сторінки з результатом тестування, оскільки на даному етапі існують блокери коду [50], які можуть привести до більш серйозних проблем у майбутньому.

### **Висновки до розділу 3**

У даному розділі було визначено та обґрунтовано оптимальне апаратне та програмне забезпечення для системи вивчення технологій веб-розробки, яке походить від аналізу сучасних потреб ринку. Окрім того, завдяки результат тестування системи вивчення технологій з веб-розробки, було обґрунтовано подальше вдосконалення та оптимізація системи, що дозволить покращити поточне рішення.



## ВИСНОВКИ

1. Було проведено тестування системи рекомендацій, використовуючи різне апаратне та програмне забезпечення, для отримання результатів продуктивності системи, та дозволило знайти візуальні недоліки системи, а також побачити продуктивність роботи для різних типів користувачів.
2. Використовуючи результати тестування апаратного та програмного забезпечення, було проведено детальний аналіз різних показників, що дозволило знайти слабкі місця у системі, що навпаки перенавантажували систему на стороні користувача, та ускладнювало подальше проходження тестування.
3. Було обґрунтовано мінімально оптимальне апаратне та програмне забезпечення, яке зменшує час обробки результатів тестування та пришвидшую рендеринг системи, саме завдяки отриманим результатам тестування та статистикою використання програмного забезпечення за 2023 рік, а саме:
  - *Процесор: Intel i3 або еквівалентний.*
  - *Оперативна пам'ять: мінімум 4 Gb.*
  - *Відеокарта: базовий рівень, такий як Intel HD Graphics або еквівалентний.*
4. У подальшій розробці система надання рекомендацій буде адаптованою під кожного користувача, оскільки буде покращено продуктивність та час обробки результатів тестування завдяки вирішенню поточних проблем та створено мобільну версію системи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Сидоров М., Табаков Д. Інтернет як засіб соціальної комунікації та соціального впливу // Політичний менеджмент. – 2008.
- 2 Сабадишина Ю. «Тенденції невтішні». Ми проаналізували обсяг українського ІТ-експорту за III квартал й запитали в експертів, чого очікувати до кінця року. – 2023. – [Режим доступу: <https://dou.ua/lenta/articles/it-export-3-quarter-2023/>] – (дата звернення: 13.09.2023).
- 3 Dong, Tao, Elizabeth F. Churchill, Jeffrey Nichols. Understanding the challenges of designing and developing multi-device experiences // In Proceedings of the 2016 ACM conference on designing interactive systems. – 2016. – С. 62-72.
- 4 Основні параметри для класифікації програмного та апаратного забезпечення. – [Режим доступу: <https://worldvision.com.ua/osnovnye-parametry-dlya-klassifikatsii-programmnogo-i-apparatnogo-obespecheniya/>] – (дата звернення: 15.09.2023).
- 5 Wikipedia. Ноутбук, виробництво. – [Режим доступу: <https://uk.wikipedia.org/wiki/Ноутбук.>] – (дата звернення: 20.09.2023).
- 6 Rotem, Efi, Avi Mendelson, Alon Naveh, Micha Moffie. "Analysis of The Enhanced Intel® Speedstep® Technology of the Pentium® M Processor." // In Proceedings of the First Workshop on Temperature-Aware Computer Systems (TACS-1). – 2004.
- 7 Bansal, Nikhil, Kirk Pruhs, Cliff Stein. "Speed scaling for weighted flow time." // SIAM Journal on Computing. – 2010. – Т. 39, № 4. – С. 1294-1308.
- 8 Berson, Alex. Client/server architecture. // McGraw-Hill, Inc. – 1996.
- 9 Carpenter, Tom. Microsoft Windows Operating System Essentials. // John Wiley & Sons. – 2011.
- 10 Smith, Drew, Drew Smith. "Introduction to Apple Platforms." // Apple macOS and iOS System Administration: Integrating and Supporting iPhones, iPads, and MacBooks. – 2020. – С. 1-19.

- 11 Ward, Brian. How Linux works: What every superuser should know. // No Starch Press. – 2021.
- 12 Тарасенко А. "ФІШИНГ В СУЧАСНИХ УМОВАХ." // РЕДАКЦІЙНА КОЛЕГІЯ. – С. 131. [Електронний ресурс].
- 13 Allsopp, John. Developing with web standards. // New Riders. – 2009.
- 14 Carlini, Nicholas, Adrienne Porter Felt, David Wagner. "An evaluation of the google chrome extension security architecture." // In 21st USENIX Security Symposium (USENIX Security 12). – 2012. – С. 97-111.
- 15 Krylov, Georgiy, Maria Patrou, Gerhard W. Dueck, Joran Siu. "The evolution of garbage collection in v8: google's javascript engine." // In 2020 9th Mediterranean Conference on Embedded Computing (MECO). – IEEE. – 2020. – С. 1-6.
- 16 Sarkar, Soumodip. "Staying Alive: Struggles in Innovation Space." // Innovation, Market Archetypes and Outcome: An Integrated Framework. – 2007. – С. 97-116.
- 17 Sadun, Erica, Michael Grothaus, Steve Sande, Erica Sadun, Michael Grothaus, Steve Sande. "Browsing the Internet with Safari." // Taking Your iPad 2 to the Max. – 2011. – С. 123-146.
- 18 Bick, Julie. The microsoft edge: Insider strategies for building success. // Pocket Books. – 1999.
- 19 Ingraio, Daniella. Top 6 Scale Up Challenges. // [Електронний ресурс]. – Режим доступу: <https://staxbill.com/top-6-scale-up-challenges/>. – (дата звернення: 24.09.2023).
- 20 Mehta, Prateek, Prateek Mehta. "Introduction to Google Chrome Extensions." // Creating Google Chrome Extensions. – 2016. – С. 1-33.
- 21 Berners-Lee T. The World Wide Web-past, present and future // Journal of Digital Information. – 1996. – Т. 1. – №. 1.
- 22 Lawson B., Sharp R. Introducing HTML5. // New Riders. – 2011.
- 23 Frain B. Responsive Web Design with HTML5 and CSS3. // Packt Publishing Ltd. – 2012.

- 24 Crockford D. JavaScript: The Good Parts: The Good Parts. // "O'Reilly Media, Inc." – 2008.
- 25 Karlsson, Jill. "Responsive Web Design with CSS Frameworks." – 2014.
- 26 «Задачі з Frontend», [Електронний ресурс], посилання – <https://t.me/frontendquestions> – (дата звернення: 02.10.2023).
- 27 DOU, [Електронний ресурс], посилання – <https://dou.ua/> – (дата звернення: 12.10.2023).
- 28 Lindley C. Frontend Developer Handbook 2017 // Frontend Masters. – 2017.
- 29 Ostertagová E. Modelling Using Polynomial Regression // Procedia Engineering. – 2012. – Т. 48. – С. 500-506.
- 30 Morizot, Julien, Andrew T. Ainsworth, Steven P. Reise. "Toward Modern Psychometrics." // Handbook of Research Methods in Personality Psychology. – 2009. – 407 с.
- 31 Hartig, Johannes, Jana Höhler. "Multidimensional IRT Models for the Assessment of Competencies." // Studies in Educational Evaluation. – 2009. – Т. 35, № 2-3. – С. 57-63.
- 32 Embretson S. E., Reise S. P. Item Response Theory. // Psychology Press. – 2013.
- 33 Tay, Louis, Jeroen K. Vermunt, Chun Wang. "Assessing the Item Response Theory with Covariate (IRT-C) Procedure for Ascertaining Differential Item Functioning." // International Journal of Testing. – 2013. – Т. 13, № 3. – С. 201-222.
- 34 Kuo, Tzu-Chun, Yanyan Sheng. "A Comparison of Estimation Methods for a Multi-Unidimensional Graded Response IRT Model." // Frontiers in Psychology. – 2016. – Т. 7. – 880 с.
- 35 Picard, Etienne, Elias Tahoumi, Franck Plestan, Stéphane Caro, Fabien Claveau. "A New Control Scheme of Cable-Driven Parallel Robot Balancing between Sliding Mode and Linear Feedback." // IFAC-PapersOnLine. – 2020. – Т. 53, № 2. – С. 9936-9943.

