

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА WEB-ДОДАТКУ ДЛЯ ВИВЧЕННЯ
ФРЕЙМВОРКУ JAVASCRIPT VUE.JS»**

Виконав: студент 4 курсу, групи ПД-44

спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Ролдугін Є.В.

(прізвище та ініціали)

Керівник

Яскевич В.О.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Нормоконтроль

(прізвище та ініціали)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ О.В. Негоденко

« ____ » _____ 2021 року

**З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ
РОЛДУГІНУ ЄВГЕНІЮ ВІТАЛІЙОВИЧУ**

1. Тема роботи: «Розробка web-додатку для вивчення фреймворку JavaScript Vue.js»

Керівник роботи Яскевич Владислав Олександрович, кандидат технічних наук затверджені наказом вищого навчального закладу від — «12» березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.1. Visual Studio Code як середовище для розробки додатку;

3.2. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

4.1. Проведення аналізу ринку додатків-аналогів;

4.2. Визначення вимог до веб-додатку;

- 4.3. Аналіз обраних інструментів;
- 4.4. Опис проектування додатку.
5. Перелік графічного матеріалу (назва основних слайдів)
- 5.1. Аналоги;
- 5.2. Порівняльний аналіз переваг та недоліків;
- 5.3. Програмні засоби реалізації;
- 5.4. Алгоритм роботи користувача;
- 5.5 Апробація результатів дослідження.
6. Дата видачі завдання 19.04.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.21 – 22.04.21	
2	Аналіз ринку додатків-аналогів	23.04.21 – 25.04.21	
3	Визначення вимог до додатку	26.04.21 – 05.05.21	
4	Підбір технологій для розробки	05.05.21 – 06.05.21	
5	Розробка додатку	07.05.21 – 21.05.21	
6	Вступ, висновки, реферат	21.05.21 – 23.05.21	
7	Розробка презентації додатку	23.05.21 – 24.05.21	
8	Попередній захист роботи	25.05.21	

Студент _____
 (підпис) (прізвище та ініціали)

Керівник роботи _____
 (підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 60 с., 32 рис., 21 джерело.

Ключові слова: РОЗРОБКА, ФРЕЙМВОРК, VUE, ДОДАТОК.

Об'єкт дослідження – розробка веб-додатків.

Предмет дослідження – веб-додаток для вивчення фреймворку JavaScript Vue.js.

Мета роботи – розвинути здібностей у розробці веб-додатків.

Методи дослідження – теоретичні, аналіз науково-методичної літератури з теми створення веб-додатків, емпіричні, спостереження, проведення апробації проведених досліджень з теми створення веб-додатків.

У роботі проведено аналіз вже існуючих додатків-аналогів. Проаналізовані їх плюси та мінуси використання. Розглянуто які інструменти та технології використовуються для розробки. Обрано найбільш простий та зручний спосіб розробки додатку для вивчення фреймворку JavaScript Vue.js.

Описана архітектура та проектування додатку.

Описана взаємодія клієнта з сервером та базою даних.

Галузь використання – так як додаток буде опублікований на домені, кожен, хто має пристрій з можливістю підключення до мережі Інтернет та браузер, може користуватись веб-додатком.

ЗМІСТ

ВСТУП	10
1 ТЕОРЕТИЧНІ ВІДОМОСТІ СТВОРЕННЯ ВЕБ-ДОДАТКІВ ДЛЯ ВИВЧЕННЯ ФРЕЙМВОРКУ VUEJS.....	13
1.1 Фреймворк та Web-технології	13
1.2 Фреймворк Vue.js.....	16
1.2.1 Що таке Vue.js	17
1.2.2 Переваги Vue.js	18
1.2.3 Недоліки Vue.js	22
1.2.4 Використання Vue.js.....	24
1.3 Приклади додатків-аналогів	24
1.4 Порівняльний аналіз та виявлення проблематики	28
2 АНАЛІЗ ОБРАНИХ ІНСТРУМЕНТІВ І ТЕХНОЛОГІЙ ТА ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ	31
2.1 Мова програмування JavaScript.....	31
2.2 Редактор вихідного коду Visual Studio Code.....	33
2.3 Проектування програмного засобу	34
3 РОЗРОБКА ТА ТЕСТУВАННЯ ВЕБ-ДОДАТКУ	37
3.1 Етапи розробки веб-додатку	37
3.2 Підключення бази даних	37
3.2.1 Реалізація реєстрації та авторизації користувачів.....	38
3.2.1.1 Реєстрація	38
3.2.1.2 Авторизація	39
3.3 Перевірка завдань.....	40

3.4 Розміщення web-додатку.....	42
3.5 Демонстрація розробленого додатку	43
3.6 Тестування розробленого додатку	46
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	53
ДОДАТКИ.....	55
Додаток А.....	55
Додаток Б	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HTTP – Протокол передачі даних

JS - JavaScript

API - Application Programming Interface

SSL - Secure Sockets Layer

DOM - Document Object Model

HTML - HyperText Markup Language

XML - Extensible Markup Language

JSON - JavaScript Object Notation

ВСТУП

У сучасному суспільстві на ринку технологій навички програмування відкривають двері для можливостей отримання високооплачуваної роботи та успішної кар'єри. Відповідно для цього потрібно опанувати певні навички роботи у сфері програмування. Як і вивчення будь-якої іншої мови, робота з кодом вимагає регулярної практики для засвоєння. Але людям не завжди легко знайти час, щоб практикуватися самостійно. Ось чому дуже корисно використовувати спеціальні веб-додатки для вивчення програмування.

Оскільки ринок постійно змінюється та розвивається, розробники, які присвячують освіті 30 хвилин на день, є більш гнучкими та мають більше шансів адаптуватися до нових тенденцій, ніж люди, які намагаються вчитися лише за підручниками, які були випущені десятки років назад.

На щастя, є безліч ресурсів — подкасти, блоги, веб додатки, які можна використовувати для покращення навичок програмування в будь-якому місці.

Сучасний веб-додаток — це унікальна розробка, орієнтована на вирішення певних завдань. Воно стає все більш затребуваним, оскільки дозволяє ефективніше вести бізнес-процеси [6].

Актуальність обраної теми. Актуальність дипломної роботи полягає в тому, що на сьогоднішній день фреймворки для розробки веб-додатків набирають все більшої популярності і стають невід'ємною частиною знань, які повинен мати розробник для успішного працевлаштування.

Веб-програмування — являється однією з найбільш перспективних напрямлень для розвитку. Будь-який бізнес, який має власний веб-додаток, за замовчуванням буде більш доступним, ніж його конкуренти.

Для створення інтерактивних і компактних веб-додатків необхідно знати чимало сучасних веб-стандартів.

Інтернет не стоїть на місці, а швидко розвивається, як результат, створюється велика кількість нових сервісів. На даний момент інтернет перетворився у всесвітню павутину з нескінченною кількістю додатків, сервісів, технологічних

рішень, став тією ланкою, яка з'єднує між собою практично усі бізнеси та організації світу.

Зараз можна спостерігати як з кожним днем, все більше і більше компаній переводять свій бізнес в електронний вигляд . Організації створюють свої корпоративні домени, свої веб-сторінки, інтернет став місце, де вони рекламують свої товари і, звичайно, починають продавати свої товари онлайн.

Щоб справлятися з стрімким зростанням бажаючих мати свій веб-сайт, потрібні розробники. Їх завдання:

- розробляти веб-сайти;
- верстка по макету;
- підключати базу даних;
- займатись редагуванням контенту;
- розробляти веб-сервіси;
- розробляти дизайн;
- створювати веб-додатки для нових напрямлень;
- створювати технічні передумови для продажів онлайн і просування товарів і послуг;
- займатись безпекою веб-додатків;
- покращувати вже існуючі додатки;
- автоматизувати публікацію та оновлення контенту [12].

Мета роботи — розвинення здібностей у розробці веб-додатків.

Об'єкт дослідження — розробка веб-додатків.

Предмет дослідження — веб-додаток для вивчення фреймворку JavaScript Vue.js.

Щоб досягнути поставленої мети, слід виконати такі завдання:

1. Провести аналіз недоліків та переваг додатків-аналогів;
2. Обрати програмні засоби реалізації та технології для розробки;
3. Розробити структуру уроків та завдань для перевірки вивченого матеріалу;
4. Розробити додаток;

5. Провести тестування створеного веб-додатку.

Під час написання роботи було використано такі *методи дослідження*:

- теоретичні: аналіз науково-методичної літератури з теми створення веб-додатків;
- емпіричні: проведення апробації проведених досліджень з теми створення веб-додатків.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ СТВОРЕННЯ ВЕБ-ДОДАТКІВ ДЛЯ ВИВЧЕННЯ ФРЕЙМВОРКУ VUEJS

1.1 Фреймворк та Web-технології

Фреймворк (для інформаційних систем) — це програмна структура системи, яка полегшує, пришвидшує розробку та об'єднання різних компонентів які потрібні для розробки. Бібліотеки об'єднують набір підпрограм схожої функціональності, а фреймворк тримає у собі великий набір бібліотек різних за призначенням. Широкого вжитку також набув вираз «каркас», слід зазначити, що автори не задумуючись використовують його, як основний варіант, взагалі не беручи до уваги англomовний аналог. Каркаcний підхід являє собою спосіб розробки програм, згідно підходу, конфігурація програми будується за допомогою двох частин: перша — каркас, в якому конфігурації не пов'язані між собою і міститься гніздо, в якому розміщується друга частина — в якій знаходяться змінні модулі та точки, які, у свою чергу, використовуються для розширення [4].

Щоб дати відповідь на запитання, що має входити до фреймворка, щоб останній розглядався як WEB-технологія, необхідно розглянути сучасні види фреймворків:

- Фреймворки програмних систем;
- Фреймворки веб-додатків;
- Фреймворки моделей.

Фреймворк — це основа системи, що має можливість підключати додаткові програми, мови сценаріїв — тобто все, що може полегшити розробку й об'єднання різних компонентів. Фреймворк відрізняється від поняття бібліотеки тим, що бібліотека може бути використана в програмному продукті просто як набір підпрограм з близькою функціональністю, не впливаючи на архітектуру програмного продукту і не накладаючи на неї ніяких обмежень. У той час як фреймворк диктує правила побудови архітектури додатку, задаючи на початковому етапі розробки поведінку за замовчуванням, каркас, який потрібно 14 буде

розширювати і змінювати відповідно до зазначених вимог. До цього виду відносять фреймворки для WEB [19].

Фреймворк веб-додатків або веб-фреймворк — це програмне забезпечення, створене для підтримки розробки динамічних веб-сайтів, веб-сервісів та веб-додатків [12].

Веб-фреймворк (WF) підтримує створення, розробку та публікацію веб-додатків та веб-сайтів. Підтримка може включати веб-сервіси, API та інші ресурси. Веб-фреймворки — це програмні фреймворки, які включають в себе стандартний і доступний спосіб створення та розробки веб-додатків.

Хоча додатки з підтримкою Інтернету є звичним явищем у Всесвітній павутині і більшість людей користуються цими ресурсами щодня, розвиток засобів та інструментів за допомогою яких створюють відповідні веб-додатки все ще перебувають у стані розвитку. Ось чому так важливо мати веб-фреймворк, який би виступав системою підтримки для розробки та розгортання таких додатків. Фреймворк веб-додатків є одним із таких інструментів, тому в роботі буде описано створення веб-додатку який відповідає найсучаснішим веб-стандартам.

Веб-фреймворк — це бібліотека коду, яка робить веб-розробку швидшою та простішою, надаючи основні шаблони для створення надійних та масштабованих веб-додатків. Веб-фреймворки існують, щоб спростити розробнику створення веб-програми [18].

Веб-фреймворк відрізняється від веб-сервера тим, що веб-сервер насправді запускає веб-додаток, тоді як веб-фреймворк більше нагадує віртуальну базу даних або бібліотеку, яка допомагає пришвидшити процес розробки та написання.

Фреймворки надають функціональність коду через розширення для завершення повсякденних операцій, необхідних для запуску веб-додатків. Ці операції включають:

1. Маршрутизація URL-адрес;
2. Управління та перевірка форми вводу;
3. Налаштування HTML, XML, JSON та інших продуктів із механізмом шаблонування;

4. Конфігурація з'єднання з базою даних та маніпуляція даними за допомогою об'єктно-реляційного картографування (ORM);

5. Веб-безпека від міжсайтової підробки запитів (CSRF), SQL Injection, міжсайтових сценаріїв (XSS) та інших частих зловмисних атак;

а. Існує кілька веб-фреймворків, розроблених для полегшення кодування та розробки програм для тих, хто працює в цій галузі. Серед найбільш популярних та успішних веб-фреймворків:

- Ruby on Rails;
- Django;

Ці фреймворки спрощують роботу з кодом HTML, Ruby та Python, коли мова йде про створення веб-додатків та веб-сервісів. Існує безліч речей, які можна зробити за допомогою веб-фреймворку, наприклад, генерація HTML, формування шаблонів за замовчуванням та можливість доступу до бібліотеки даних про кодування, щоб зробити код простим та автоматизованим кожного разу, коли розробник створює новий веб-додаток.

Починаючи роботу з веб-фреймворком, слід врахувати:

- Взаємодія з базами даних — багато веб-фреймворків мають можливість застосовувати взаємодію з базами даних у всеосяжному пакеті.
- Генерація HTML — фреймворки мають можливість генерувати HTML-коди за допомогою шаблонів, що може полегшити розробникам використання під час встановлення значень та встановлення стандартів.
- Маршрутизація — більшість фреймворків можуть обробляти можливість зіставити URL-адресу, яку запитує користувач, з кодом, який відповідає їй [16].

Під час роботи з веб-фреймворками можна виконувати різноманітні завдання, починаючи від створення шаблонів та доступу до бази даних, до управління сеансами та повторного використання коду.

Виділяють п'ять типів веб-фреймворків:

- Hybrid;

- RIA-based.
- Component-based;
- Request-base;
- Meta;

Request-based — це фреймворки, які потрібні для обробки вхідних запитів. Стан зберігається за допомогою так званих серверних сесій. Приклади: Grails, Struts, Ruby on Rails, Django,.

Component-based — це фреймворки, які призначені для абстрагування запитів всередині компонентів та самостійно відстежують стан. Дані каркаси мають багато спільного зі стандартними програмними графічними інтерфейсами.

