

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка Web-сайту для вивчення італійської мови
з використанням HTML, CSS, JS»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

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

(підпис)

Анастасія ОСАДЧА

Виконала: здобувачка вищої освіти групи ППЗ-51

Анастасія ОСАДЧА

Керівник: _____ Віктор ГРЕБЕНІЮК

доктор філософії
(PhD)

Рецензент: _____

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

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Осадчій Анастасії Андріївні

1. Тема кваліфікаційної роботи: «Розробка Web-сайту для вивчення італійської мови з використанням HTML, CSS, JS»

Керівник кваліфікаційної роботи доктор філософії (PhD), доцент кафедри ІІЗ Віктор ГРЕБЕНЮК,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. №36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: офіційна документація HTML, офіційна документація CSS, офіційна документація JavaScript, документація для роботи у середовищі розробки Visual Studio Code.

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

1. Дослідження особливостей процесу вивчення італійської мови.
2. Визначення вимог до вебсайту для вивчення італійської мови.
3. Проектування вебсайту для вивчення італійської мови.
4. Розробка вебсайту для вивчення італійської мови.
5. Тестування розробленого вебсайту.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до вебсайту.
3. Програмні засоби реалізації.
4. Діаграма варіантів використання.
5. Схема структури вебсайту.
6. Діаграма пакетів скриптів вебсайту.
7. Екранні форми.
8. Апробація результатів дослідження.

6. Дата видачі завдання «28» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	28.02-06.03.2024	
2	Аналіз та дослідження існуючих аналогів	07.03-13.03.2024	
3	Дослідження особливостей процесу вивчення італійської мови	14.03-20.03.2024	
4	Визначення вимог до вебсайту для вивчення італійської мови	21.03-25.03.2024	
5	Проектування вебсайту для вивчення італійської мови	26.03-03.04.2024	
6	Розробка вебсайту для вивчення італійської мови	04.04-18.04.2024	
7	Тестування розробленого вебсайту	19.04-28.04.2024	
8	Оформлення роботи: вступ, висновки, реферат	29.04-05.05.2024	
9	Розробка демонстраційних матеріалів	06.05-12.05.2024	
10	Попередній захист роботи	13.05-31.05.2024	

Здобувачка вищої освіти

(підпис)

Анастасія ОСАДЧА

Керівник
кваліфікаційної роботи

(підпис)

Віктор ГРЕБЕНЮК

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 44 стор., 36 рис., 6 табл., 18 джерел.

Мета роботи – спрощення процесу вивчення італійської мови за рахунок використання вебсайту, розробленого за допомогою HTML, CSS, JS.

Об'єкт дослідження – процес вивчення італійської мови.

Предмет дослідження – програмне забезпечення для вивчення італійської мови.

Короткий зміст роботи: В роботі досліджено особливості процесу вивчення італійської мови. Проаналізовано існуючі рішення для вивчення італійської мови: Duolingo, Mondly, Drops, Busuu, Memrise, uTalk та Wlingua, а також визначено переваги та недоліки цих рішень. Визначено функціональні та нефункціональні вимоги до вебсайту для вивчення італійської мови з урахуванням недоліків існуючих рішень. Спроектовано діаграму варіантів використання вебсайту та обрано деревоподібну структуру вебсайту. Програмно реалізовані ключові функції вебсайту: автоматична перевірка тестів в уроках, озвучення слів та фраз при натисканні іконки вуха, функція пошуку перекладу слів у словнику, а також розроблено адаптивну версію вебсайту. В роботі використано мову розмітки HTML, мову таблиць стилів CSS, мову програмування JavaScript та середовище розробки Visual Studio Code для розробки вебсайту. Для проектування графічного інтерфейсу вебсайту було використано графічний редактор Figma. Проведено тестування вебсайту для вивчення італійської мови.

Сферою використання вебсайту є організація процесу самостійного вивчення італійської мови.