- 36 de la Torre, Jimmy, Hao Song, Yuan Hong. "A Comparison of Four Methods of IRT Subscoring." // *Applied Psychological Measurement*. – 2011. – Т. 35, № 4. – С. 296-316.
- 37 Looney, Stephen W., Thomas R. Gulledge Jr. "Use of the Correlation Coefficient with Normal Probability Plots." // *The American Statistician*. – 1985. – Т. 39, № 1. – С. 75-79.
- 38 Singh, Surendra N., Nikunj Dalal, Nancy Spears. "Understanding Web Home Page Perception." // *European Journal of Information Systems*. – 2005. – Т. 14. – С. 288-302.
- 39 «Statista», [Електронний ресурс], посилання – <https://www.statista.com> – (дата звернення: 17.10.2023).
- 40 «Steam», [Електронний ресурс], посилання – <https://store.steampowered.com> – (дата звернення: 18.10.2023).
- 41 «Statcounter», [Електронний ресурс], посилання – <https://gs.statcounter.com> – (дата звернення: 20.10.2023).
- 42 Rostampour, Shahram. "An Application of TOPSIS for Ranking Internet Web Browsers." // *Decision Science Letters*. – 2012. – Т. 1, № 2. – С. 53-58.
- 43 Каменєв О., Щєбликіна О. "Тестування і верифікація програмного забезпечення." – 2020.
- 44 Myung, Jay I., Daniel R. Cavagnaro, Mark A. Pitt. "A Tutorial on Adaptive Design Optimization." // *Journal of Mathematical Psychology*. – 2013. – Т. 57, № 3-4. – С. 53-67.
- 45 Marques, Leonardo, et al. "Understanding UX Better: A New Technique to Go Beyond Emotion Assessment." // *Sensors*. – 2021. – Т. 21, № 21. – 7183 с.
- 46 Kim, Miryung, Thomas Zimmermann, Nachiappan Nagappan. "A Field Study of Refactoring Challenges and Benefits." // *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*. – 2012. – С. 1-11.

- 47 Anand, Vishal, Deepanker Saxena. "Comparative Study of Modern Web Browsers Based on Their Performance and Evolution." // 2013 IEEE International Conference on Computational Intelligence and Computing Research. – IEEE. – 2013. – C. 1-5.
- 48 Lu, Xudong, Jiancheng Wan. "Model Driven Development of Complex User Interface." // MDDAUI@ MoDELS. – 2007.
- 49 Chęć, Dariusz, Ziemowit Nowak. "The Performance Analysis of Web Applications Based on Virtual DOM and Reactive User Interfaces." // Engineering Software Systems: Research and Praxis. – Springer International Publishing. – 2019. – C. 119-134.
- 50 Li, Zhenhao, Tse-Hsun Chen, Weiyi Shang. "Where Shall We Log? Studying and Suggesting Logging Locations in Code Blocks." // Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering. – 2020. – C. 361-372.

## ДОДАТКИ

## Додаток А

## Результат перевірки тестувань використовуючи Item Response Theory

Таблиця А.1 Результат перевірки тестувань використовуючи Item Response Theory

№	$\theta$ , риси респондента	a, нахил	b, складність	c, вгадування	P, результат
1	0.5	1	-2	0.2	0.9393134559830052
2	0.5	1	2	0.2	0.3459404190450851
3	0.5	1	-2	0.2	0.9393134559830052
4	0.5	1	-2	0.167	0.9367848499822969
5	0.5	1	0	0.167	0.6853827760015454
6	0.5	1	-2	0.2	0.9393134559830052
7	0.5	1	0	0.334	0.7483062208012364
8	0.5	1	2	0.2	0.3459404190450851
9	0.5	1	0	0.167	0.6853827760015454
10	0.5	1	-2	0.2	0.9393134559830052
11	0.5	1	-2	0.167	0.9367848499822969
12	0.5	1	0	0.167	0.6853827760015454
13	0.5	1	2	0.167	0.3186879365052969
14	0.5	1	0	0.2	0.6979674649614838
15	0.5	1	-2	0.167	0.9367848499822969
16	0.5	1	0	0.2	0.6979674649614838
17	0.5	1	0	0.25	0.7168444984013909
18	0.5	1	2	0.2	0.3459404190450851
19	0.5	1	-2	0.167	0.9367848499822969
20	0.5	1	-2	0.2	0.9393134559830052
21	0.5	1	0	0.167	0.6853827760015454
22	0.5	1	2	0.167	0.3186879365052969
23	0.5	1	0	0.167	0.6853827760015454
24	0.5	1	2	0.167	0.3186879365052969
25	0.5	1	-2	0.167	0.9367848499822969
26	0.5	1	2	0.2	0.3459404190450851
27	0.5	1	2	0.25	0.38681914285476726
28	0.5	1	-2	0.2	0.9393134559830052
29	0.5	1	0	0.167	0.6853827760015454
30	0.5	1	0	0.167	0.6853827760015

## Додаток Б

### Інтерфейс системи

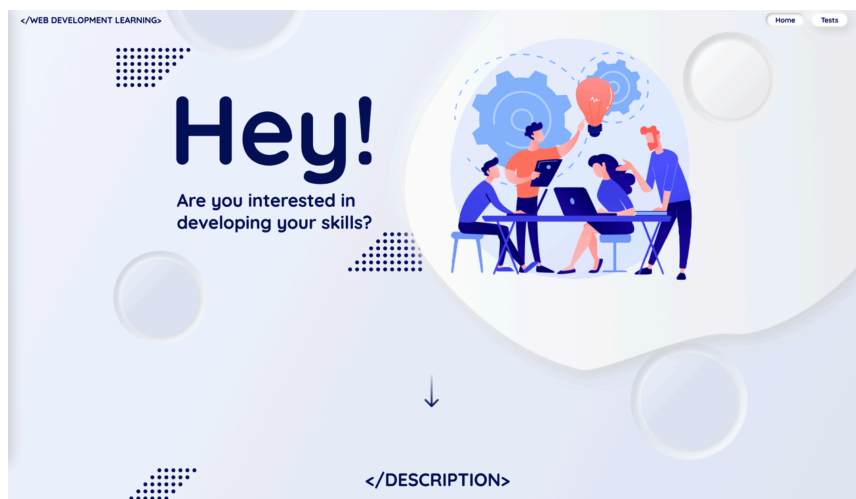


Рисунок Б.1 Головна сторінка

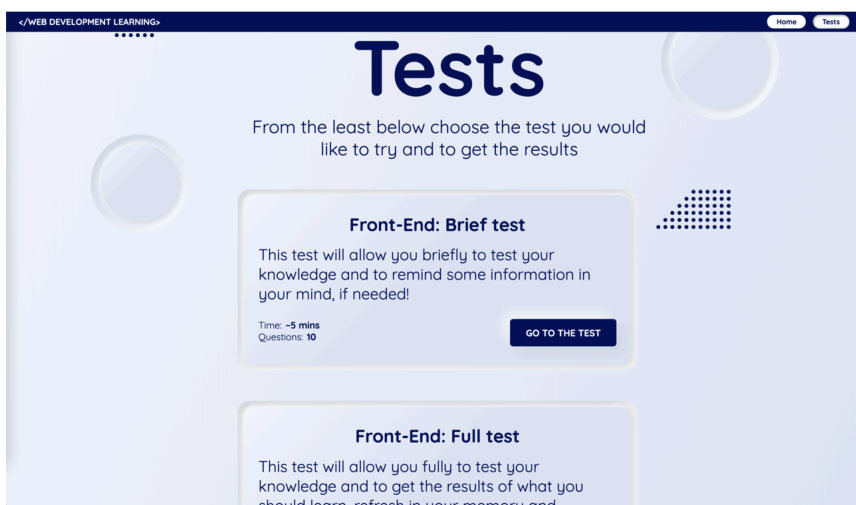


Рисунок Б.2 Сторінка “Tests”