Hybrid-based — це фреймворки, які тримають в собі Component-based та Request-based фреймворки, вони контролюють усі дані і сформований логічний потік у заснованої моделі. Розробники повністю контролюють посилання, форми, pathinfos і cookies. Проте замість того, відобразити дії та контролери відразу до запиту, гібридні фреймворки створені для забезпечення об'єктної моделі компонентів, поведінка такої моделі у різних ситуаціях схожа, наприклад, у ситуаціях, як окремі сторінки, запити, фрагменти сторінок порталу та віджети. Такі компоненти можуть з легкістю розділятися та приєднуватися в інші, вже існуючі проекти.

Meta-based — фреймворки, що мають цілу низку основних інтерфейсів, які використовуються для загального обслуговування і основну частину, яка легко розширюється з метою інтегрування компонентів і служб. До них відноситься Keel. RIA-based — фреймворки створені для прозробки Rich Internet Applications (RIA) — фреймворки, створені для повноцінних додатків, які функціонують безпосередньо в браузері, прикладом може слугувати Flex.

Найбільшої популярності набули Request-based та Component-based фреймворки [5].

1.2 Фреймворк Vue.js

Хоча JavaScript є повноцінною мовою програмування, його екосистема означає набагато більше, ніж сама мова. Такі інструменти, як фреймворки, значно полегшують життя розробника, забезпечуючи основу роботи для подальшої розробки.

Якщо стежити за ринковими тенденціями JS, можна побачити, що основними конкурентними претендентами є React та Angular. Останнє опитування Stack Overflow підтверджує перевагу використання React над Angular, хоча саме Angular використовується більше серед професіоналів. Але не слід забувати про Vue.js. З 15,2 відсотками Vue.js знаходиться на 6 місці. Використання цього фреймворку значно зросло (229 відсотків), у порівнянні з застосуванням React (34 відсотки) [3].

Отже, у цьому розділі буде розглянуто, що таке Vue.js, його екосистема, переваги та недоліки. А також порівняння інструментів та функції Vue.js з його конкурентними структурами, для об'єктивного розуміння його якостей.

1.2.1 Що таке Vue.js

Історія Vue.js починається в 2013 році, коли Еван Ю працював у Google, створюючи безліч прототипів прямо в браузері. Для цього Еван використав зручні фрагменти з інших фреймворків, з якими працював, і офіційно випустив Vue.js у 2014 році.

Vue.js — це прогресивний фреймворк для JavaScript, який використовується для побудови веб-інтерфейсів та односторінкових додатків [3]. Не тільки для веб-інтерфейсів, Vue.js також використовується як для розробки настільних ПК, так і для мобільних додатків з фреймворком Electron. Розширення HTML і база JS швидко зробили Vue улюбленим інтерфейсним інструментом, про що свідчать прийняття таких гігантів, як Adobe, Behance, Alibaba, Gitlab та Xiaomi.

Назва фреймворку Vue — англійською мовою фонетично схоже зі словом *view*, і воно відповідає традиційній архітектурі *Model-View-Controller* (MVC). Простіше кажучи, *view* — це інтерфейс користувача програми / веб-сайту. Але, MVC не означає, що Vue.js не можна використовувати з іншим архітектурним

підходом, як *Архітектура на основі компонентів* (CBA), що використовується в React [3].

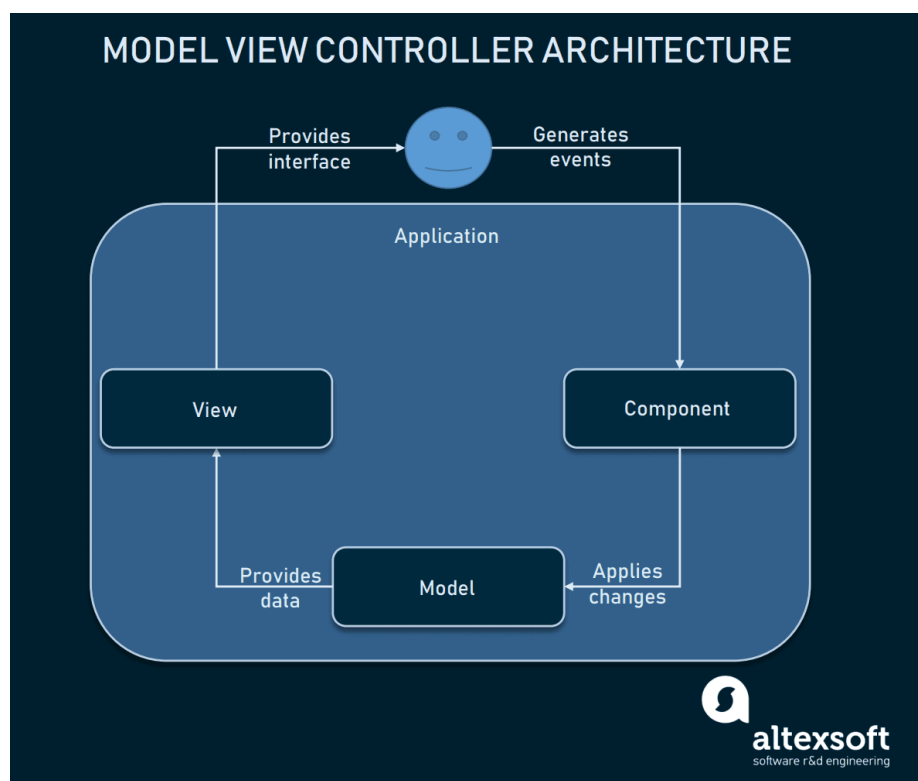


Рисунок 1 – Архітектура контролера перегляду моделі [1]

У нинішній формі Vue.js, розроблений Еваном Ю на постійній основі, отримував вигоду від внесків членів громади та фінансувався переважно від Patreon. Без фінансової підтримки таких підприємств, як Facebook (React) та Google (Angular), Vue все ще домігся широкого поширення на Github.

1.2.2 Переваги Vue.js

Як і будь-яка технологія, яка набирає популярність, програмування Vue.js викликає суперечки у суспільстві. І є причини, чому Vue став другим найулюбленішим фреймворком у 2019 році. Переваги фреймворку:

- Оптимальний розмір

Завантажений zip-файл із фреймворком важить 18 КБ. Малий розмір програми - це не тільки швидке завантаження та встановлення бібліотеки, а й позитивний вплив на SEO та UX.

- Віртуальний рендеринг і продуктивність DOM

Document Object Model (DOM) — це те, з чим, ймовірно, можна зіткнутися при візуалізації веб-сторінок. DOM — це представлення HTML-сторінок з їх стилями, елементами та вмістом сторінки як об'єктами [3]. Об'єкти, що зберігаються як деревоподібна структура, створюються браузером під час завантаження сторінки.

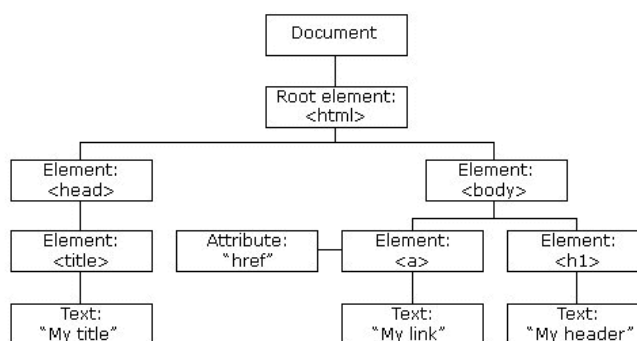


Рисунок 2 – Дерево DOM об'єктів [3]

Коли користувач взаємодіє зі сторінкою, об'єкти змінюють свій стан, так що браузер повинен оновити інформацію та відобразити її на екрані. Але, оновлення всього DOM є громіздким. Заради швидкості Vue.js використовує віртуальний DOM: можна уявити це як копію оригінального DOM, який з'ясовує, які елементи потрібно оновлювати, не відтворюючи цілий DOM. Цей підхід робить візуалізацію сторінок досить швидкою та покращує продуктивність програми.

Ефективність є одним із ключових факторів, який може зумовити вибір фреймворку. Фактичні показники наведені на сторінці порівняння Vue. Наприклад, під час тестування компонентів DOM, пов'язаних із оновленими даними, Vue.js здається більш продуктивним, ніж Angular та React.

Потрібно зауважити, що будь-яке тестування продуктивності залежить від конкретного випадку: у реальному житті більша частина продуктивності відбувається завдяки оптимізації та якості коду.

- Реактивна двостороння прив'язка даних

Ще однією перевагою маніпуляцій DOM є двостороння прив'язка даних, успадкована Vue від Angular. Двостороння прив'язка даних — це зв'язок між оновленнями даних моделі та поданням (UI) [3]. Зв'язані компоненти містять дані, які можна час від часу оновлювати. За допомогою двосторонньої прив'язки даних легше оновлювати пов'язані компоненти та відстежувати оновлення даних.

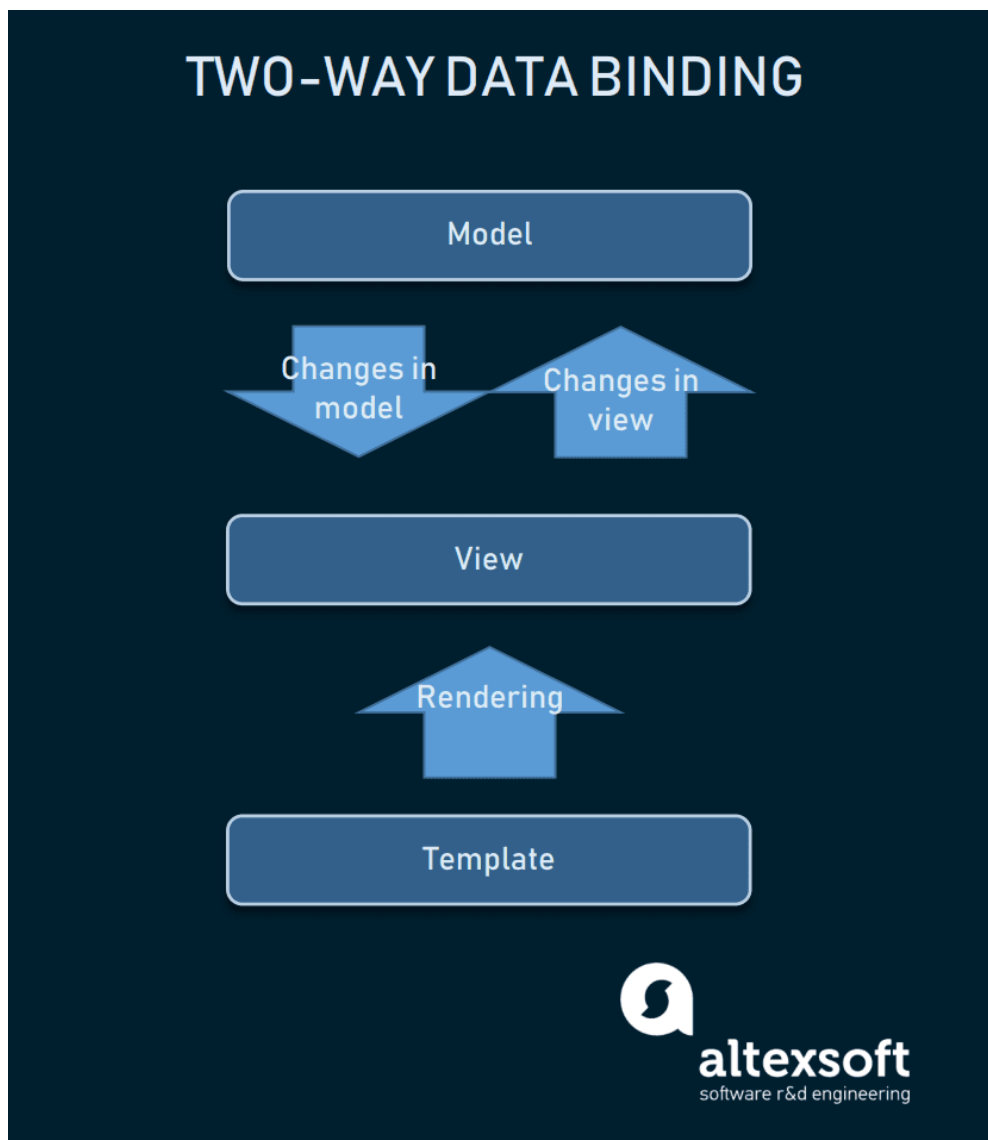


Рисунок 3 – Оновлення даних у зв'язаних компонентах [8]

У Vue пов'язані дані оновлюються реактивно, як і об'єкти DOM, що чудово підходить для будь-якої програми, яка вимагає оновлення в режимі реального часу. З точки зору розробників, реактивність Vue зробить оновлення даних зрозумілішим та простішим для заповнення. Існує кілька правил, що застосовуються для реагування реактивності, які будуть описані в розділі недоліків.

- Однофайлові компоненти та читабельність

Кожна частина програми / веб-сторінки у Vue є компонентом. Компоненти представляють інкапсульовані елементи інтерфейсу. У Vue.js компоненти можна писати у форматі HTML, CSS та JavaScript, не розділяючи їх на окремі файли.

Розбиття коду програми насправді є архітектурним підходом, який називається Component Based Architecture (CBA), і він також використовується в Angular та React. Такий архітектурний підхід має масу переваг:

- 1) Повторне використання компонентів. Інкапсульовані компоненти — це в основному фрагменти коду, які можна використовувати повторно як шаблони для подібних елементів системи.

- 2) Читабельність коду. Оскільки всі компоненти зберігаються в окремих файлах (а кожен компонент — це лише один файл), код легше читати та розуміти, що полегшує обслуговування та виправлення.

- 3) Зручне використання при модульному тестуванні. Тестування модулів — це перевірка якості, спрямована на перевірку того, як найменші частини програми працюють самі по собі. Наявність компонентів значно спрощує це завдання [8].

- Інтеграційні можливості та гнучкість

Важливим аспектом будь-якої нової технології є можливість інтеграції з існуючими додатками. З Vue.js це дуже просто, оскільки він покладається лише на JavaScript і не вимагає жодних інших інструментів для роботи.

Vue також дозволяє писати шаблони різними способами: за допомогою HTML, JS або JSX (розширення синтаксису JavaScript). Своїми компонентами та полегшеною природою Vue можна використовувати практично в будь-якому

проекті. Слід зазначити, що перехід з React або Angular не складає великої проблеми, оскільки внутрішня організація Vue в основному є сумішшю двох.

- Тверда інструментальна екосистема

За 7 років існування Vue.js отримав потужний набір інструментів для роботи. Випуск Vue CLI 3 — це повний рерайт, який забезпечує купу нових функцій. Vue CLI 3 підтримує Babel і TypeScript, може надавати модульне тестування, наскрізні інструменти тестування та систему встановлення плагіна. І якщо цього недостатньо, Vue також має власні інструменти налагодження браузера, візуалізатор сервера та менеджер стану.