КЛЮЧОВІ СЛОВА: ІТАЛІЙСЬКА МОВА, НАВЧАННЯ, ВЕБСАЙТ, HTML, CSS, JAVASCRIPT.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Особливості процесу вивчення італійської мови.....	9
1.2 Аналіз існуючих рішень для вивчення італійської мови.....	10
1.3 Огляд засобів реалізації вебсайту для вивчення італійської мови.....	23
1.4 Визначення вимог до вебсайту для вивчення італійської мови.....	31
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБСАЙТУ.....	32
2.1 Аналіз варіантів використання вебсайту.....	32
2.2 Проектування структури вебсайту.....	34
2.3 Проектування графічного інтерфейсу вебсайту.....	35
2.4 Розробка вебсайту.....	39
2.5 Розробка адаптивної версії вебсайту.....	42
РОЗДІЛ 3 ТЕСТУВАННЯ ВЕБСАЙТУ.....	45
3.1 Тестування інтерфейсу користувача вебсайту.....	46
3.2 Тестування зручності використання вебсайту.....	48
3.3 Функціональне тестування вебсайту.....	49
ВИСНОВКИ.....	51
ПЕРЕЛІК ПОСИЛАНЬ.....	52
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	54
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ.....	63

ВСТУП

Нині знання кількох мов відкриває для людини багато нових можливостей: відсутність мовних бар'єрів при спілкуванні з іноземцями, доступ до більшого обсягу інформації та швидку адаптацію до нового місця перебування. Знання італійської мови дозволяє краще розуміти італійську культуру та полегшує процес вивчення інших романських мов. Для співпраці з італійськими бізнес-партнерами також буде корисно знати італійську мову.

Метою даного дослідження є спрощення процесу вивчення італійської мови за рахунок використання вебсайту, розробленого за допомогою HTML, CSS, JS.

В процесі дослідження вирішувалися наступні завдання:

- дослідження особливостей процесу вивчення італійської мови;
- аналіз існуючих рішень для вивчення італійської мови та визначення переваг і недоліків цих рішень;
- аналіз програмних засобів та вибір найкращих для реалізації вебсайту для вивчення італійської мови;
- визначення вимог до вебсайту для вивчення італійської мови з урахуванням недоліків існуючих рішень;
- проєктування та розробка вебсайту для вивчення італійської мови;
- тестування розробленого вебсайту.

Об'єкт дослідження – процес вивчення італійської мови.

Предмет дослідження – програмне забезпечення для вивчення італійської мови.

Робота пройшла апробацію на Всеукраїнській науково-технічній конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях», яка пройшла 24 квітня 2024 року в Державному університеті інформаційно-комунікаційних технологій. За результатами апробації опубліковано тези доповідей: “Аналіз існуючих рішень для вивчення італійської мови” та “Визначення вимог до вебсайту для вивчення італійської мови”.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Особливості процесу вивчення італійської мови

Згідно з результатами дослідження Мінцифри, USAID та КМІС 99% опитаних вважають потрібним вивчати іноземні мови. [1] Ця цифра не є дивною, адже знання кількох мов відкриває для людини багато нових можливостей: відсутність мовних бар'єрів при спілкуванні з іноземцями, доступ до більшого обсягу інформації та швидку адаптацію до нового місця перебування.

Вивчення іноземних мов покращує здоров'я завдяки постійному тренуванню мозку та поліпшенню пам'яті, що зменшує ризик прояву хвороби Альцгеймера та деменції. Згідно з результатами дослідження, завдяки активному використанню кількох мов покращується концентрація уваги у стресових ситуаціях. [2]

Також вивчення іноземної мови покращує знання рідної завдяки пошуку подібностей у лексиці, вивченню граматичних правил, структури слів та речень, побудові логічних зв'язків.

Вивчення італійської мови дає можливість краще розуміти культуру італійців. Італійська використовується у багатьох сферах: моді, кулінарії, музиці, мистецтві тощо. Знання італійської полегшить процес вивчення інших романських мов. Для співпраці з італійськими бізнес-партнерами також буде корисно знати італійську мову.