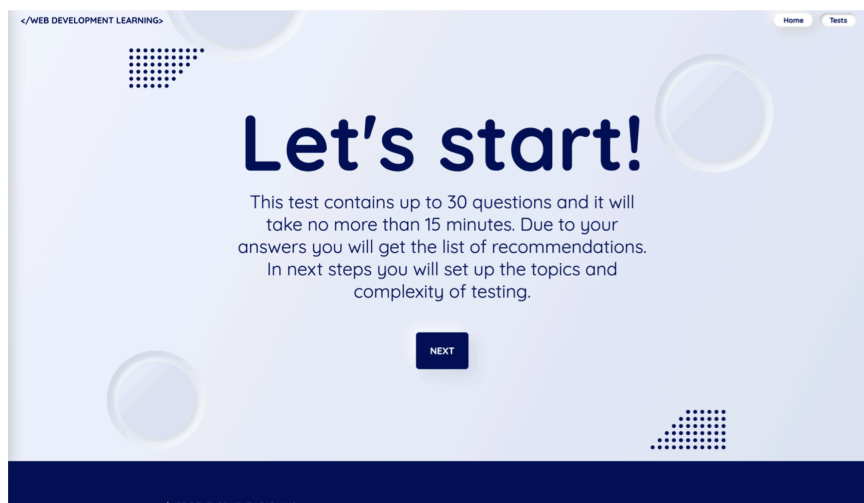


Рисунок Б.3 Сторінка з детальним описом тесту

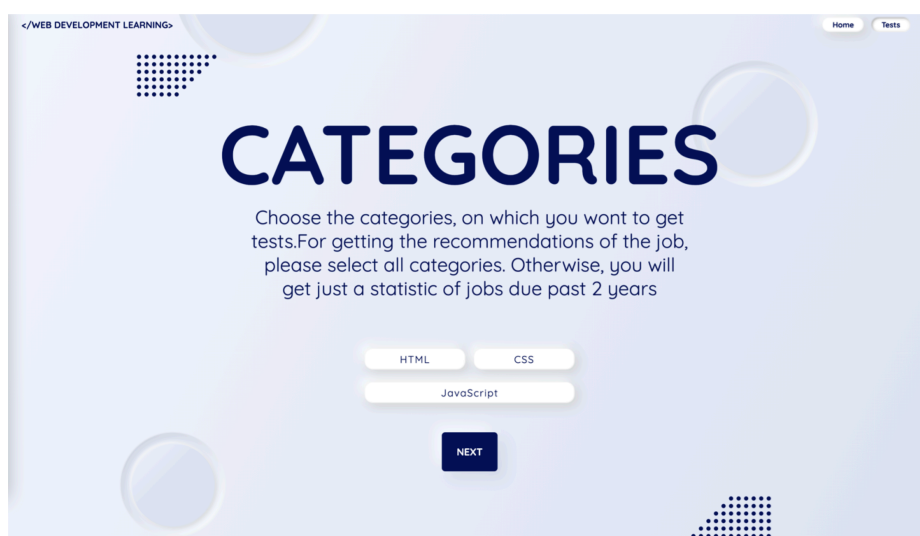


Рисунок Б.4 Сторінка з персоналізацією тестування

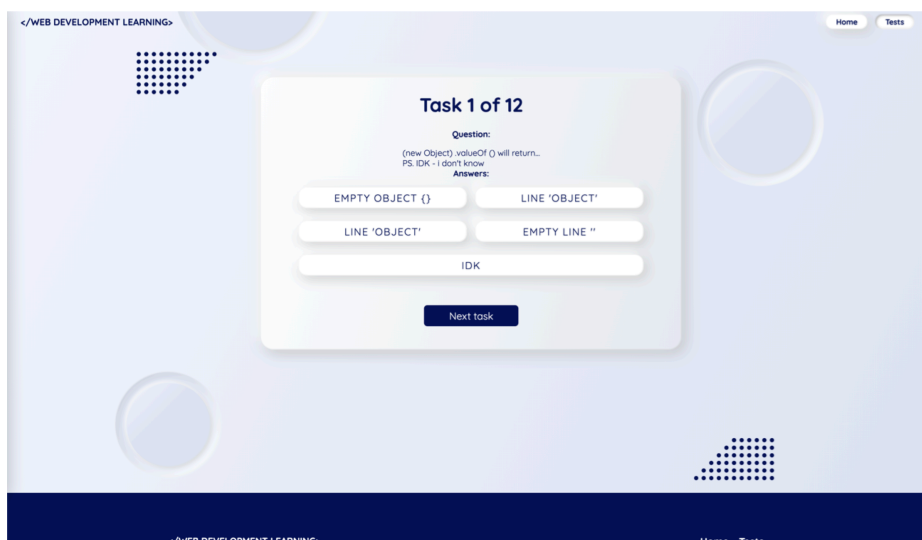


Рисунок Б.5 Сторінка з тестуванням



Рисунок Б.6 Сторінка з результатом тестування



Рисунок Б.7 Результат тестування з слабким рівнем знань

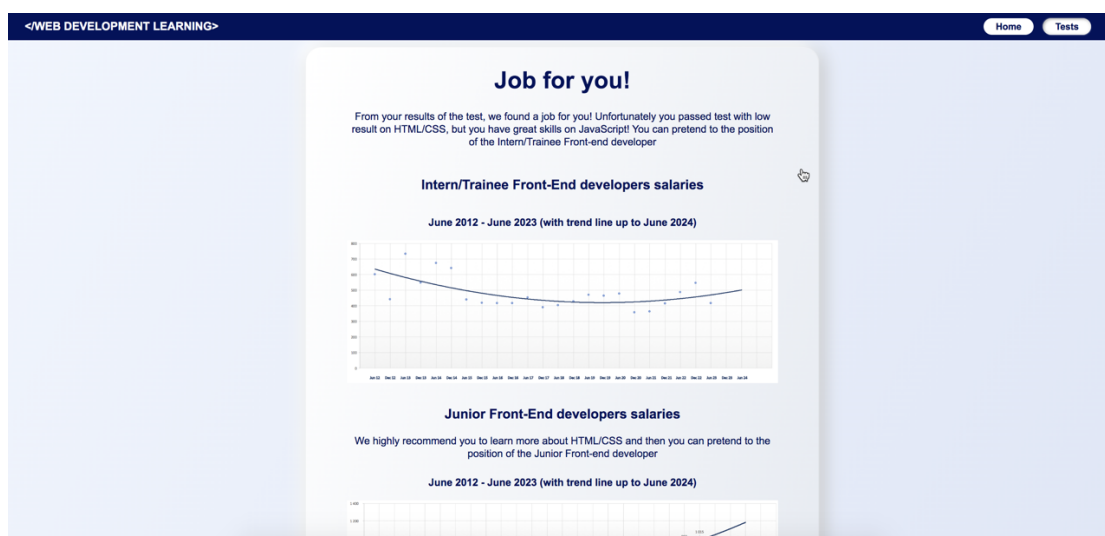