- Легкий у навчанні

Інструмент може досягти масового прийняття лише тоді, коли це легко зрозуміти, що може бути у випадку з вивченням Vue.js. Щоб розпочати роботу з кодом, Vue не вимагає глибоких знань бібліотек, JSX та TypeScript, як це часто буває з іншими інтерфейсними технологіями. Для початку знадобляться лише базові знання HTML, CSS та JavaScript.

Найпопулярніші редактори коду, такі як Sublime text, Visual Studio та Atom, підтримують Vue, що полегшує його використання.

- Коротка інструкція

Якщо при вивченні фреймворку можуть виникнути якісь проблеми, можна скористатися документацією Vue. Відомості про роботу з фреймворком структуровані і охоплюють всі можливі теми, від встановлення до більш поглиблених речей, таких як реактивність та масштабування програми.

1.2.3 Недоліки Vue.js

- Мовний бар'єр

Впровадження Vue на таких підприємствах, як Xiaomi та Alibaba, допомогло популяризувати структуру та створило попит на ринку праці. Зважаючи на те, що Vue.js стає все більш популярним у Китаї, значна частина його змісту та обговорень, як не дивно, є китайською.

Великий китайський брандмауер робить у цій країні все трохи інакше, оскільки в даний час там немає багатьох популярних ресурсів. Це ускладнює навчання та використання React або Angular. Vue є кращим варіантом.

Отже, при пошуку відомостей про Vue, можна зіткнутися з обговореннями на форумі, описами плагінів та інструкціями китайською мовою. Це може бути проблемою для більшості користувачів.

- Складність реактивності

Хоча це зручний інструмент для синхронізації компонентів, є одне питання щодо того, як працює система реактивності, як її ще називають.

Програма Vue.js складається з компонентів, з якими користувач може взаємодіяти. Кожен компонент має свого спостерігача, який рендерує дані кожного разу, коли користувач запускає компонент. Система реактивності відображає лише ті фрагменти даних, які були ініційовані. Під час роботи можуть виникнути помилки зчитування даних.

Однак ця проблема є дуже популярною, тому вона розглядається в документації Vue, надаючи вказівки щодо правильного налаштування реакційної здатності.

- Відсутність підтримки масштабних проектів

Vue.js — все ще “молода” структура. Розмір спільноти та команди розробників менший у порівнянні із командою Angular. Також вона не користується фінансовою підтримкою великих підприємств.

- Обмеженість ресурсів

Незважаючи на те, що екосистема досить широка, і є всі необхідні інструменти, щоб розпочати розробку з Vue, вона все ще не така велика, як React або Angular. Для наочного результату, можна порівняти кількість плагінів, доступних для React та Vue.js: різниця у сотнях. Існуючі плагіни, які можна використовувати з іншими фреймворками, часто також не підтримуються, але це може бути лише питанням часу.

1.2.4 Використання Vue.js

Враховуючи всі концептуальні та технічні аспекти Vue.js, виникає питання: “Де краще використовувати Vue.js”. Окрім прямого призначення, створення односторінкових додатків та роботи на веб-сторінках, він також підходить для ряду завдань.

- Робота з прототипами

Перш за все, Vue.js був розроблений для створення прототипів. За умови належної прив’язки даних зручно працювати з великою кількістю анімацій, інтерактивних елементів та графіки.

- Зосередження уваги на інтерфейсі користувача

Vue.js постійно фокусується на роботі з користувальницьким інтерфейсом, оскільки для роботи з ним потрібні лише HTML, CSS та JS, не надто багато специфічних для Vue матеріалів.

- Інтеграційний попит

Для додавання інтерактивності до створеної програми, Vue стане одним із найзручніших засобів. Оскільки він заснований на JavaScript, його можна легко інтегрувати в будь-який проект за допомогою JS. Більше того, він сумісний з багатьма внутрішніми технологіями та фреймворками, такими як Laravel, Express, Rails та Django.

Навіть враховуючи мінуси Vue.js, його все одно можна використовувати у великих проектах.

1.3 Приклади додатків-аналогів

Використання комп’ютерів у сучасному світі стало невід’ємною частиною як навчання так і роботи, але більшість людей не розуміють, як вони працюють. Завдяки вивченню програмування учні та студенти формують основні навички грамотності, отримують розуміння логіки та послідовності. Веб-додатки для вивчення програмування, зокрема фреймворків, підтримують проектне навчання та

дають можливість створювати, видаляти, реміксувати та майструвати власні унікальні розробки. Все це можна навчитися за допомогою таких веб-додатків:

1. SoloLearn

SoloLearn — програма, доступна як для пристроїв iOS, так і для Android, завдяки якій можна навчитися програмуванню з будь-якого місця. Додаток пропонує низку різних курсів по вивченню таких мов, як JavaScript, Python, Java тощо. Наявна безкоштовна пробна версія, а також додаток надає можливість купувати щомісячні та річні тарифи [7].

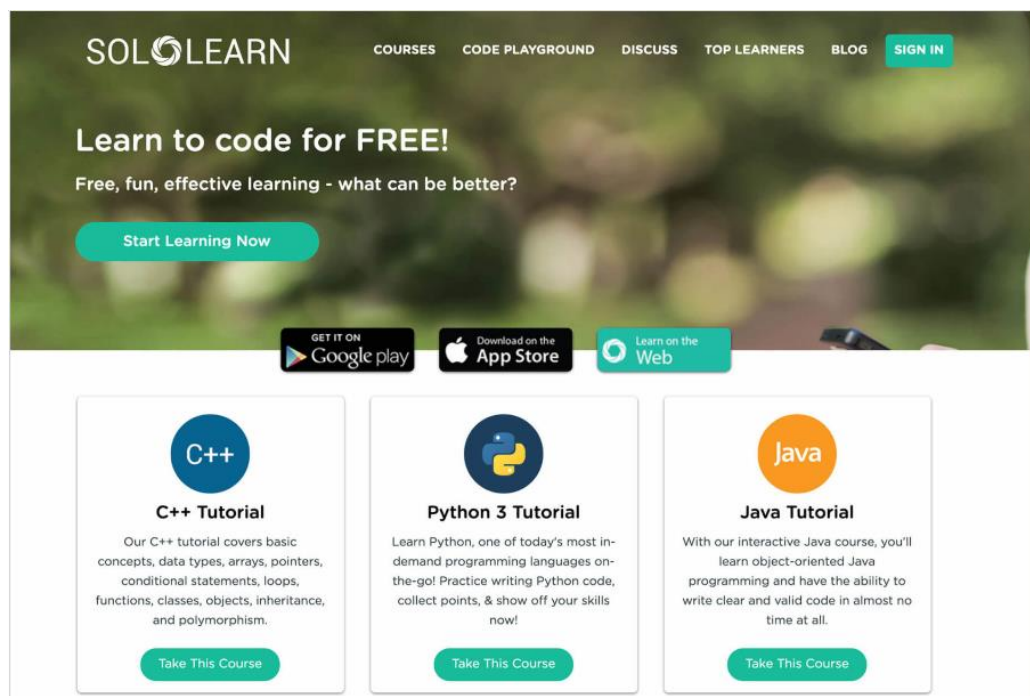


Рисунок 4 – Програма SoloLearn [2].

2. Encode

Encode- це програма для Android, яка пропонує уроки програмування поетапно. Додаток має завдання з програмування, які потрібно виконувати послідовно для того щоб відкривати нові досягнення. Додаток містить практичні завдання для вивчення HTML, CSS, JavaScript та Python. Крім того, програма має автономний режим, що дає змогу навчатися без підключення до Інтернету [10].

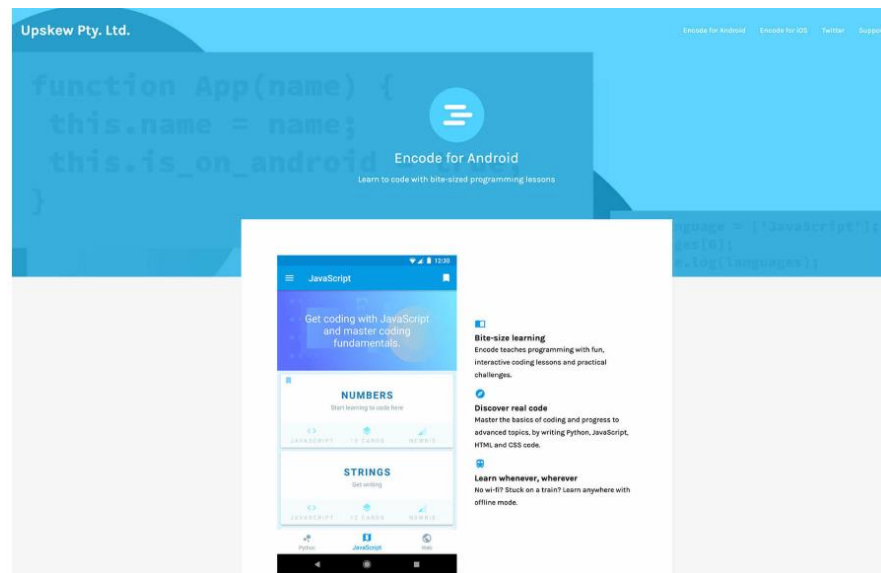


Рисунок 5 – Веб-додаток Encode [10].

3. Mimo

Mimo — це програма для вивчення програмування, яка дозволяє розвивати відповідні навички, які допоможуть розробляти різні програми або ігри, створити веб-сайт або стати хакером. Виходячи з уподобань та інтересів, можна отримати персональний трек, який навчить необхідним навичкам. Доступний як для пристроїв iOS, так і для Android. Можливості: вивчення JavaScript, Ruby, Swift, C, C ++ та інших популярних мов [13].

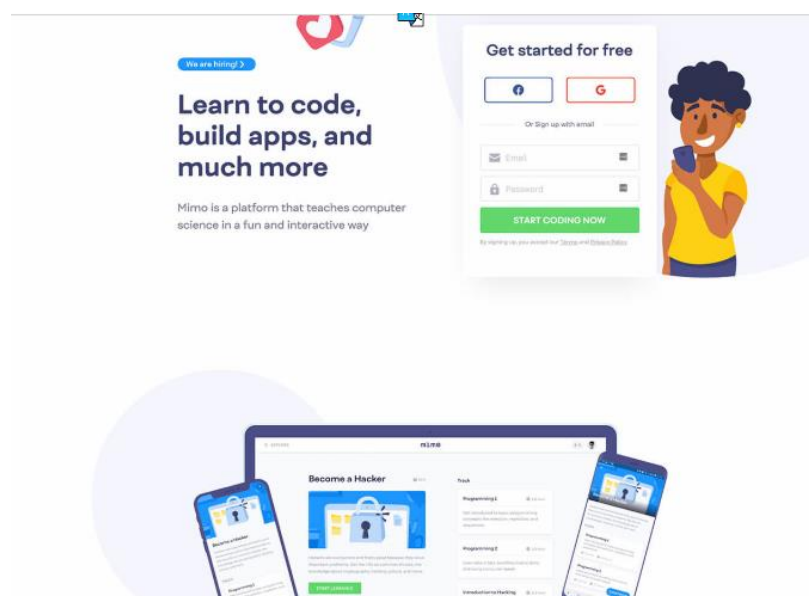


Рисунок 6 – Програма для вивчення програмування – Mimo [13].

4. Grasshopper

Grasshopper — це безкоштовна програма для iOS та Android, де можна навчитися мові програмування JavaScript, вирішуючи головоломки. Додаток простий у використанні, оскільки робочий код представлений у вигляді зручних блоків, що дозволяє концентрувати увагу на логічній послідовності та обдумувати написання програми, не витрачаючи час на виправлення синтаксичних помилок та структурування коду. Ресурс містить зворотний зв'язок у реальному часі [15].

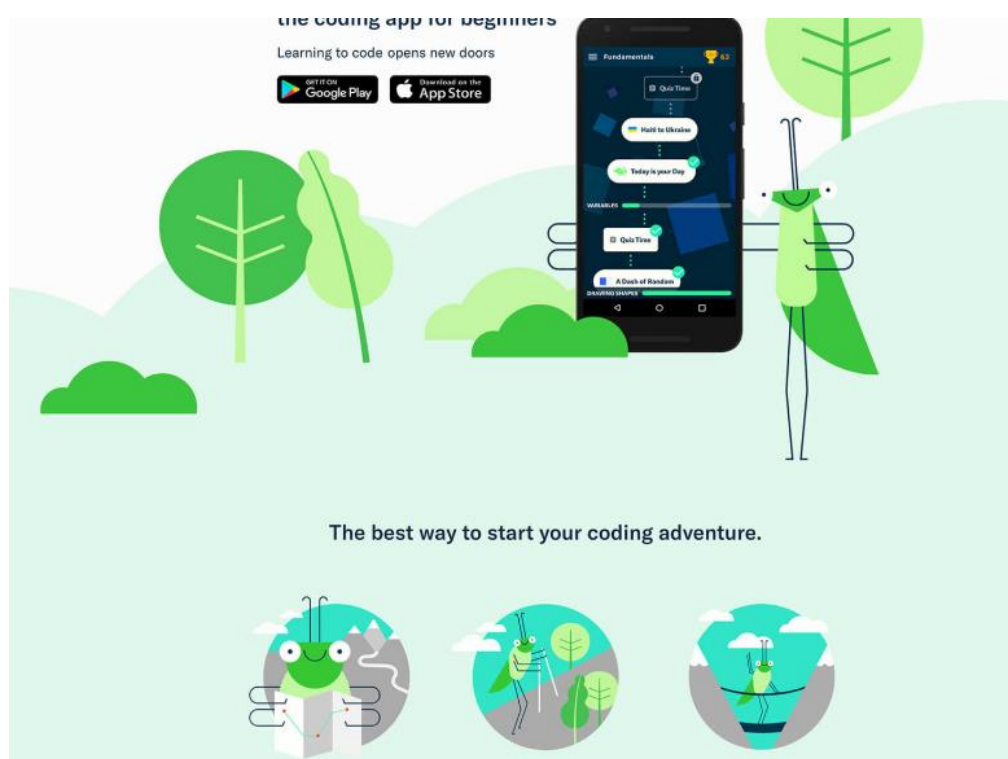


Рисунок 7 – Веб-ресурс Grasshopper [15].

5. Tynker

Незважаючи на те, що Tynker в першу чергу призначений для дітей та батьків чи вихователів, які хочуть навчити програмуванню своїх дітей та учнів, у цьому додатку можна багато чому навчитися і студентам. По-перше, додаток підтримує не тільки вивчення програмування для створення дизайну ігор, але також основи робототехніки та більш просунуті мови, такі як JavaScript та Python. По-друге, веб-

ресурс містить вікторини, редактори рівнів та програми для створення персонажів. Додаток доступний для пристроїв iOS [11].