Для того, щоб мати можливість вільно використовувати італійську мову необхідно вивчити лексику, знати граматичні та фонетичні правила. Також вільному володінню італійською мовою допоможуть такі навички: сприйняття мови на слух, розуміння мови у контексті, усне мовлення, вміння побудови речень.

Вивчати італійську мову можна як з викладачем, так і самостійно. Перевагою занять з викладачем є структурованість матеріалу та наявність часових рамок для здачі завдань на перевірку, що робить процес вивчення мови швидшим.

Проте, навіть вивчаючи мову з викладачем більший відсоток успіху буде залежати від самостійної роботи студента.

Ключовою перевагою самостійного вивчення італійської мови є гнучкість навчального процесу: студент має можливість самостійно обирати графік навчання, вивчати теми, які задовольняють його інтереси та місце в якому відбуватиметься навчання.

Поради для самостійного вивчення італійської мови від Павійського університету [3]:

- ставте досяжні цілі;
- оберіть навчальні матеріали;
- збалансуйте своє навантаження: урізноманітніть свій розпорядок і насолоджуйтеся процесом навчання;
- навчіться приймати свої помилки;
- досліджуйте свої інтереси італійською мовою;
- зробіть звичкою вивчення мови кожного дня;
- словниковий запас є основою мови, але граматичні правила також важливі.

1.2 Аналіз існуючих рішень для вивчення італійської мови

Одними з основних способів вивчення італійської мови є відвідування мовних курсів або самостійне навчання. Для самостійного навчання використовуються різні ресурси: книги, відеокурси на YouTube або платформи для вивчення іноземних мов, ін.

Навчання з використанням комп'ютерної техніки в наш час є досить зручним. Багато викладачів, а також тих, хто вивчає іноземні мови використовують мобільні пристрої для вивчення мов через легкий доступ до ресурсів, зручність та менше мовне занепокоєння [4].

Використання різноманітного програмного забезпечення для вивчення іноземних мов може позитивно впливати на розуміння, знижувати тривожність,

підвищити мотивацію та сприяти взаємодії між учнями незалежно від віку чи рідної мови [5].

Нині існує безліч різноманітних вебсайтів та додатків для вивчення італійської мови, які мають як переваги, так і недоліки. Нижче наведено аналіз деяких з них.

1.2.1 Duolingo

Duolingo [6] – це платформа для вивчення іноземних мов, що пропонує короткі уроки, які допомагають засвоїти нову лексику та граматичні конструкції, покращити сприйняття речень на слух та потренувати вимову. Платформа має кілька секцій, розділених на теми, наприклад, говоримо про родину, черга у ресторані. Курс італійської доступний англійською мовою.

Кожна тема в Duolingo має кілька уроків, які складаються з завдань на переклад слів та речень з італійської на англійську та навпаки, сприйняття слів та речень на слух та вправ з перевіркою вимови. Під час письмового перекладу речень на італійську є можливість голосового вводу. На рисунку 1.1 зображено фрагмент уроку у вебверсії Duolingo.