Рисунок Б.8 Результат тестування з середнім рівнем знань

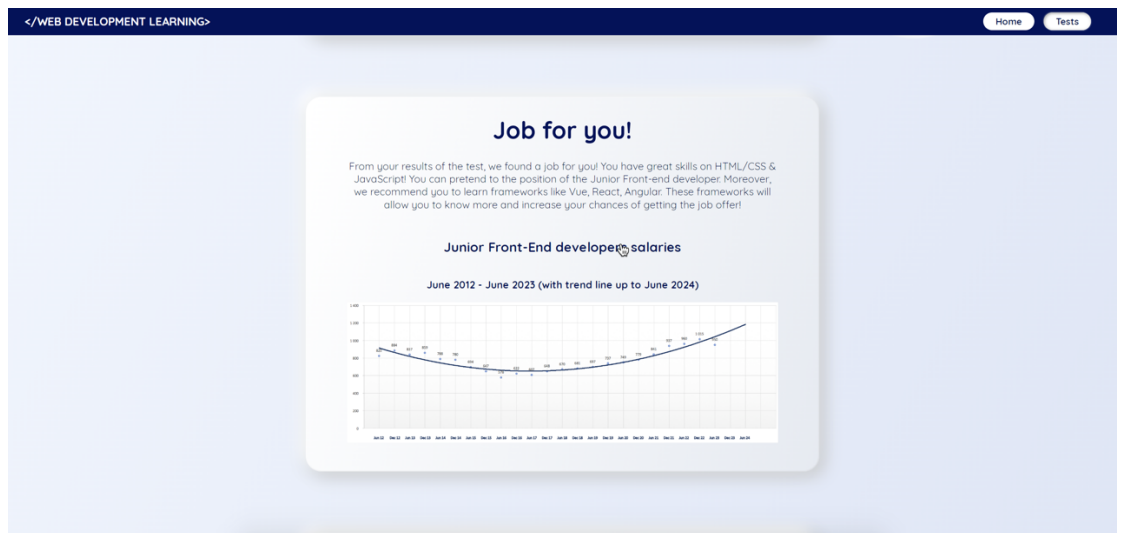


Рисунок Б.9 Результат тестування з високим рівнем знань

## Додаток В

# Проведення тестування системи на різному апаратному та програмному забезпеченні

### 1) Тести для АЗІ:

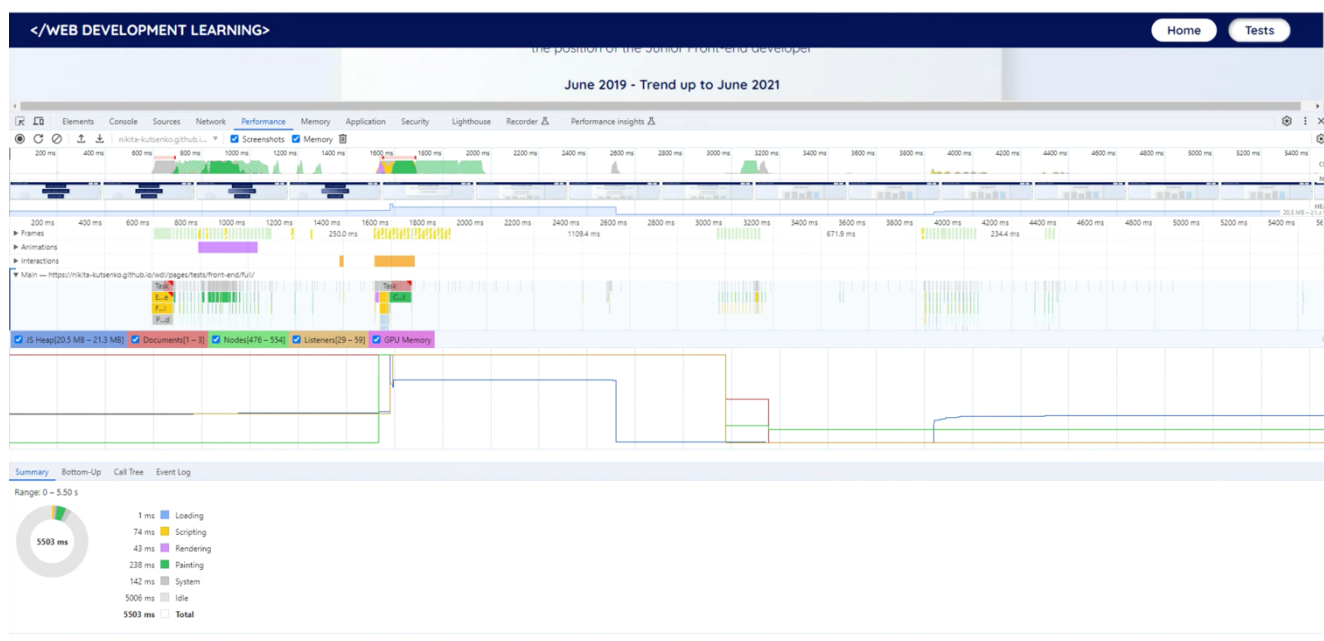


Рисунок В.1 – Результат тесту АЗІ та браузеру Chrome



Рисунок В.2 – Результат тесту АЗІ та браузеру Firefox

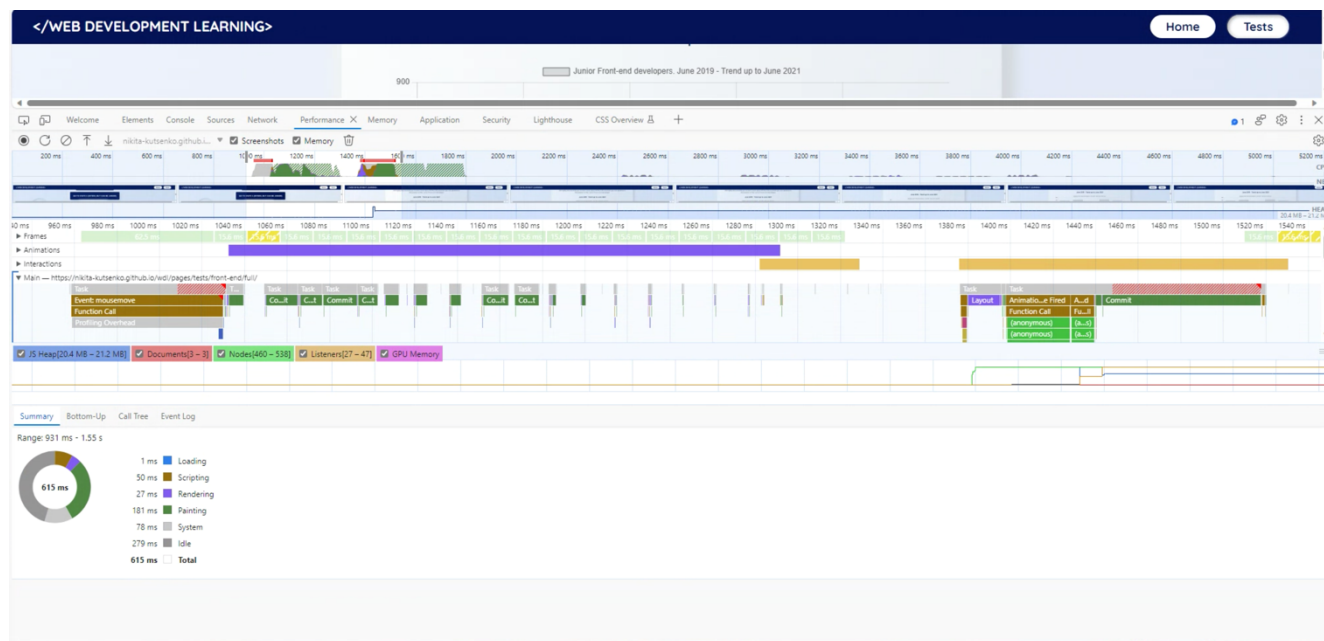