Рисунок 8 – Додаток Tynker [11].

1.4 Порівняльний аналіз та виявлення проблематики

Усі описані веб-додатки мають велику популярність серед користувачів, які спрямовані на вивчення програмування. Але кожна програма має як переваги так і недоліки. Які можна виразити в порівняльному аналізі за різними критеріями відносно аспектів вивчення мов програмування (див. табл. 1.1).

Таблиця 1.1

Порівняльний аналіз сучасних веб-додатків для вивчення програмування

Назва	Переваги	Недоліки
SoloLearn	1. Великий функціонал безкоштовної версії.	1. Висока вартість оформлення Pro версії.

	<p>2. Простий у використанні.</p> <p>3. Навчання можна проходити будь-де і з будь-якого гаджету без підключення до мережі Інтернет.</p>	<p>2. Тільки англomовний інтерфейс.</p>
Mimo	<p>1. Програмування з нуля.</p> <p>2. Створення додатків.</p> <p>3. Створення ігор.</p> <p>4. Створення веб-сайтів.</p>	<p>1. Відсутність українomовного інтерфейсу.</p> <p>2. Безкоштовна версія лише 7 днів.</p>
Grasshopper	<p>1. Різноманітний функціонал;</p> <p>2. Полегшення роботи з архітектурними схемами;</p> <p>3. Цікаві завдання;</p> <p>4. Кілька сценаріїв роботи з даною програмою.</p>	<p>1. Складний інтерфейс.</p> <p>2. Не достатня кількість теоретичних даних.</p>
Tynker	<p>1. Простий інтерфейс.</p> <p>2. Цікаві завдання.</p>	<p>1. Англomовний інтерфейс.</p> <p>2. Створений для дітей.</p>

	<p>3. Навчання можна проходити будь-де і з будь-якого гаджету при наявності підключення до мережі Інтернет.</p>	
Encode	<p>1. Простий інтерфейс.</p> <p>2. Велика кількість різноманітних цікавих завдань.</p> <p>3. Наявна безкоштовна версія.</p>	<p>1. Англomовний інтерфейс.</p> <p>2. Велика вартість pro-версії.</p>

Порівняльний аналіз виявив, що основними недоліками майже усіх веб-додатків є англomовний інтерфейс, що може викликати труднощі у людей, які не мають мінімальних навичок з англійської мови та платні додаткові інструменти для поглибленого вивчення відповідного матеріалу. Тому на основі виявлених недоліків було створено веб-додаток для вивчення js фреймворку vueJs, який компенсує вищеперераховані недоліки.

2 АНАЛІЗ ОБРАНИХ ІНСТРУМЕНТІВ І ТЕХНОЛОГІЙ ТА ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ

2.1 Мова програмування JavaScript

JavaScript — це мова сценаріїв, яка використовується для створення та управління динамічним вмістом веб-сайту, тобто будь-чого, що переміщується, оновлюється чи змінюється на екрані, не вимагаючи перезавантаження веб-сторінки вручну. За допомогою JavaScript можна виконувати такі функції:

- використання анімованої графіки;
- створення слайд-шоу з фото;
- розробка інтерактивних форм.

Доречним способом для пояснення поняття роботи JavaScript є — демонстрація певних веб-функцій, якими люди користуються щодня і, ймовірно, сприймають їх як належне — наприклад, автоматичне оновлення хронології Facebook або представлення безліч варіантів пошукових запитів Google на основі декількох введених букв. В обох випадках демонструється робота JavaScript.

При навчанні написання коду, зазвичай, користувачі починають роботу з HTML та CSS. І уже потім переходять до JavaScript. Що має сенс! Саме ці три елементи разом складають основу веб-розробки:

- HTML — це структура сторінки - заголовки, основний текст, будь-які зображення, які потрібно помістити на сайт.
- CSS контролює вигляд цієї сторінки (використання для налаштування шрифтів, кольорів тла тощо)
- JavaScript — третій елемент. Після створення структури (HTML) та естетичного вигляду (CSS), JavaScript робить сайт або проект динамічним [4].

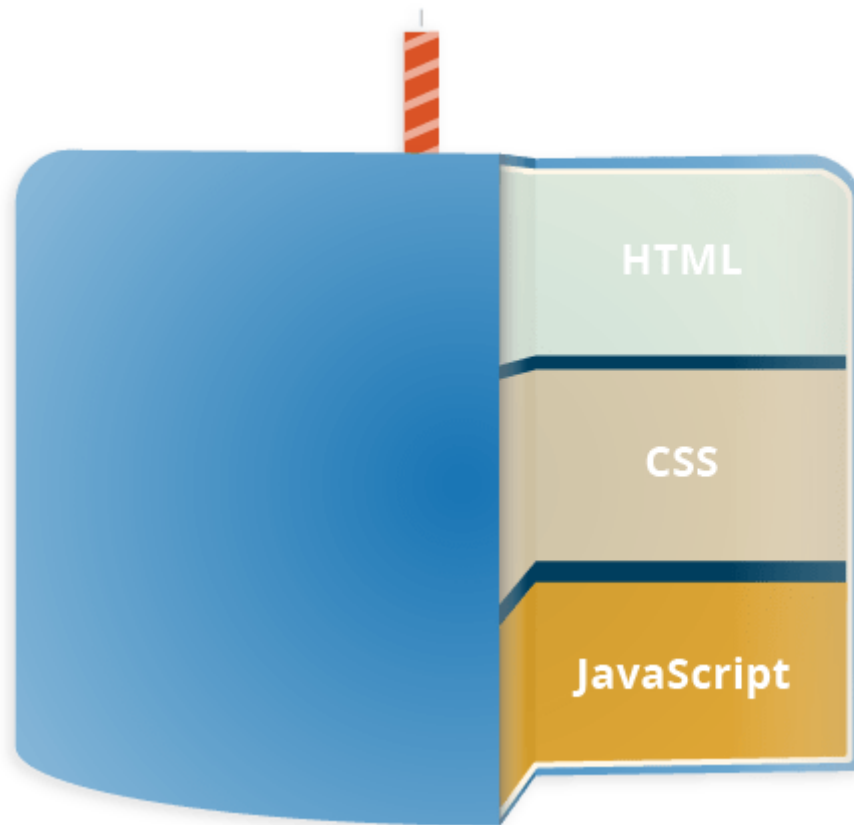


Рисунок 9 – Елементи веб-розробки [4]

JavaScript — це "мова сценаріїв". Мови сценаріїв — це мови кодування, які використовуються для автоматизації процесів, які користувачі мали б виконувати самостійно, поетапно [14]. Мови сценаріїв дозволяють включати в web-сторінки програмний код, який задасть поведінку web-сторінки в браузері при роботі з ними користувачів.

Оскільки JavaScript є невід'ємною частиною веб-функціональності, усі основні веб-браузери мають вбудовані механізми, які можуть відображати JavaScript. Це означає, що команди JS можна вводити безпосередньо в документ HTML, і веб-браузери зможуть їх опрацювати. Іншими словами, використання JavaScript не вимагає завантаження будь-яких додаткових програм або компіляторів.

Основні функції JavaScript:

- Додавання інтерактивності до веб-сайтів;

- Розробка мобільних;
- Створення ігор на основі веб-браузера;
- Веб-розробка сайтів.

2.2 Редактор вихідного коду Visual Studio Code

Visual Studio Code — редактор вихідного коду, розроблений компанією Microsoft.

Програма має безкоштовний доступ і підтримується усіма актуальними операційними системами: Windows, Linux и macOS. VS Code представляє звичайний текстовий редактор з можливістю підключення різних плагінів, що дає можливість працювати з усіма можливими мовами програмування для розробки будь-якого ІТ-продукту.

Особливості VS Code:

1) VS Code дозволяє розробляти як консольні додатки, так і додатки з графічним інтерфейсом з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки та веб-служби.

2) У редакторі вбудований відладчик, інструменти для роботи з Git і засоби рефакторингу, навігація по коду, автодоповнення типових конструкцій і контекстні підказки.

3) Продукт підтримує розробку для платформ ASP.NET і Node.js, та вважається зручним засобом, який дозволяє обійтися без повної інтеграції середовища розробки.

4) Підтримка великої кількості мов, таких як C++, C#, Python, PHP, JavaScript і багато інших.

Можливості Visual Studio Code:

1. Вбудовані інструменти інтеграції з GitHub, GIT, а також Visual Studio Team Services для швидкого тестування, збирання, упакування і розгортання різних типів додатків.

2. Зручна робота з Unity-проектами.
3. Робота з Mono і Node.js за допомогою вбудованого відладчика.
4. Підтримка TypeScript і JavaScript.
5. Публікація створених додатків в Microsoft Azure через сервіс Visual Studio Team Services.
6. Підтримка майже всіх мов програмування.

Переваги VS Code:

1. Велика база налаштувань (як всієї програми, так і інтерфейсу).
2. Розширена бібліотека доповнень і готових рішень.
3. Багатофункціональність (редактор підтримує майже всі мови, використовувані для створення додатків).
4. Простота і гнучкість.

Отже, дивлячись на всі переваги Visual Studio Code можна зробити висновки, що використання цього редактору вихідного коду як середовища для розробки веб-додатку є доречним, так як ця програма має безліч можливостей та особливостей, які не завжди можуть бути доступними в аналогах.

2.3 Проектування програмного засобу

Цикл тем. Уроки поділені на 3 розділи:

1. Основи Vue.js, де користувач протягом декількох уроків ознайомлюється з основами фреймворку. Як результат, користувач, не витрачає гроші на вивчення матеріалу, який йому може не сподобатись та економить багато часу, оскільки уроки короткі, але місткі, інформація подається в повній мірі, описуються можливі нюанси.

2. Поглиблені відомості, якщо користувач дійшов до цього розділу — це означає, що матеріал корисний, цікавий та легко засвоюється, відповідно далі можна перейти до уроків, які містять більш детальні та глибокі відомості, специфічні моменти та особливості фреймворку.

3. Задачі — це розділ, де знаходяться конкретні задачі, цінність цього розділу полягає в тому, якщо у розробника, на роботі, може виникнути схоже завдання, з такою підказкою можна буде вирішити поставлену задачу значно швидше. Оскільки можна взяти за основу готовий варіант, який показаний у прикладі і залишиться лише додати мінімальні правки, враховуючи специфічність проекту.

Як результат проектування програмного засобу була створена наступна схема додатку (див. рисунок 10).

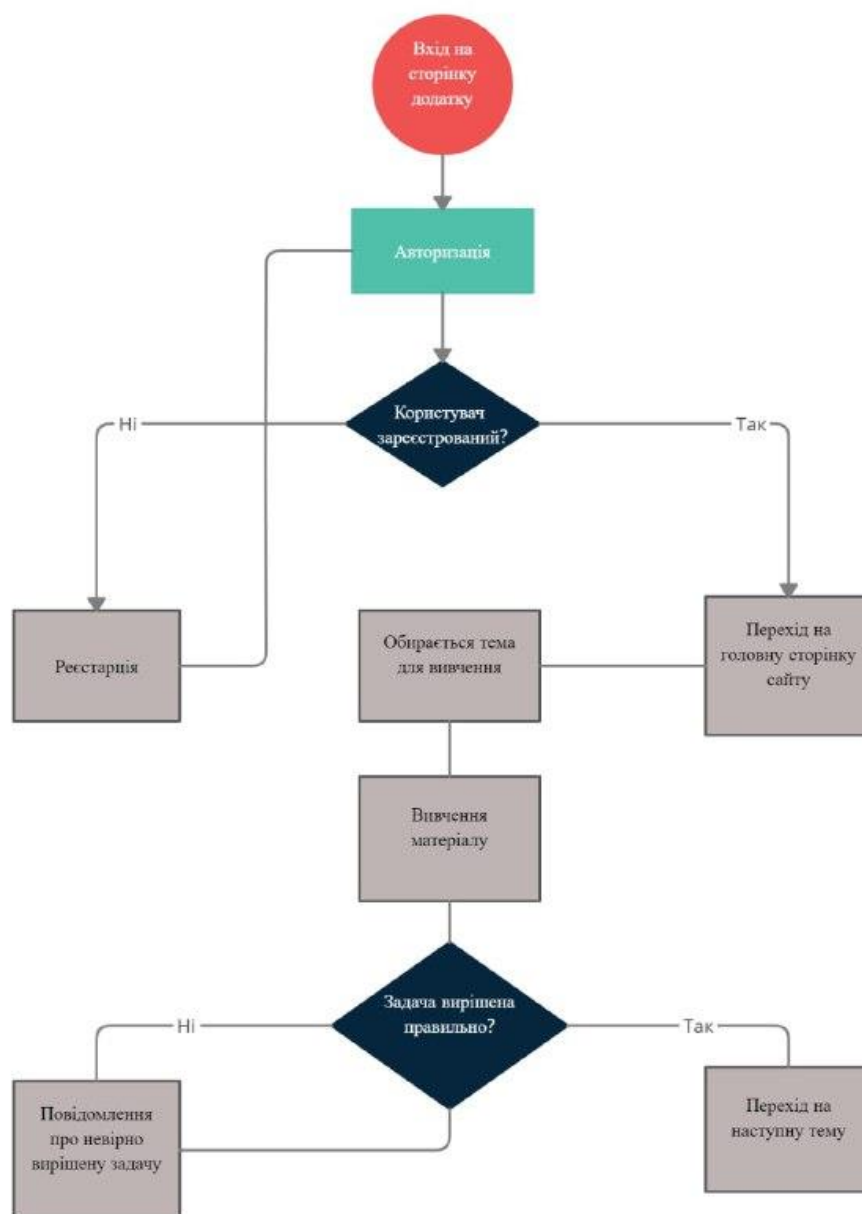


Рисунок 10 – Схема додатку

Архітектура додатку

Основою веб-додатку являється клієнт-серверна взаємодія, в якій роль клієнту займає браузер, який відправляє запити. Сервер, в свою чергу, обробляє отримані запити і, якщо потрібно, робить запит до бази даних, де знаходяться потрібні дані, які повертаються серверу, сервер обробляє отриману інформацію, формує кінцевий результат, наприклад веб-сторінку, та повертає клієнту, схему такої взаємодії зображено на рисунку 11.