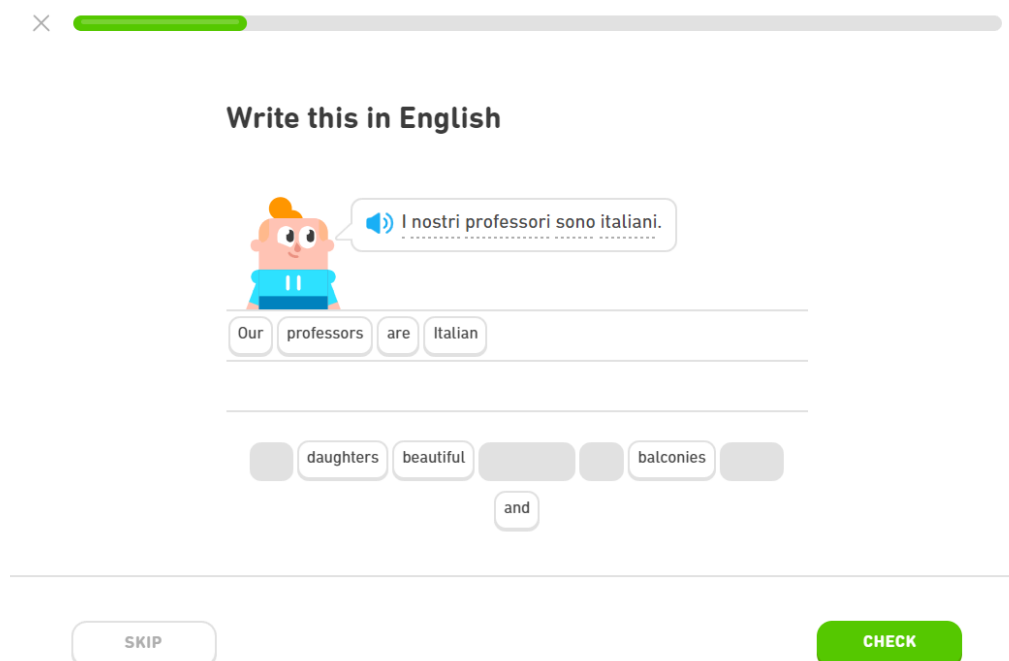


Рис. 1.1 Фрагмент уроку в Duolingo

Кожна секція курсу містить кілька граматичних правил, використання яких практикуватиметься під час проходження уроків секції. Кожна тема має кілька ключових фраз уроків, що розміщені у ній.

Також Duolingo пропонує студентам короткі історії, які створені у форматі діалогів персонажів та містять вправи для розуміння мови у контексті. Фрагмент однієї з історій зображено на рисунку 1.2.

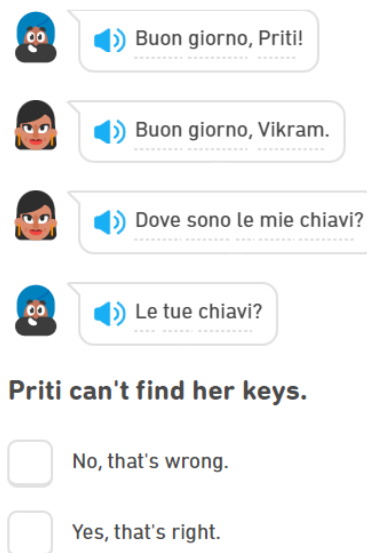


Рис. 1.2 Фрагмент історії в Duolingo

Duolingo має вебверсію, а також додатки для Android та iOS. Двигун Duolingo розроблений з використанням мови програмування Scala. [7]

Переваги вивчення італійської мови у Duolingo:

- наявність граматичних правил;
- наявність ключових фраз теми;
- наявність словника з вивченою лексикою;
- курс італійської має кілька рівнів.

До недоліків Duolingo можна віднести:

- відсутність української версії курсу італійської мови;
- відсутність фонетичних правил.

1.2.2 Mondly

Mondly [8] – це платформа для вивчення іноземних мов, що пропонує короткі уроки, які допомагають засвоїти нову лексику та граматичні конструкції, а також потренувати вимову. Уроки згруповані за темами, наприклад, родина, подорож, школа, ін. Кожна тема має декілька уроків, бесіду у чат-боті, а також повторення вивчених слів. Курс італійської доступний українською мовою.

Кожен урок в Mondly складається з декількох завдань серед яких: вибір перекладу слова, вибір пропущеного слова у реченні, переклад речення з італійської на українську та навпаки, перевірка вимови та сприйняття на слух. Під час письмового перекладу речень є можливість голосового вводу. На рисунку 1.3 зображено фрагмент уроку у вебверсії Mondly.