Рисунок В.3 – Результат тесту А31 та браузеру Edge

## 2) Тести для А32:

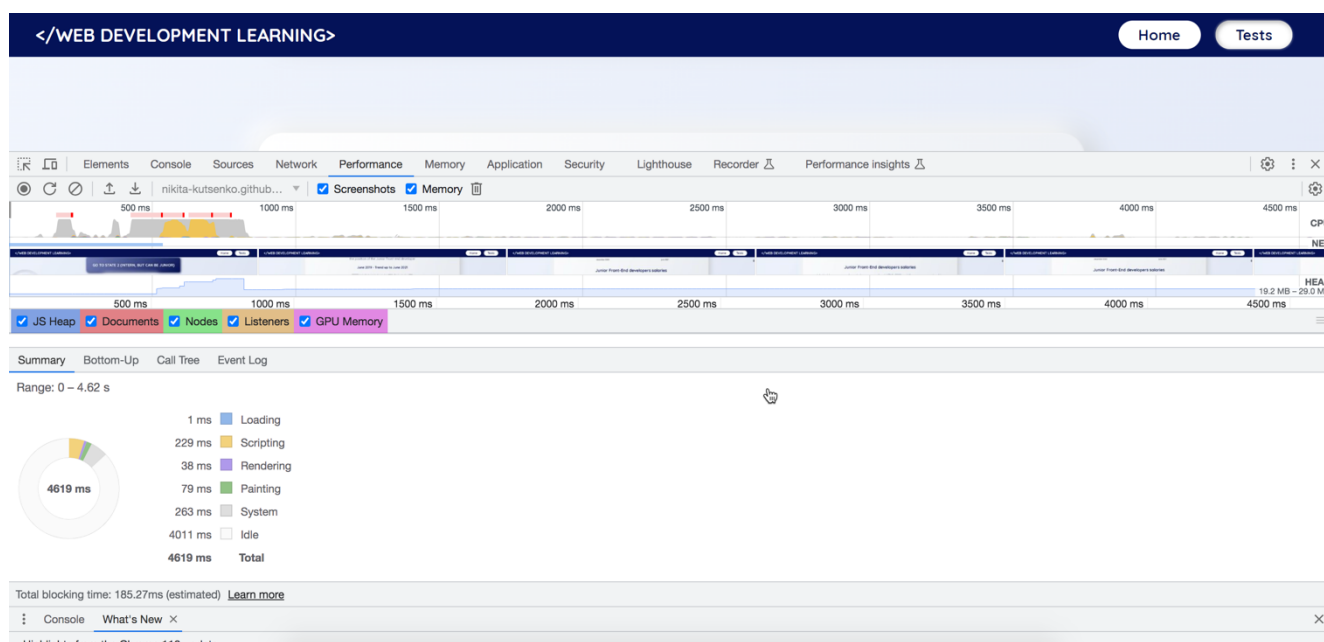


Рисунок В.4 – Результат тесту А32 та браузеру Chrome

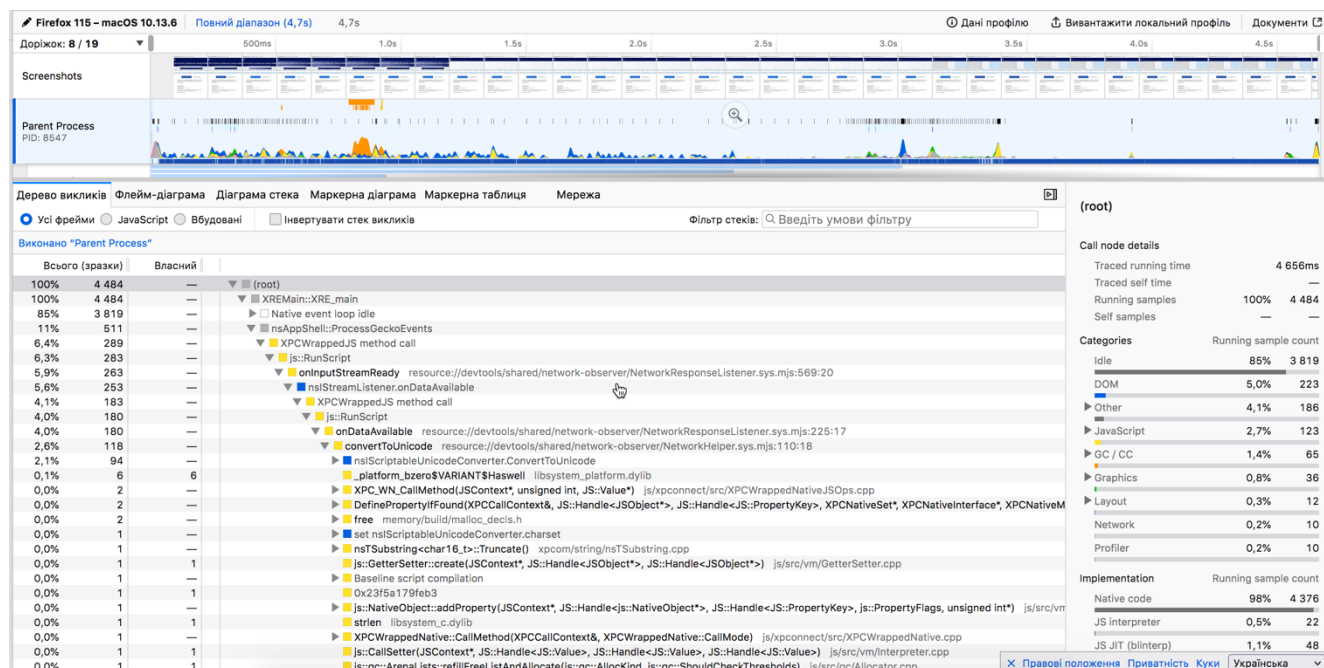


Рисунок В.5 – Результат тесту А32 та браузеру Firefox

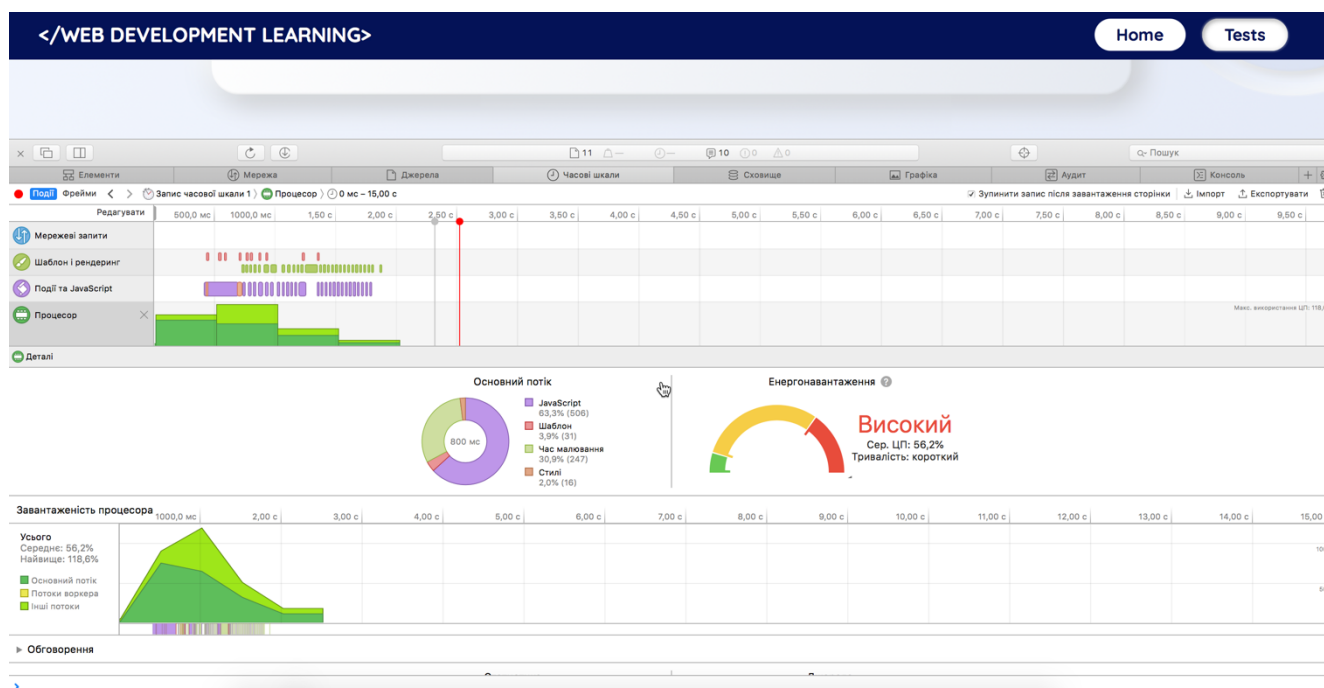


Рисунок В.6 – Результат тесту А32 та браузеру Safari

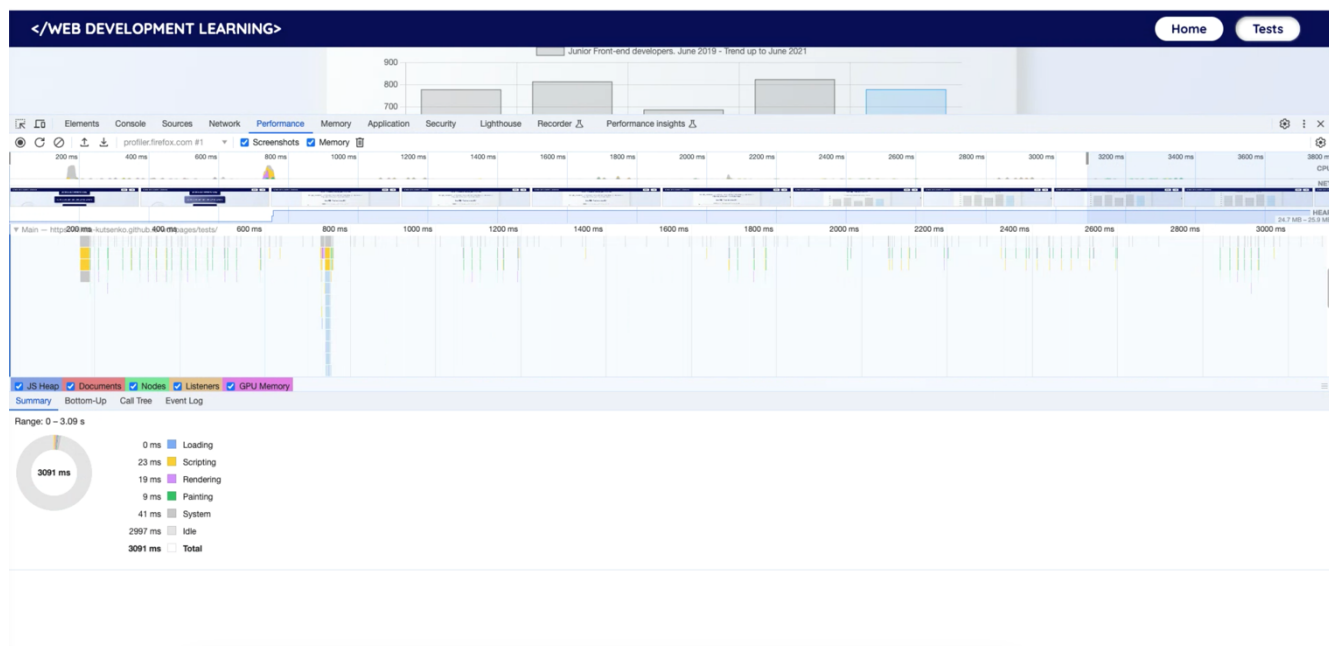
3) Тести для АЗЗ:

Рисунок В.7 – Результат тесту АЗЗ та браузеру Chrome

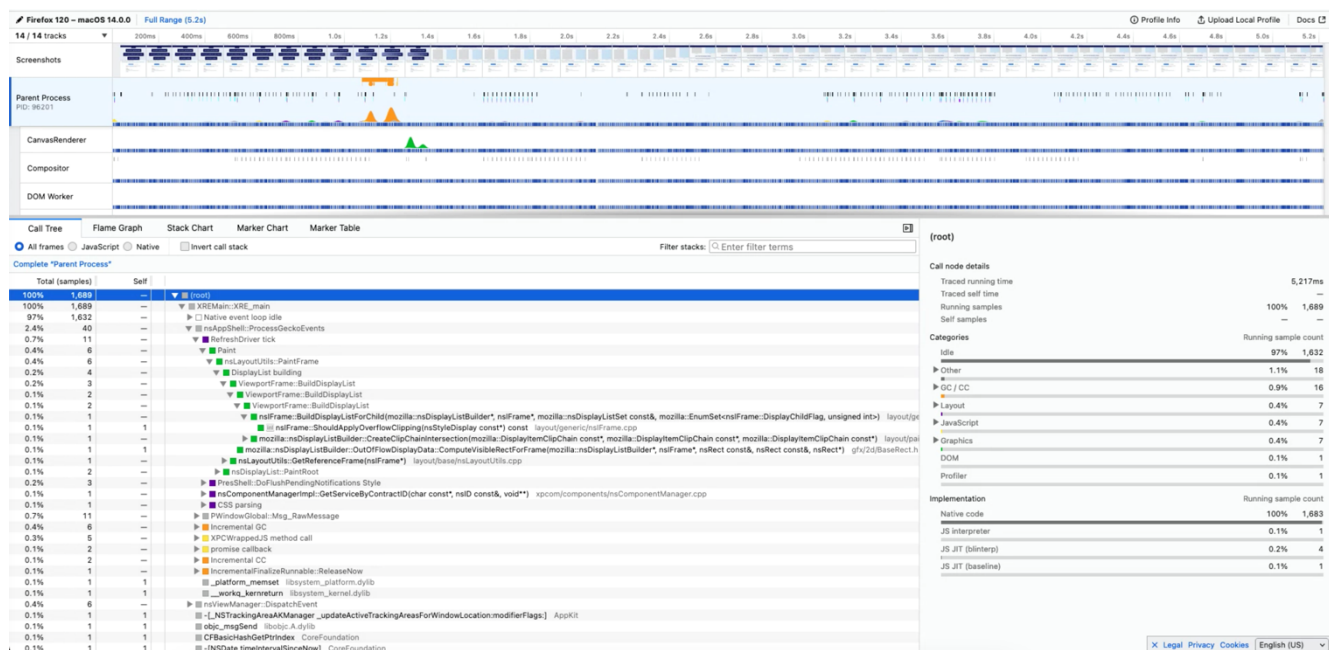


Рисунок В.8 – Результат тесту АЗЗ та браузеру Firefox

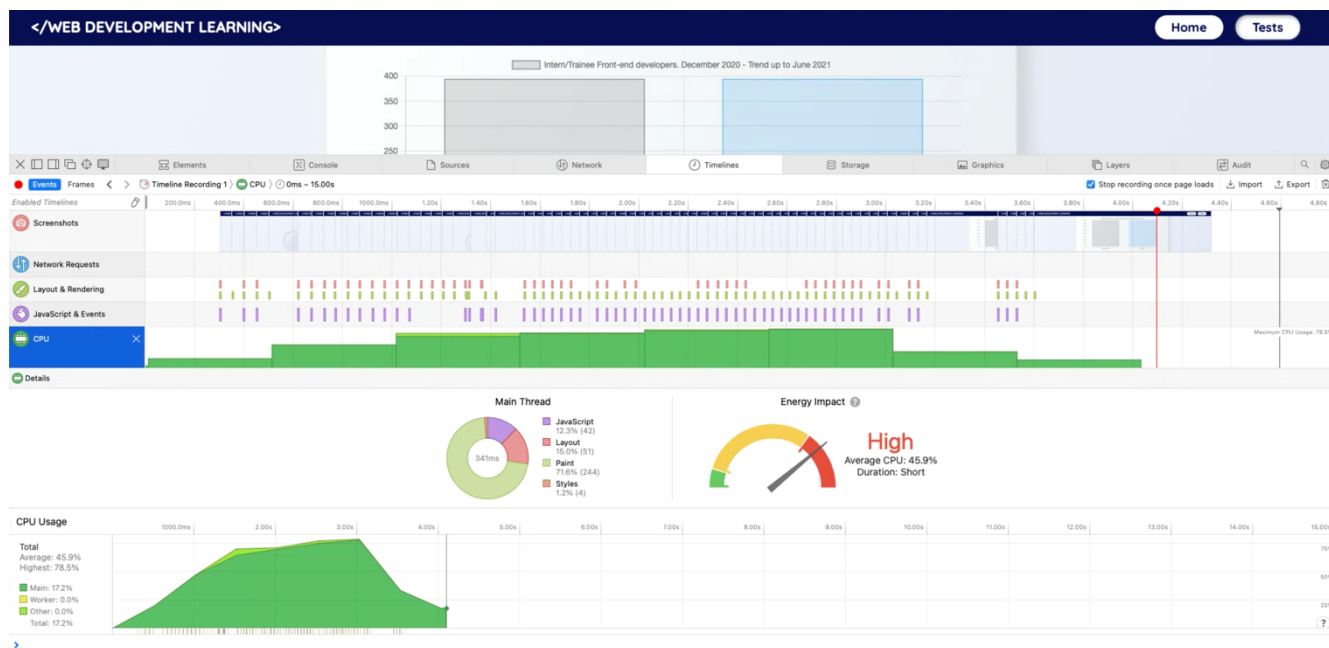


Рисунок В.9 – Результат тесту А33 та браузеру Safari

#### 4) Тести для А34:

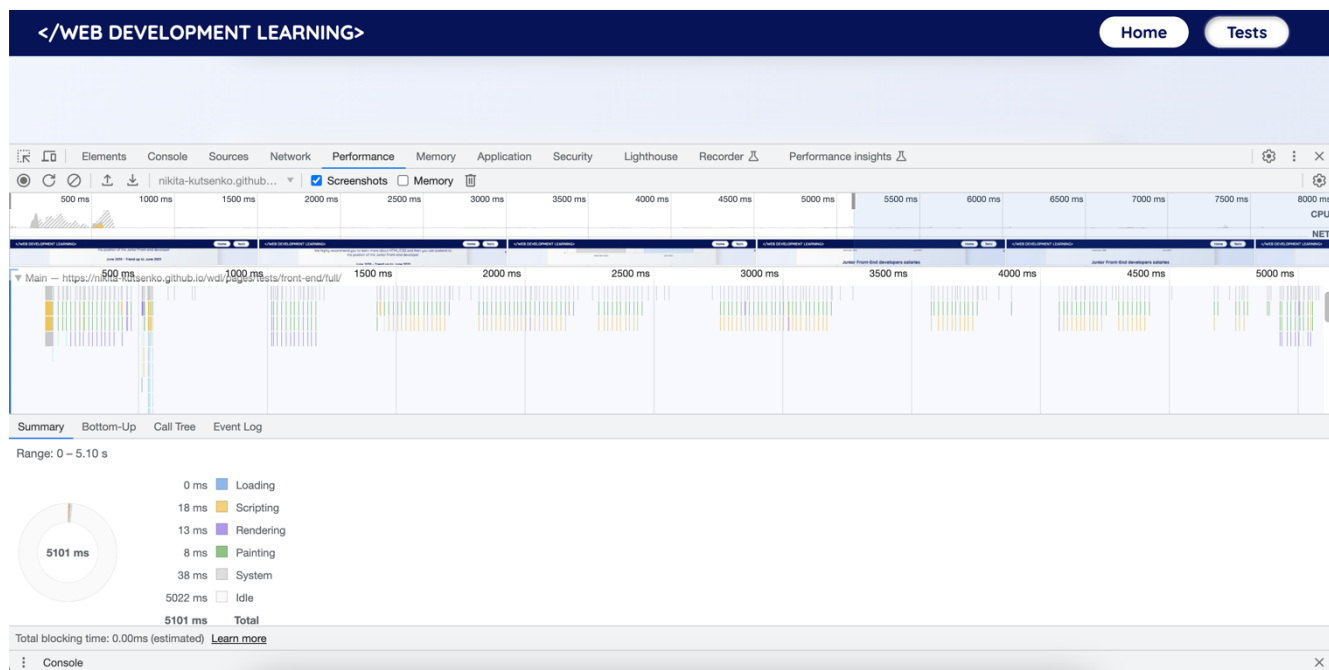


Рисунок В.10 – Результат тесту А34 та браузеру Chrome



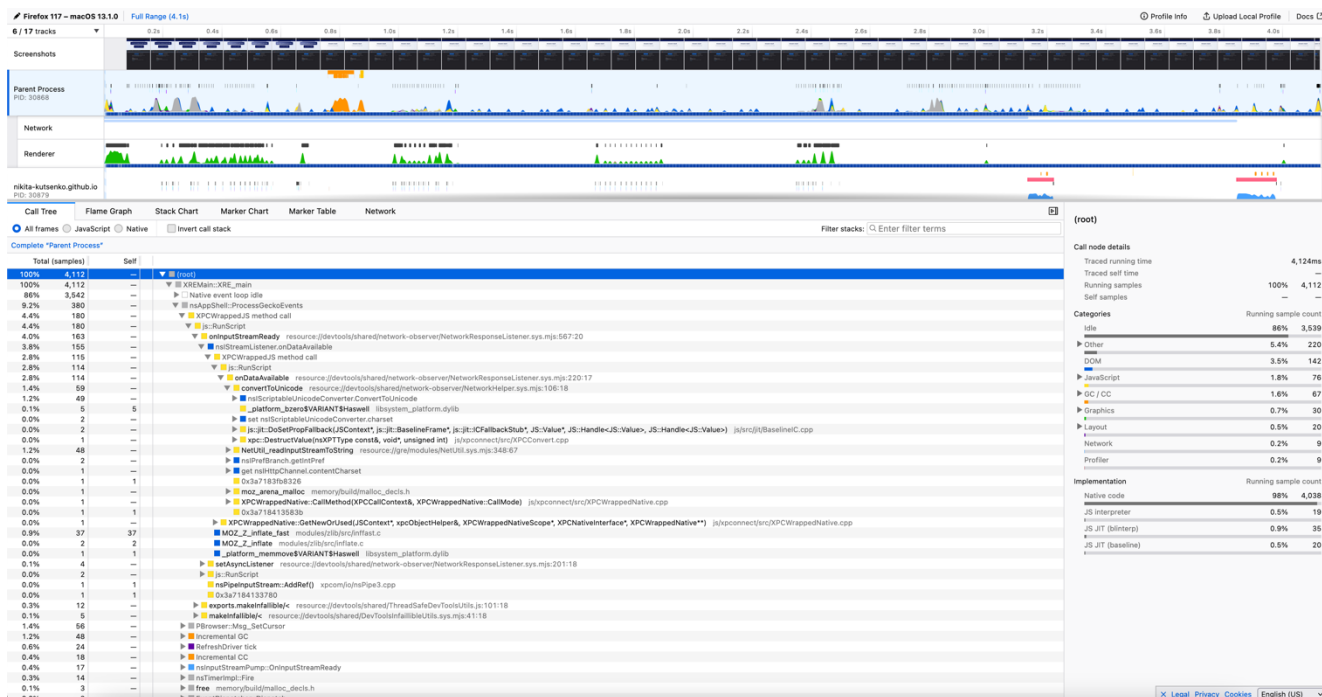


Рисунок В.11 – Результат тесту А34 та браузеру Firefox

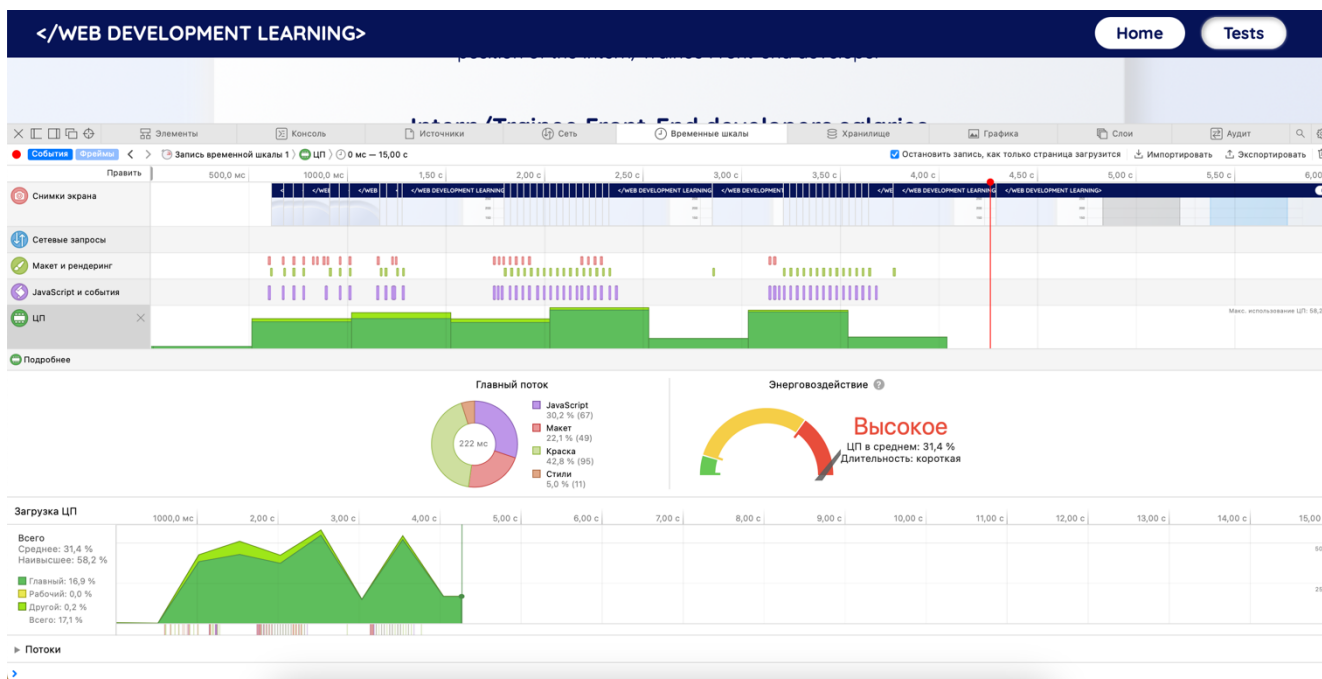


Рисунок В.12 – Результат тесту А34 та браузеру Safari

## Додаток Г

## Оцінка продуктивності системи

Таблиця Г.1 Результат тестування продуктивності системи на А31 та різних браузерах

<i>Браузер</i>	<i>Швидкість обробки даних</i>	<i>Швидкість рендерингу та відрисовки</i>	<i>Навантаження CPU</i>	<i>Блокери коду</i>
<i>Chrome</i>	216 ms	281 ms	~50%	~300 ms
<i>Firefox</i>	221 ms	212 ms	~ 40%	N/A
<i>Edge</i>	229 ms	208 ms	~25%	~300 ms

Таблиця Г.2 Результат тестування продуктивності системи на А32 та різних браузерах

<i>Браузер</i>	<i>Швидкість обробки даних</i>	<i>Швидкість рендерингу та відрисовки</i>	<i>Навантаження CPU</i>	<i>Блокери коду</i>
<i>Chrome</i>	493 ms	117 ms	~50%	~400 ms
<i>Firefox</i>	528 ms	284 ms	~ 40%	N/A
<i>Safari</i>	506 ms	278 ms	56.2%	N/A

Таблиця Г.3 Результат тестування продуктивності системи на А33 та різних браузерах

<i>Браузер</i>	<i>Швидкість обробки даних</i>	<i>Швидкість рендерингу та відрисовки</i>	<i>Навантаження CPU</i>	<i>Блокери коду</i>
<i>Chrome</i>	64 ms	28 ms	~40%	N/A
<i>Firefox</i>	53 ms	326 ms	~ 50%	N/A
<i>Safari</i>	42 ms	295 ms	45.9%	N/A

Таблиця Г.4 Результат тестування продуктивності системи на АЗ4 та різних браузерах