Для даного додатку сервер потрібен для реєстрації користувачів. У базі даних повинні зберігатись пошти та паролі користувачів. Під час авторизації потрібно щоб сервер зрівнював дані, які відправив користувач, з поштою та паролем, які знаходяться в базі даних і, якщо такі дані є та пошта відповідає паролю, переводити користувача на головну сторінку, або повідомляти про неправильне введення даних. Також потрібно зберігати відповіді на задачі, які будуть вкінці кожного уроку. Для цього користувач відправляє запит, який містить відповідь до задачі, сервер зрівнює її з відповіддю, яка є на сервері, якщо вони збігаються, студент переходить до вивчення наступної теми, якщо ні, то повертається повідомлення про неправильну відповідь та пропозиція надання нової.

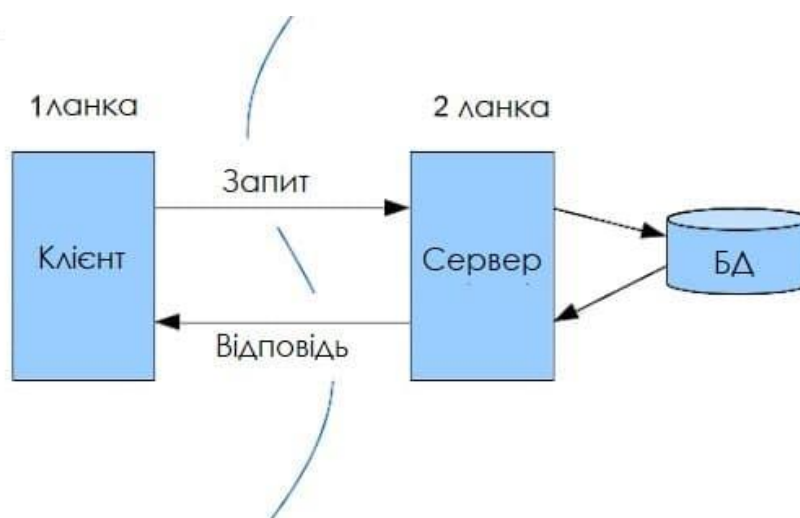


Рисунок 11 – Архітектура додатку

3 РОЗРОБКА ТА ТЕСТУВАННЯ ВЕБ-ДОДАТКУ

3.1 Етапи розробки веб-додатку

1. Створення index.html файлу, який буде головною сторінкою. Ця сторінка буде містити розділи для вивчення, у форматі посилань на сторінки з відповідними темами.

2. Створення сторінок з інформацією для вивчення, наповнення їх контентом(теорією та вставками коду).

3. Стилізація сторінок, робота над UI дизайном, адаптація додатку для планшетів та смартфонів, розробка підтримки популярних браузерів.

4. Створення бази даних та підключення серверу.

5. Створення форми для реєстрації та авторизації користувача.

6. Написання коду для валідації введених даних у формі та створення запитів на сервер і обробки отриманих відповідей.

Веб-додаток — це клієнт-серверний додаток, у якому клієнтом виступає браузер, а сервером — веб-сервер [17]. Основна частина додатку, як правило, знаходиться на стороні веб-серверу, який обробляє отримані запити і формує відповідь, який відправляється користувачу. Далі в роботу включається сам браузер, він перетворює отриману відповідь від серверу в графічний інтерфейс, який уже зрозумілий користувачу.

Веб-сервер — приймає HTTP-запити від клієнта і повертає HTTP-відповіді. Найбільш поширеними типами програмного забезпечення веб-серверів є Apache, IIS і NGINX. Додаток запускається на веб-сервері, який може бути реалізований з використанням різних мов програмування, таких як: PHP, Python, Ruby, Java, Perl.

3.2 Підключення бази даних

Для того щоб додаток міг взаємодіяти з користувачем, потрібно підключити базу даних як показано на рисунку 12.

```

php > bd.php > ...
1  <?php
2  $domen = "localhost";
3  $user = "root";
4  $password = "";
5  $name = "vue_bd";
6
7  $db = new mysqli($domen, $user, $password);
8  $db->set_charset("utf8mb4");
9  $db->select_db($name);
10 ?>

```

Рисунок 12 – Підключення бази даних

3.2.1 Реалізація реєстрації та авторизації користувачів

3.2.1.1 Реєстрація

Спочатку потрібно створити таблицю в базі даних, яка буде зберігати дані користувачів з полями пошти та паролю.

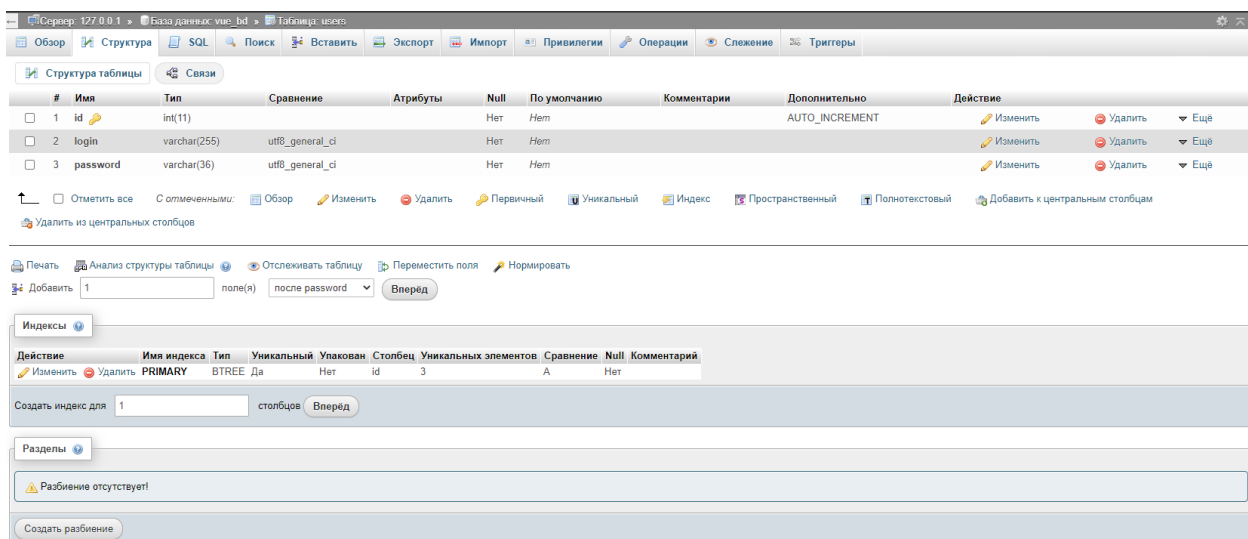


Рисунок 13 – Таблица для реєстрації користувачів

Після цього, на рисунку 14 показано як, зі сторони клієнту, відправляються дані користувача та обробляється отримана відповідь.

```

document.querySelector('#reg_btn').addEventListener('click', (event) => {
    event.preventDefault();
    if(validation_check()) {
        $.ajax({
            type: "POST",
            url: "php/reg.php",
            dataType: "text",
            data: {
                email: document.querySelector('#email').value,
                password: document.querySelector('#password').value
            },
            success: function (data) {
                alert(data)
                if (data == 'Ok') {
                    window.location.href = window.location.origin + '/authorization.php';
                }
            }
        });
    }
});

```

Рисунок 14 – Відправка пошти та паролю введених користувачем

Також потрібно описати запис даних користувача зі сторони бази даних, як показано на рисунку 15.

```

reg.php > ...
<?php
include_once "bd.php";

if (isset($_POST['email']) && isset($_POST['password'])) {
    $pass = md5($_POST['password']);
    if ($db->query("INSERT `users` (`login`, `password`) VALUES ('". $_POST['email'] ."', '". $pass ."'")) {
        echo 'Ok';
        exit(0);
    }
}

```

Рисунок 15 – Додавання даних користувачів у базу даних

3.2.1.2 Авторизація

Після того, як користувач зареєструвався, потрібно пройти авторизацію. Реалізація відправлення даних зі сторони клієнта показано на рисунку 16.

```

document.querySelector('#auth_btn').addEventListener('click', (event) => {
    event.preventDefault();
    if(validation_check()) {
        $.ajax({
            type: "POST",
            url: "php/auth.php",
            dataType: "text",
            data: {
                email: document.querySelector('#email').value,
                password: document.querySelector('#password').value
            },
            success: function (data) {
                if (data == 1) {
                    window.location.href = window.location.origin + '/index.php';
                } else {
                    error.style.display = 'block';
                    error.innerHTML = data;
                }
            }
        });
    }
});

```

Рисунок 16 – Відправлення даних для авторизації.

У базі даних перевіряється наявність введеної пошти та відповідного паролю і повертається відповідь, в залежності від якої, користувач потрапляє на головну сторінку або отримує повідомлення про те, що дані були введені невірно, код для обробки такого запиту показаний на рисунку 17.

```

<?php
include_once "bd.php";

if (isset($_POST['email']) && isset($_POST['password'])) {

    $pass = md5($_POST['password']);

    $check_user = $db->query("SELECT `id` FROM `users` WHERE `login` = '" . $_POST['email'] . "' && `password` = '" . $pass . "'");
    if ($check_user->num_rows > 0) {
        echo 1;
    } else {
        echo 'Невірний логін або пароль';
    }
} else {
    echo 'Відсутній логін або пароль';
}

```

Рисунок 17 – Обробка запиту під час авторизації

3.3 Перевірка завдань

В кінці кожного уроку, як додається матеріал показано у додатку Б, знаходиться задача, виконання якої дозволить перейти до наступного уроку. Для того щоб перевіряти відповідь, потрібно створити таблицю, яка буде містити правильні відповіді та написати відповідний код для відправки запитів як показано на рисунку 18. Також описати обробку запиту та повернення відповіді як показано на рисунку 19. Після успішного вирішення задачі, користувач переходить до наступного уроку.

```
document.querySelector('#checkBtn').addEventListener('click', () => {  
  $.ajax({  
    type: "POST",  
    url: "../php/check_answer.php",  
    dataType: "text",  
    data: {  
      val: document.querySelector('answer').value,  
      bnt_id: document.querySelector('answer').getAttribute('bd-id')  
    },  
    success: function (data) {  
      if (data == 1) {  
        window.location.href = window.location.origin + '/pages/loop-and-conditions.php';  
        localStorage.themes_completed = 1;  
      } else {  
        error.style.display = 'block';  
        error.innerHTML = data;  
      }  
    }  
  });  
});
```

Рисунок 18 – Відправлення відповіді користувача

```
check_answer.php > ...  
<?php  
include_once "bd.php";  
  
if (isset($_POST['val']) && isset($_POST['bnt_id'])) {  
  $right_answer = $db->query("SELECT `answer` FROM `answers` WHERE `id` = '". $_POST['bnt_id'] ."'");  
  if ($right_answer->num_rows > 0) {  
    $right_answer = $right_answer->fetch_array(MYSQLI_ASSOC);  
  
    if ($_POST['val'] == $right_answer['answer']){  
      echo 1;  
    } else {  
      echo 'Відповідь не правильна, спробуйте ще раз';  
    }  
  } else {  
    echo 0;  
  }  
}
```

Рисунок 19 – Повернення результату вирішення задачі

3.4 Розміщення web-додатку

Щоб опублікувати додаток потрібен сервер. Для цього потрібно обрати виділений сервер.

Виділений сервер — один з видів хостингу, клієнт отримує повністю окрему фізичну машину, на протигагу віртуальному хостингу. Зазвичай використовується для запуску додатків, які не можуть існувати на одному сервері з іншими проектами або ж мають підвищену потребу до ресурсів [21].

Далі за допомогою конфігуратора потрібно обрати сервер, процесор, кількість оперативної пам'яті та диск. На рисунку 10 показано які конфігурації буде обрано.

The screenshot shows the 'Конфигуратор выделенных серверов' (Dedicated Server Configurator) interface. It features several columns for configuration options:

- Сервер (Server):** Intel Xeon E3, Intel Xeon, Intel Core (selected), AMD EPYC.
- Процессор (Processor):** i9-9900K 8C/16T 3.60GHz, i3-4160 2C/4T 3.60GHz, i3-6100 4C/4T 3.70GHz (selected), i3-7100 2C/4T 3.90GHz, i7-3770T 4C/8T 2.50GHz, i7-4790 4C/8T 3.60GHz, i7-2600 4C/8T 3.40GHz, i7-6700 4C/8T 3.40GHz.
- Память (Memory):** 16 Gb DDR4 (selected), 32 Gb DDR4.
- Диск (Disk):** 256Gb SSD, 480Gb SSD, 512Gb SSD, 960Gb SSD, 1Tb SSD, 2Tb SSD, 1Tb HDD, 3Tb HDD.

On the right, a 'Конфигурация' (Configuration) summary panel displays the selected options:

- Сервер: Intel Core
- Процессор: i3-6100 4C/4T 3.70GHz
- Память: 16 Gb DDR4
- Диск: 1 x 1Tb HDD
- Price: **1610.00 грн / 1 мес.**
- Операционная система: Ubuntu 18.04 LTS

A 'Заказать' (Order) button is visible at the bottom of the summary panel. Below the configurator, there is a note: 'Количество выделенных серверов - ограничено, получает сервер тот, кто первым его оплатит.' (Number of dedicated servers is limited, the server is received by the first one to pay for it.)

Выделенный сервер или dedicated server — вид хостинга, при котором клиенту целиком предоставляется отдельная физическая машина (в противоположность виртуальному хостингу). Обычно используется для запуска приложений, которые не могут сосуществовать на одном сервере с другими проектами или имеют повышенные требования к ресурсам.

Преимущества аренды выделенных серверов

- Прямое включение в точки обмена украинским трафиком (UA-IX, Giganet, DATA-IX, DTEL-IX), прямое подключение клиентов Укртелекома (UKRTEL-IX), подключение к сети обмена трафиком с Россией (RETN, DATA-IX). Суммарная пропускная способность — до 100 Гбит/с.

Рисунок 20 – Конфігурація серверу [21]

Було обрано процесор Intel Core i3-6100 4C/4T 3.70GHz, оперативна пам'ять: 16Gb DDR4, та диск на 1Tb HDD.

Далі було обрано версію операційної системи: Ubuntu 18.04 LTS. Перевагою цієї операційної системи являється підтримка протягом п'яти років з моменту релізу.

Тепер потрібно встановити програмне забезпечення на сервер, для цього використовується пакетний менеджер aptget — це менеджер пакетів який входить до складу ubuntu, для встановлення пакетів потрібно користуватись командою: `sudo apt install <потрібний пакет>`. Встановлюємо наступні пакети: PHP, PhpMyAdmin, Apache, Nginx, MySQL.

Останній крок перед публікацією, потрібно придбати домен, налаштувати його, та підключити SSL-сертифікат.

3.5 Демонстрація розробленого додатку

З початку користувач потрапляє на сторінку авторизації, де вводить пошту та пароль.

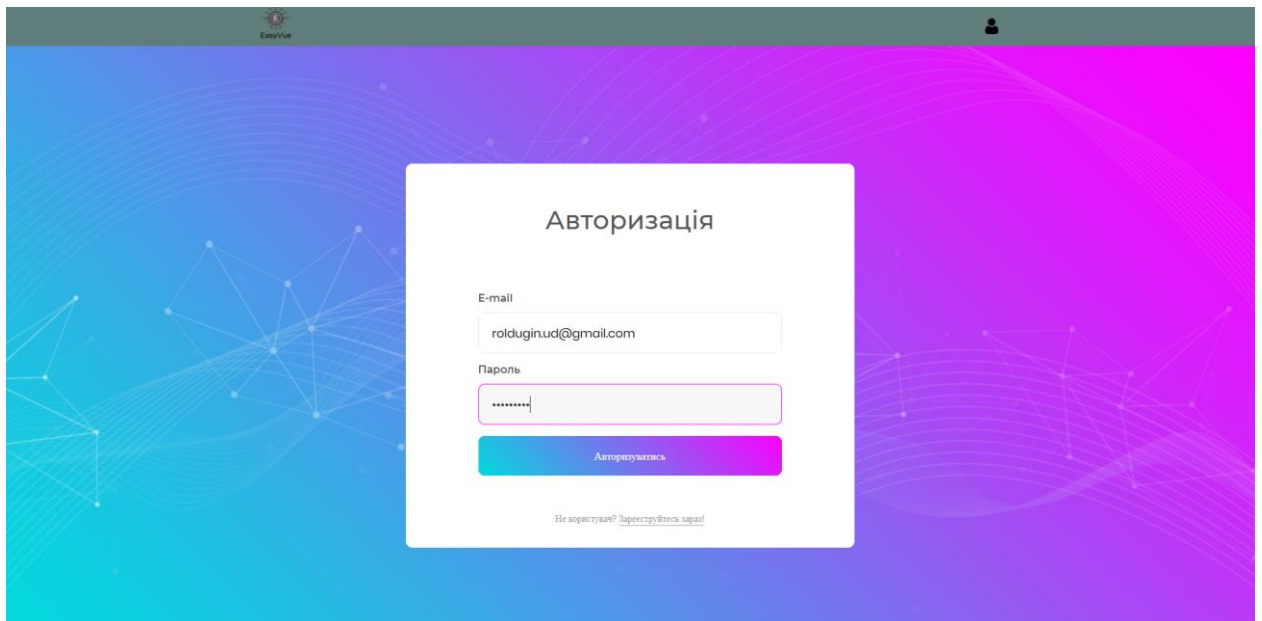


Рисунок 21 – Авторизація користувача

Якщо ж це новий користувач, натиснувши на посилання внизу форми авторизації, відкриється форма для реєстрації.

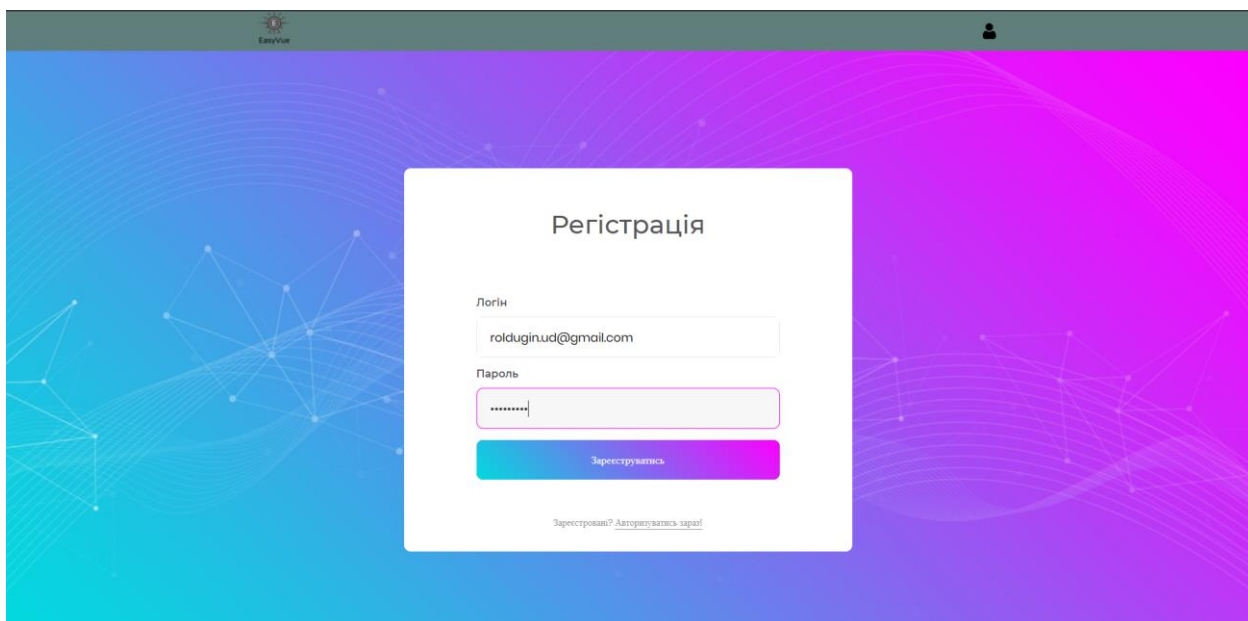


Рисунок 22 – Реєстрація користувача

Після реєстрації та авторизації, користувач потрапляє на головну сторінку, де йому поки доступний лише один, перший урок.

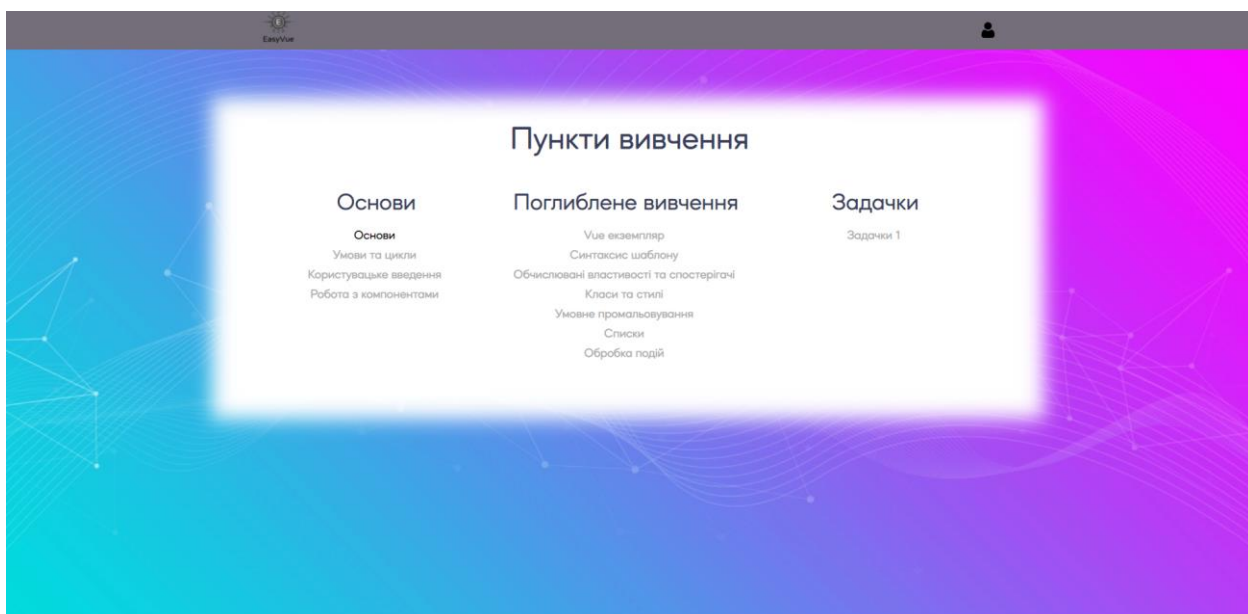


Рисунок 23 – Головна сторінка додатку

Користувач переходить на доступний урок, ознайомлюється з матеріалом, аналізує вставки коду та пояснення.

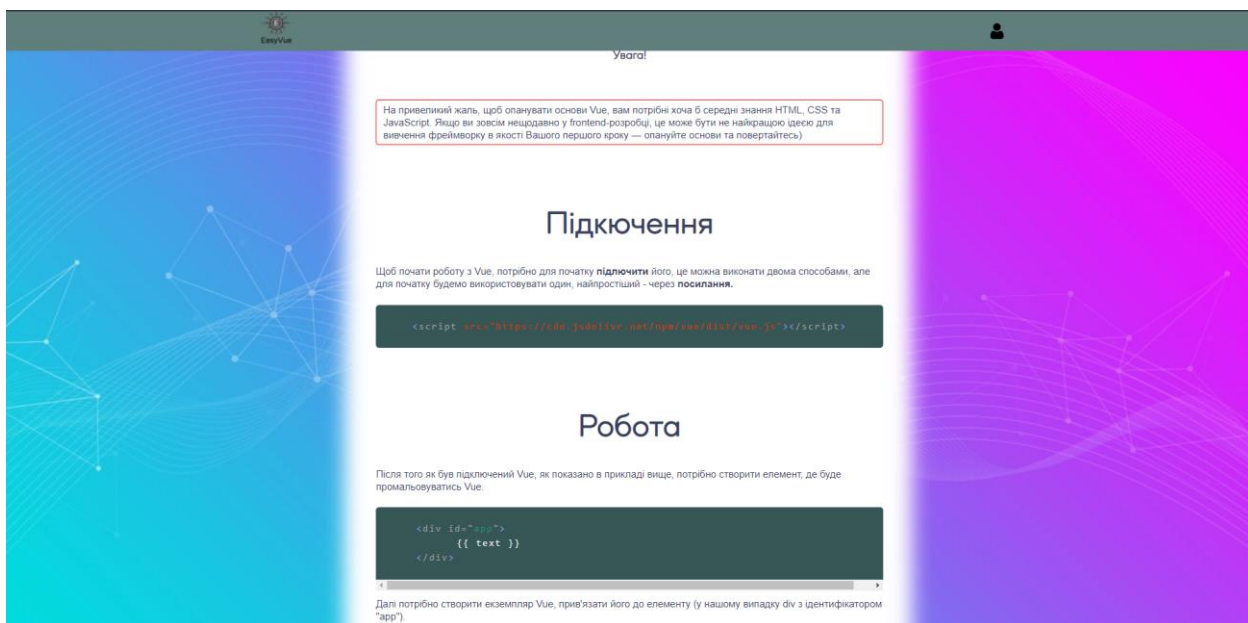


Рисунок 24 – Сторінка додатку з матеріалом для вивчення

Щоб перейти до наступного розділу потрібно вирішити задачу, такі завдання зустрічаються вкінці кожного уроку, успішне вирішення дає гарантію, що користувач засвоїв матеріал і може переходити до наступного уроку. Це необхідно для того, щоб знання отримувались в правильному порядку та залишались у пам'яті, оскільки для вивчення наступних тем, в основному, потрібні знання з попередніх.

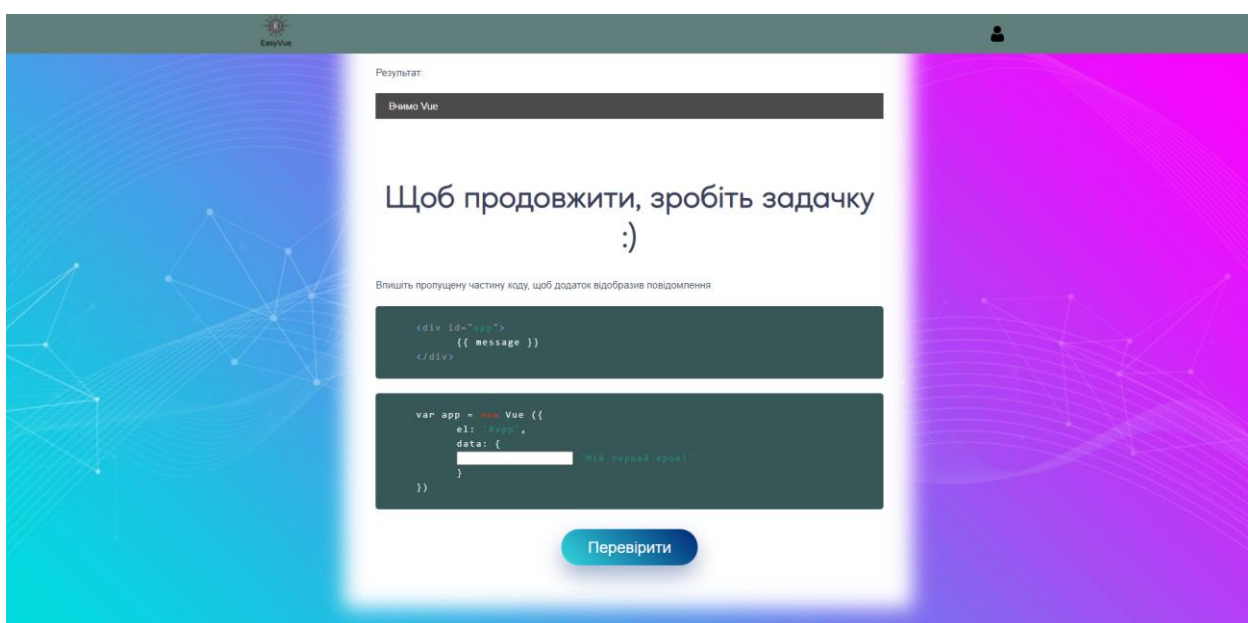


Рисунок 25 – Завдання для перевірки отриманих знань

3.6 Тестування розробленого додатку

У наш час у постійному мінливому і конкурентному світі, інтернет став невід'ємною частиною життя. Більшість людей приймають рішення, переглядаючи інформацію в Інтернеті, тому хостинг веб-додатку є не стільки обов'язковою, а скільки необхідною опцією для всіх використання таких програмних продуктів.

Просто мати додаток недостатньо. Організації потрібно розробити інформативний, доступний і зручний веб-продукт. Для збереження всіх цих якостей, веб-додаток повинен бути протестований, і цей процес називається веб-тестуванням або тестуванням додатку.

Веб-тестування — це ретельна перевірка сайту або додатку на наявність потенційних помилок [9]. Це повне тестування веб-додатків перед запуском. Веб-система повинна бути повністю перевірена від початку до кінця, перш ніж вона буде запущена для кінцевих користувачів. Також, в ході тестування визначається зручність і привабливість додатку для користувача, швидкість отримання та доступність потрібної інформації, надійність і безпека використання. Веб-тестування також можна використовувати для виявлення певних областей або аспектів веб-додатку, які можна поліпшити і отримати кращі результати, наприклад, більше постійних відвідувачів.