<i>Браузер</i>	<i>Швидкість обробки даних</i>	<i>Швидкість рендерингу та відрисовки</i>	<i>Навантаження CPU</i>	<i>Блокери коду</i>
<i>Chrome</i>	56 ms	21 ms	~25%	N/A
<i>Firefox</i>	53 ms	171 ms	~ 50%	N/A
<i>Safari</i>	67 ms	114 ms	31.4 %	N/A

## ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-  
КОМУНІКАЦІЙНИХ  
ТЕХНОЛОГІЙ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-  
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ




Кафедра Комп'ютерної інженерії

### МАГІСТЕРСЬКА РОБОТА

На тему: «Обґрунтування вибору апаратного та програмного забезпечення для побудови системи вивчення технологій з веб-розробки»

Виконав студент групи КСДМ-62:  
**Куценко Микита Олександрович**  
Науковий керівник:  
**Торошанко Ярослав Іванович**

Київ 2024




## Об'єкт, предмет, мета дослідження


**Об'єкт дослідження:** Система вивчення технологій з веб-розробки.

**Предмет дослідження:** Аналіз апаратного та програмного забезпечення.

**Мета роботи:** Покращити продуктивність роботи системи вивчення технологій з веб-розробки.



2



## Актуальність роботи

Існує потреба у створенні системи орієнтованої на людину та ІТ-ринок, яка б підтримувалась на будь якому сучасному пристрої, та дозволила би отримати рівень знань користувача, а також отримати поради з того, що необхідно ще вивчити або повторити, для того щоб людина могла прагнути на посаду розробника.

3

## Наукова новизна

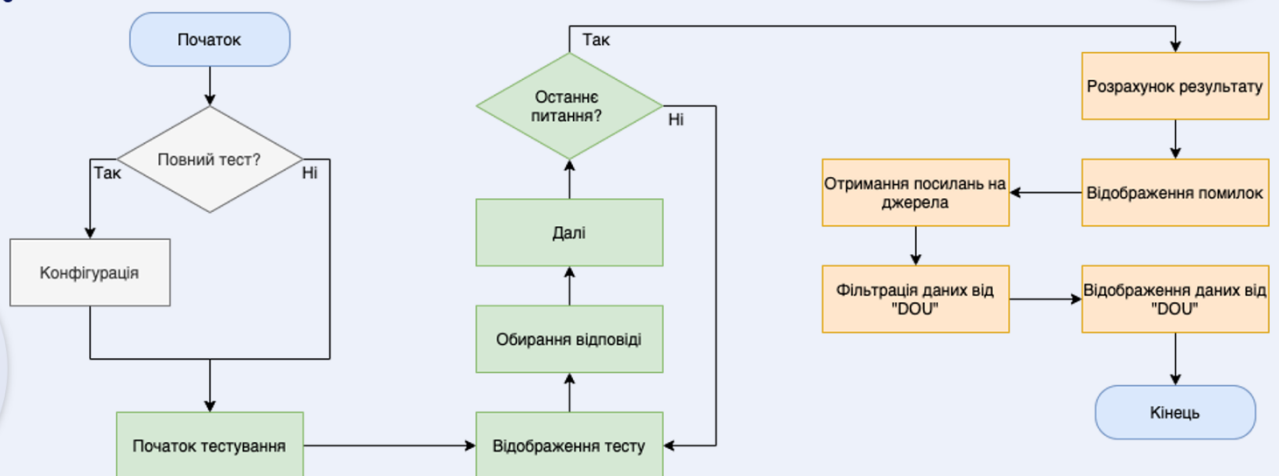
Було досліджено вплив використання різного апаратного та програмного забезпечення для оптимізації роботи системи вивчення технології з веб-розробки; зокрема було винайдено оптимальне апаратне та програмне забезпечення, яке зменшує час обробки результатів тестування та пришвидшую рендерінг системи.