1) Тестування верстки:

Візуальна цілісність веб-додатку при масштабуванні (75-150%):

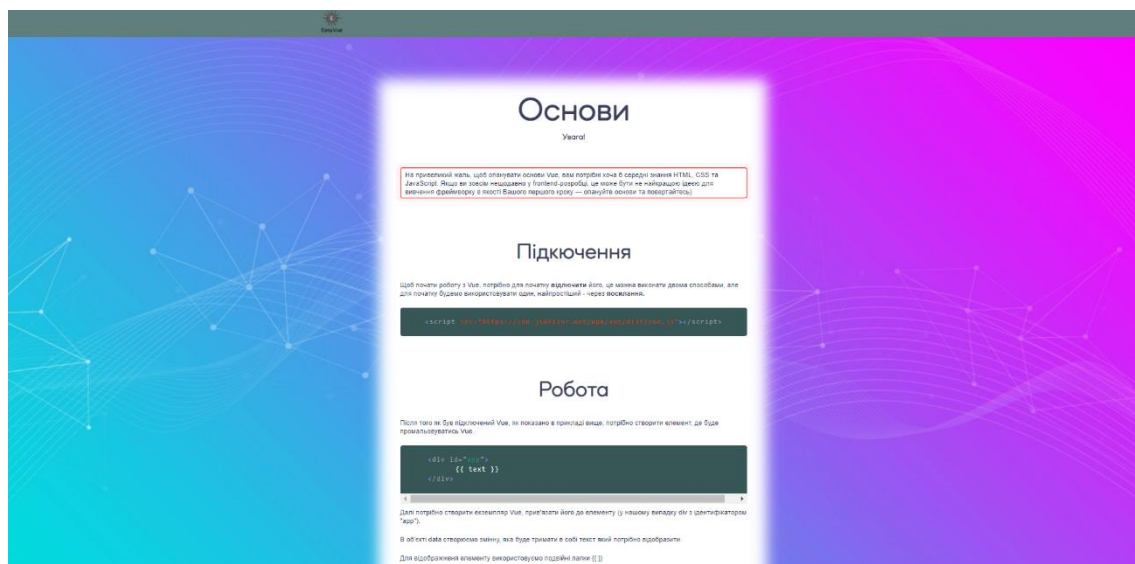


Рисунок 26 – Масштаб сторінки 75%

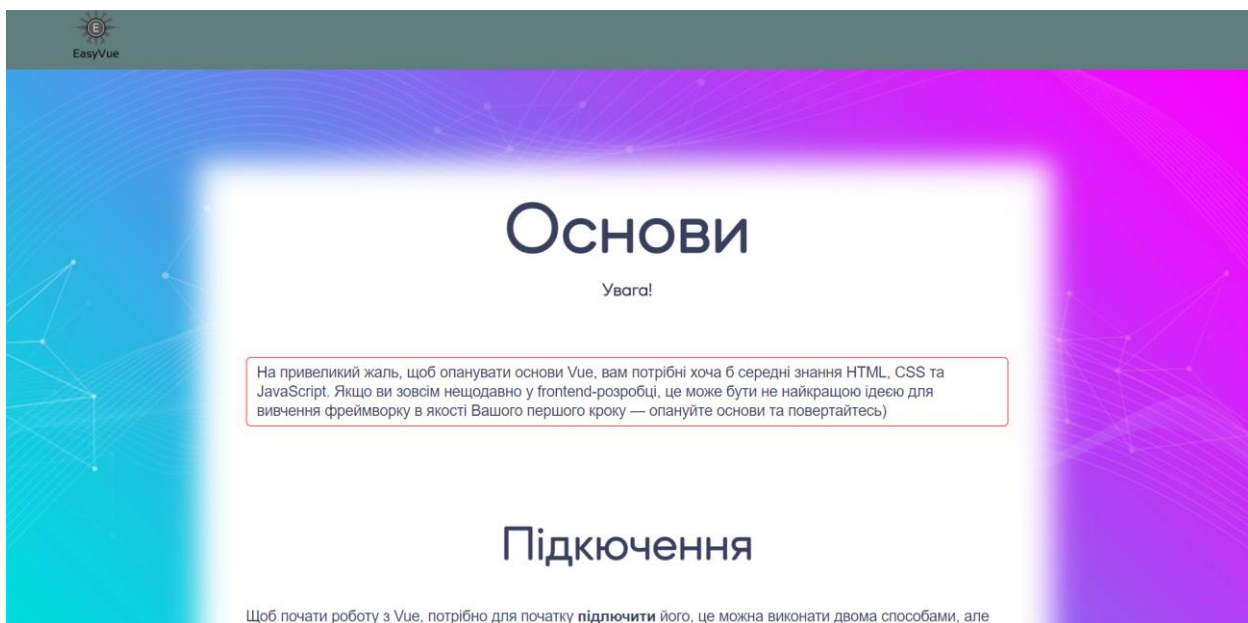


Рисунок 27 – Масштаб сторінки 150%

2) Кросбраузерність:

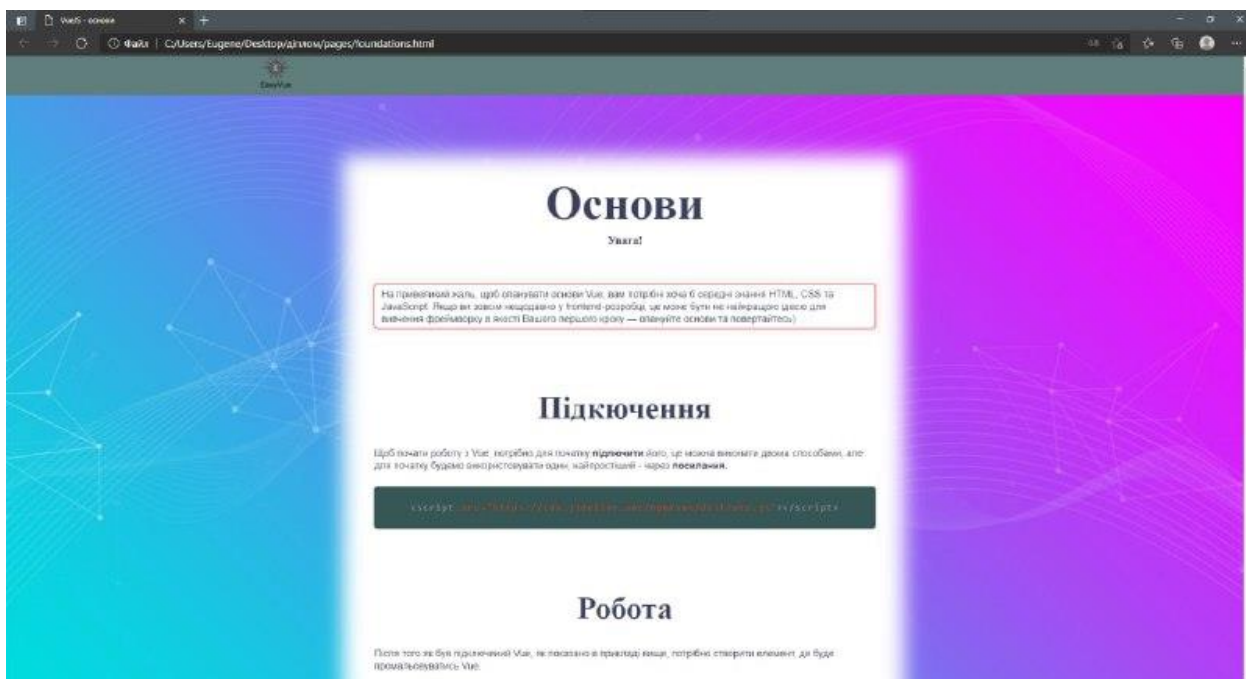


Рисунок 28 – Вигляд веб-додатку у Microsoft Edge

3) Адаптивність. Цей пункт обумовлений потребою користувачів повноцінно користуватись веб-додатком незалежно від пристрою.

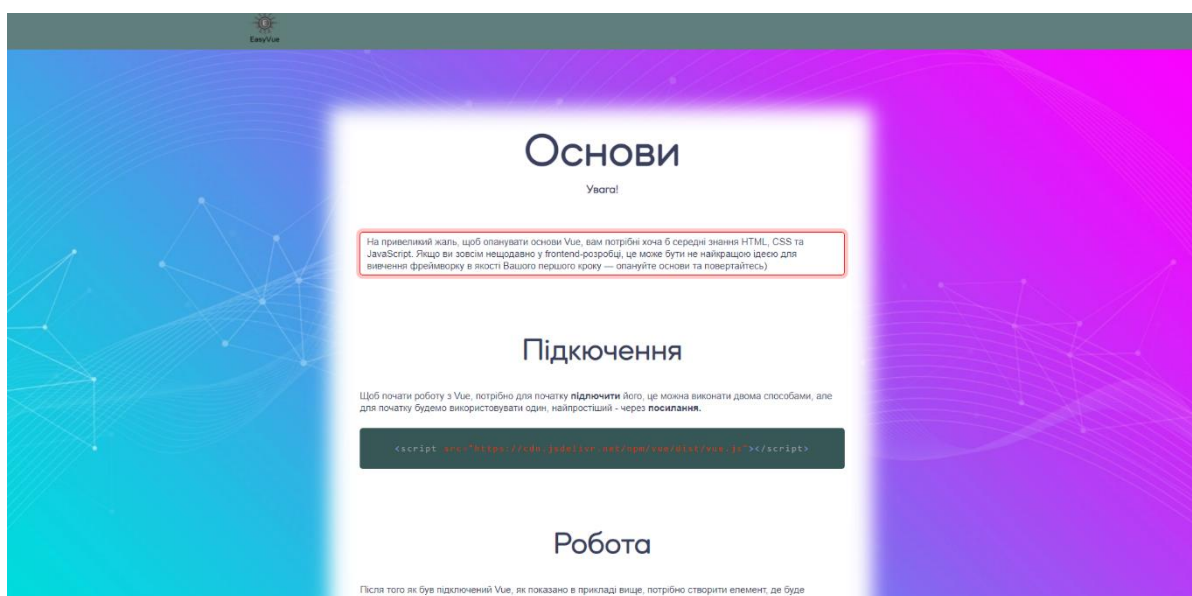


Рисунок 29 – Вигляд веб-додатку на ПК

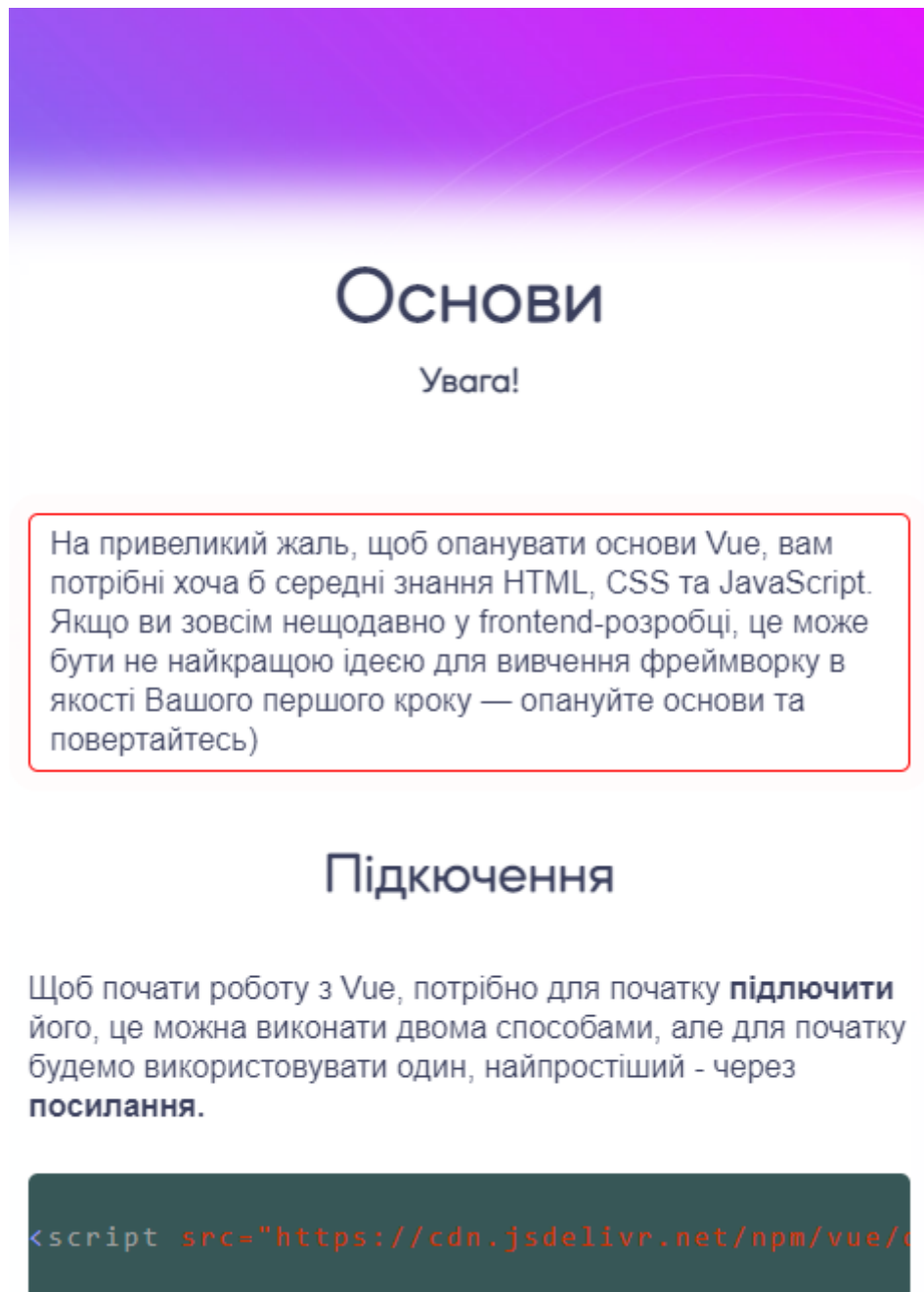


Рисунок 30 – Вигляд веб-додатку на мобільних пристроях

- 4) Функціональне тестування:
 - а. Тестування форми авторизації;

ВИСНОВКИ

Дана кваліфікаційна робота присвячена розробці веб-додатку для вивчення фреймворку JavaScript Vue.js.

В ході виконання дипломної роботи було отримано такі результати:

1. Визначено ефективність використання веб-додатків для вивчення фреймворку VueJs;
2. Представлено приклади додатків-аналогів;
3. Описано алгоритм створення веб-додатку для вивчення фреймворку JS VueJs;
4. Проведено тестування створеного веб-додатку.

На основі отриманих результатів можна зробити такі висновки:

1. Використання веб-додатків для вивчення фреймворку VueJs є ефективним, так як більшість успішних програмістів, для того щоб вивчати програмування з нуля, а саме фреймворк VueJs, використовували саме вищевказані додатки-аналоги, які містять різноманітні завдання та курси.

2. Проведений аналіз переваг та недоліків аналогів дає підстави стверджувати, що основними мінусами таких програмних засобів є: англomовний інтерфейс, що може викликати труднощі у людей, які не мають мінімальних навичок з англійської мови та платні додаткові інструменти для поглибленого вивчення відповідного матеріалу. Тому на основі виявлених недоліків було створено веб-додаток для вивчення js фреймворку vueJs, який компенсує вищеперераховані недоліки.

3. Опираючись на вимоги, був складений план розробки, який містить у собі потрібний функціонал, та особливості, які б вирізняли його на фоні додатків-аналогів.

4. Тестування розробленого веб-додатку дало підстави стверджувати, що створений програмний засіб можна використовувати для вивчення js фреймворку Vuejs

Web-додаток можна використовувати для вивчення Vue.js, так як цей додаток містить теоретичні відомості, практичні завдання та задачі, які полегшують і пришвидшують вивчення фреймворку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Best Practice Software Engineering – Model View Controller [Електронний ресурс] – Режим доступу: <http://best-practice-software-engineering.ifs.tuwien.ac.at/patterns/mvc.html>.
2. SoloLearn: Learn to Code for Free! [Електронний ресурс] – Режим доступу: <https://www.sololearn.com/>.
3. The Pros and Cons of Vue.js | AltexSoft [Електронний ресурс] – Режим доступу: <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>.
4. What is JavaScript? - Learn web development | MDN [Електронний ресурс] – Режим доступу: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript.
5. Аналіз фреймворків як засобів розробки Web-додатків [Електронний ресурс] – 2016 – Режим доступу: http://cad.kpi.ua/attachments/093_2016d_Protsenko.pdf.
6. Веб-програмування | Кафедра АПЕПС ТЕФ КПІ — програмна інженерія [Електронний ресурс] – Режим доступу: <http://apeps.kpi.ua/web-programuvania>.
7. Веб-студія Centum-D: розробка і створення сайтів [Електронний ресурс] – Режим доступу: <https://www.centum-d.com/uk/veb-dodatok-yogo-harakteristiki>.
8. Дерево програмування: Концепція прив'язки даних [Angular] [Електронний ресурс] – Режим доступу: <http://theprogrammingtree.blogspot.com/2015/07/data-binding-concept-angular.html>.
9. Етапи і види тестування web-додатків [Електронний ресурс] – Режим доступу: <https://polyarix.com/blog/testirovanie-sajta/>.
10. Кодування для Android - Upskew Pty. Ltd [Електронний ресурс] – Режим доступу: <https://upskew.com/encode-android/>.
11. Кодування для дітей | Мод Minecraft | Тинкер [Електронний ресурс] – Режим доступу: <https://www.tynker.com/parents/>.
12. Методи розробки Web-сайтів [Електронний ресурс] – Режим доступу: <https://sites.google.com/site/tz5103voinovakateryna/metodi-rozrobki-web-sajtiv>.

- 13.Мімо: Навчіться кодувати [Електронний ресурс] – Режим доступу: <https://getmimo.com/>.
- 14.МОВА СЦЕНАРІЇВ JAVASCRIPT, Основи поняття - Проектування і розробка web-додатків [Електронний ресурс] – Режим доступу: https://stud.com.ua/97595/informatika/mova_stsenariyiv_javascript.
- 15.Навчіться безкоштовно кодувати – Коник [Електронний ресурс] – Режим доступу: <https://grasshopper.app/>.
- 16.Основні тренди у веб-дизайні [Електронний ресурс] – 2019 – Режим доступу: <https://fel202.files.wordpress.com/2019/09/d092d0b5d0b1-d182d0b5d185d0bdd0bed0bbd0bed0b3d196d197.pdf>.
- 17.Особливості тестування веб-додатків — Лабораторія якості [Електронний ресурс] – Режим доступу: <https://quality-lab.ru/blog/key-principles-of-web-testing/>.
- 18.Розробка шаблону сайту [Електронний ресурс] – 2020 –Режим доступу: <http://elartu.tntu.edu.ua/bitstream/lib/33611/1/%D0%9C%D0%B0%D0%BB%D1%8F%D1%80%D1%81%D1%8C%D0%BA%D0%B8%D0%B9.pdf>.
- 19.Фреймворк [Електронний ресурс] – Режим доступу: <https://znaimo.com.ua/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA>.
- 20.Що таке JavaScript? Посібник для початківців – Skillcrush [Електронний ресурс] – Режим доступу: <https://skillcrush.com/blog/javascript/>.
- 21.Хостинг Україна — купити хостинг в Україні [Електронний ресурс] – Режим доступу: <https://www.ukraine.com.ua/>.

ДОДАТКИ

Додаток А

Валідація форми для реєстрації користувача

```
function validation_check(){  
    var input = $('validate-input .input100');  
    var check = true;  
  
    for(var i=0; i<input.length; i++) {  
        if(validate(input[i]) == false){  
            showValidate(input[i]);  
            check=false;  
        }  
    }  
  
    return check;  
}
```

```
function validate (input) {  
    if($(input).attr('type') == 'email' ||  
    $(input).attr('name') == 'email') {  
        if($(input).val().trim().
```

```

        match(/^[a-zA-Z0-9_-\.\.+]@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|((([a-zA-Z0-9\-\.]+)\.)))([a-zA-Z]{1,5}|[0-9]{1,3})(\)?\$/)) == null) {

            return false;

        }

    }

    else if($(input).val().trim() == ''){

        return false

    } else {

        return true;

    }

}

function showValidate(input) {

    var thisAlert = $(input).parent();

    $(thisAlert).addClass('alert-validate');

}

function hideValidate(input) {

    var thisAlert = $(input).parent();

    $(thisAlert).removeClass('alert-validate');

}

```


Додаток Б

Розмітка матеріалу уроку з розділу «Основи»

```

<main      class="main-content      js-main-content"
role="main">
    <div class="max-width clearfix">

        <section class="max-width-col-8" >
            <h1 class="article-title" id="gerund">
                Основи      <span      class="title-
comment">Увага!</span>
            </h1>

            <div class="content">
                <article class="article-content ll-text">
                    <p class="alarm">На привеликий жаль, щоб
опанувати основи Vue, вам потрібні хоча б середні знання
HTML, CSS та JavaScript. Якщо ви зовсім нещодавно у
frontend-розробці, це може бути не найкращою ідеєю для
вивчення фреймворку в якості Вашого першого кроку – опануйте
основи та повертайтеся) </p>
                </article>
                <div
                    class="paragraphs-items      paragraphs-
items-field-paragraph      paragraphs-items-field-paragraph-
paragraphs-editor-preview      paragraphs-items-paragraphs-
editor-preview">
                    <div      class="field      field-name-field-
paragraph field-type-paragraphs field-label-hidden">

```

```

<div class="field-items">
  <div class="field-item even">
    <h2 class="caption-title"
id="formi-gerundiia" about="" typeof="">
      Підключення</h2>
    </div>
    <div class="field-item odd">
      <article class="paragraph"
about="" typeof="">
        <div class="paragraph-content
ll-text">
          <p>Щоб почати роботу з Vue,
потрібно для початку <strong>підключити</strong> його, це
можна виконати двома способами, але для початку будемо
використовувати один, найпростіший - через
<strong>посилання.</strong></p>
          <code class="center"
style="padding-bottom: 1.5rem;">
            <span class="tag">script <span
class="atr">src="https://cdn.jsdelivr.net/npm/vue/dist/vue
.js"</span></span></span><span
class="tag">/script</span></code>
          </div>
        </article>
      </div>
    <div class="field-item even">
      <h2 class="caption-title"
id="formi-gerundiia" about="" typeof="">Робота</h2>

```

```

<article class="paragraph"
about="" typeof="">
    <div class="paragraph-content
ll-text">
        <p>Після того як був
підключений Vue, як показано в прикладі вище, потрібно
створити елемент, де буде промальовуватись Vue.</p>
        <code>
    <span class="tag">div id="<span
class="vueEl">app</span>"</span>
    <span class="textInCode">{{ text }}</span>
    <span class="tag">/div</span>
        </code>
    </div>
    <div class="paragraph-content
ll-text mt10">
        <p>Далі потрібно створити
екземпляр Vue, прив'язати його до елемента (у нашому випадку
div з ідентифікатором "app").</p>
        <p>В об'єкті data створюємо
змінну, яка буде тримати в собі текст який потрібно
відобразити.</p>
        <p>Для відображення елемента
використовуємо подвійні лапки {{ }}</p>
        <code>
    <span class="art">var</span> app = <span
class="atr">new</span> Vue ({
    el: <span class="vueEl">'#app'</span>,
    data: {

```




ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА WEB-ДОДАТКУ ДЛЯ ВИВЧЕННЯ ФРЕЙМВОРКУ JAVASCRIPT VUE.JS

Виконав студент 4 курсу

Групи ПД-44

Ролдугін Євгеній Віталійович

Керівник роботи

К.т.н. Яскевич Владислав Олександрович

Київ – 2021

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи:** розвинення здібностей у розробці веб-додатків.
- **Об'єкт дослідження:** розробка веб-додатків.
- **Предмет дослідження:** веб-додаток для вивчення JS фреймворку VueJs.

АНАЛОГИ

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПЕРЕВАГ ТА НЕДОЛІКІВ

Назва	Переваги	Недоліки
SoloLearn	<ol style="list-style-type: none"> 1. Великий функціонал безкоштовної версії. 2. Простий у використанні. 3. Навчання можна проходити будь-де і з будь-якого гаджету без підключення до мережі Інтернет. 	<ol style="list-style-type: none"> 1. Висока вартість оформлення Pro версії. 2. Тільки англomовний інтерфейс.
Mimo	<ol style="list-style-type: none"> 1. Програмування з нуля. 2. Створення додатків. 3. Створення ігор. 4. Створення веб-сайтів. 	<ol style="list-style-type: none"> 1. Відсутність україномовного інтерфейсу. 2. <u>Безкоштовна версія лише 7 днів.</u>

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПЕРЕВАГ ТА НЕДОЛІКІВ

Назва	Переваги	Недоліки
<u>Encode</u>	1. Простий інтерфейс. 2. Велика кількість різноманітних цікавих завдань.	1. Англomовний інтерфейс. 2. Велика вартість <u>pro</u> -версії.

Порівняльний аналіз виявив, що основними недоліками майже усіх веб-додатків є англomовний інтерфейс, що може викликати труднощі у людей, які не мають мінімальних навичок з англійської мови та платні додаткові інструменти для поглибленого вивчення відповідного матеріалу. Тому на основі виявлених недоліків було створено веб-додаток для вивчення фреймворку JavaScript Vue.js, який компенсує вищепераховані недоліки.

5

ТЕХНІЧНІ ЗАВДАННЯ

1. Проаналізувати додатки-аналоги.
2. Обрати програмні засоби реалізації та технології для розробки.
3. Розробити структуру уроків та завдань для перевірки вивченого матеріалу.
4. Розробити додаток.
5. Протестувати створений веб-додаток.

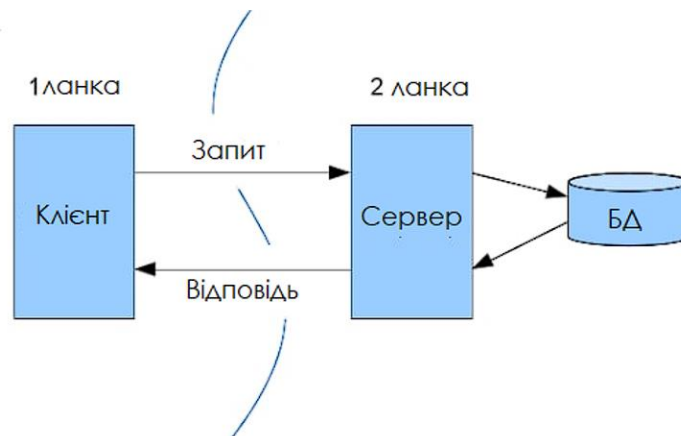
6

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



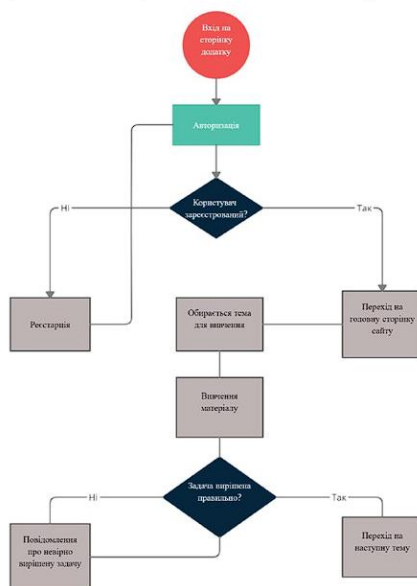
7

Схема роботи веб-додатку



8

Алгоритм роботи користувача



9

Теми для вивчення

Пункти вивчення

Основи

Основи
Умови та цикли
Користувацьке введення
Робота з компонентами

Поглиблене вивчення

Усе екземляр
Синтаксис шаблону
Обчислювані властивості та спостережачі
Класи та стилі
Умовне промальовування
Списки
Обробка подій

Задачки

Задачки 1

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Ролдугін Є.В. // Застосування програмного забезпечення в ІКТ: Матеріали першої всеукраїнської науково-технічної конференції. Збірник тез. 12.02.2021, ДУТ, м. Київ — К.: ДУТ, 2021. — С. 83.

Ролдугін Є.В. // Сучасні інтелектуальні інформаційні технології в науці та освіті /

11

ВИСНОВКИ

У ході виконання дипломної роботи були виконані поставлені завдання:

- Проаналізовано додатки-аналоги.
- Обрано програмні засоби реалізації та технології для розробки.
- Розроблено структуру уроків та завдань.
- Розроблено веб-додаток.
- Проведено тестування розробленого додатку.

Перспективи подальших досліджень

- Збільшити кількість тем для вивчення.
- Додати інтерактивних завдань для перевірки отриманих знань.

12

ДЯКУЮ ЗА УВАГУ!