4

## Огляд системи

5

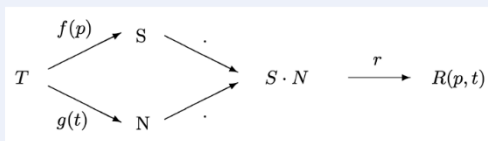
## Алгоритм проходження тестування



6



## Математичний опис задачі



Діаграма взаємозв'язку, де:

$T$  - це набір завдань,

$f(p)$ ,  $g(t)$  і  $r$  - відображення,

$t$  і  $p$  - параметри,

$R(p, t)$  - список рекомендацій.

7

## Механізм прогнозу даних

$$y = b_0 + b_1 \cdot x + b_2 \cdot x^2 + e$$

Поліноміальна регресія другого порядку, де:

$x$  - вектор значень  $x$ ;

$b$  - значення параметрів;

$e$  - вектор випадкових помилок;



## Item Response Theory

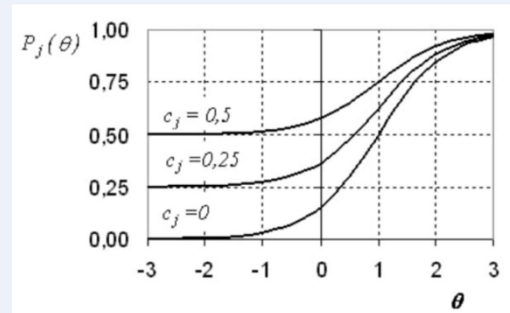
$$P(Y_{ij} = 1 | \theta_j, a_i, b_i, c_i) = c_i + (1 - c_i) \cdot \left( \frac{\exp(a_i(\theta_j - b_i))}{1 + \exp(a_i(\theta_j - b_i))} \right)$$

3PL Моделі Раша, де:

$a_i$  ( $-\infty < a_i < \infty$ ) - параметр дискримінативності;

$b_i$  ( $-\infty < b_i < \infty$ ) - параметр складності;

$c_i$  ( $0 \leq c_i \leq 1$ ) - параметр вгадування.



8

## Тестування системи для оцінки продуктивності

9

### Апаратне забезпечення

Було проведено тести на такому залізу, як:

- **Процесор:** *intel M, intel i5, intel i9, Apple M2*
- **Операційна пам'ять:** *4GB, 8GB, 16GB.*
- **Відеокарта:** *intel HD Graphics, intel UHD Graphics, AMD Radeon Pro 5500M, Apple M2.*

10



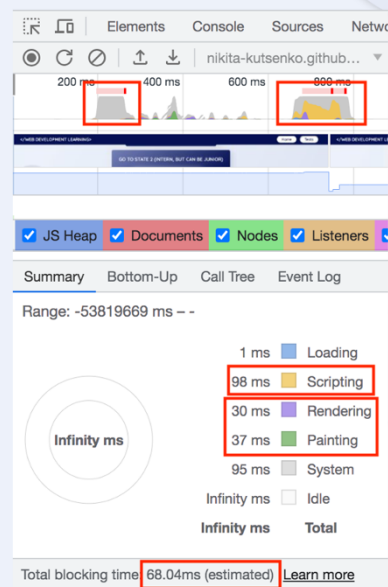
## Програмне забезпечення



11

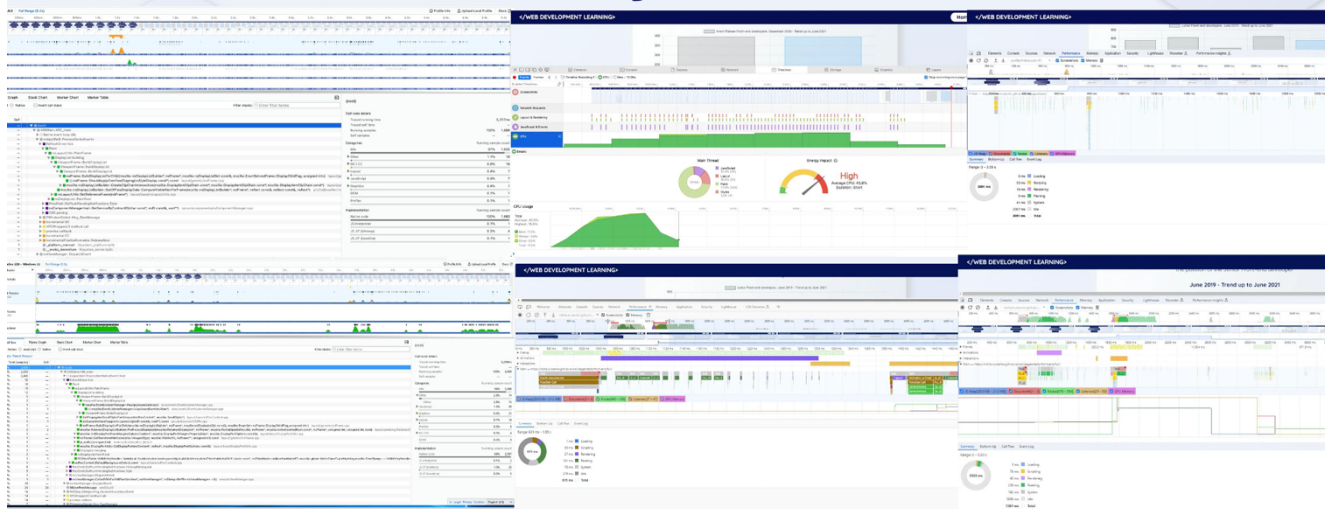
## Ключові показники

- Швидкість обробки даних;
- Швидкість рендерингу та відрисовки;
- Навантаження CPU;
- Блокери коду.



12

# Тестування



13

## Апробація результатів роботи

### Участь у конференціях:

- I Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»
- Науково-практична конференція «Проблеми комп'ютерної інженерії»
- V Міжнародна науково-практична конференція молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології» (SoftTech-2023)

### Наукова стаття у журналі категорії В:

Поперешняк С.В., Куценко М.О., Антоненко А.В. (2023). Система надання рекомендацій щодо вивчення технологій з веб-розробки. Таврійський науковий вісник. Серія: Технічні науки, Випуск 6 (2), 11-23.»

### Наукова стаття у Scopus:

Kutsenko, M., Yurchuk, I., (2022). An Intelligent System for Providing Recommendations on the Web Development Learning. In: Babichev, S., Lytvynenko, V. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2021. Lecture Notes on Data Engineering and Communications Technologies, vol 77. Springer, Cham. [https://doi.org/10.1007/978-3-030-82014-5\\_38](https://doi.org/10.1007/978-3-030-82014-5_38)

14

## Впровадження результатів роботи

Результати досліджень  
реалізовані та впровадженя у  
компанії ТОВ "СЕТТА  
КОМ'ЮНІКЕЙШНС".



15

## Висновки

1. Було проведено тестування системи рекомендацій, використовуючи різне апаратне та програмне забезпечення, для отримання результатів продуктивності системи, та дозволило знайти візуальні недоліки системи, а також побачити продуктивність роботи для різних типів користувачів.
2. Використовуючи результати тестування апаратного та програмного забезпечення, було проведено детальний аналіз різних показників, що дозволило знайти слабкі місця у системі, що навпаки перенавантажували систему на стороні користувача, та ускладнювало подальше проходження тестування.
3. Було обгрунтовано мінімально оптимальне апаратне та програмне забезпечення, яке зменшує час обробки результатів тестування та пришвидшує рендерінг системи, саме завдяки отриманим результатам тестування та статистикою використання програмного забезпечення за 2023 рік, а саме:
  - Процесор: Intel i3 або еквівалентний.
  - Оперативна пам'ять: мінімум 4 Gb.
  - Відеокарта: базовий рівень, такий як Intel HD Graphics або еквівалентний.
4. У подальшій розробці система надання рекомендацій буде адаптованою під кожного користувача, оскільки буде покращено продуктивність та час обробки результатів тестування завдяки вирішенню поточних проблем та створено мобільну версію системи.

16

The background is a light blue gradient. It features several semi-transparent circles of various sizes scattered across the page. In the top-left and bottom-right corners, there are clusters of small dark blue dots arranged in a triangular pattern.

**Дякую за увагу!**