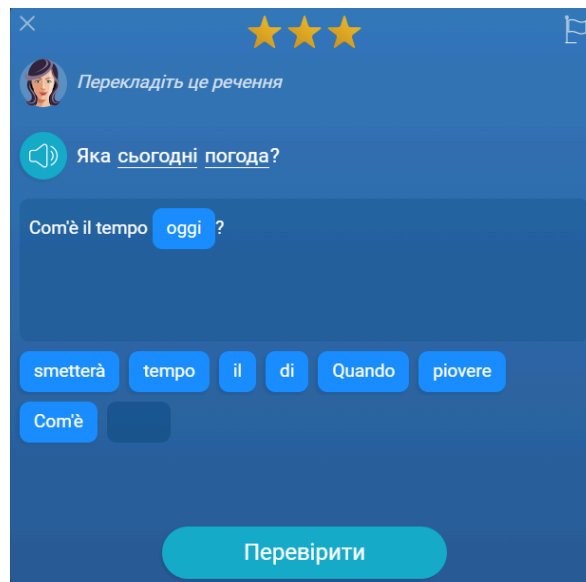


Рис. 1.3 Фрагмент уроку в Mondly

Mondly має вебверсію, а також додатки для Android та iOS. Додаток має чатбот з функцією розпізнавання голосу та можливість навчатись маючи вільні руки. [9]

Заняття у чат-боті Mondly проходять у форматі діалогу користувача з ботом. Бот ставить питання користувачу відповідь на яке вводиться голосом. Студент має можливість прослухати кілька з запропонованих додатком варіантів та обрати один з них або проговорити власну відповідь.

Слід зауважити, що власна відповідь може бути обрізаною до коротшого варіанту, який відобразиться на екрані. Наприклад, було проговорено фразу “Mi piace giocare a tennis” (мені подобається грати у теніс), бот її розпізнає, але в фінальному зарахуванні фрази виводить на екран лише слово tennis.

Фрагмент заняття у чат-боті зображено на рисунку 1.4.

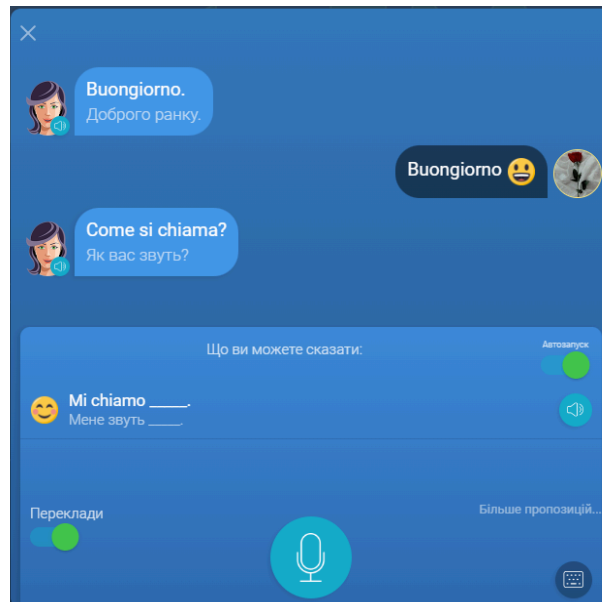


Рис. 1.4 Фрагмент заняття у чат-боті Mondly

Також Mondly дає можливість практикувати вимову завдяки запису готових фраз діалогів в однойменному розділі тем курсу. Однак, перевірка вимови не здійснюється, а користувач може записати будь-що для зарахування фрази.

Переваги вивчення італійської мови у Mondly:

- наявність української версії курсу італійської мови;
- можливість використання голосового керування;
- можливість практикувати діалоги, використовуючи чат-бот.

До недоліків Mondly можна віднести:

- відсутність граматичних та фонетичних правил;
- помилки в українському перекладі речень;
- перевірка вимови у розділі “Діалог” не здійснюється системою;
- часом некоректна робота чат-бота: обрізання проговорених фраз.

1.2.3 Drops

Drops [10] – це додаток для вивчення іноземних мов, що допомагає вивчити нові слова та фрази та тренує сприйняття мови на слух. Додаток пропонує уроки, згруповані за темами, тривалістю від 5 до 15 хвилин під час яких користувачі вивчають нову лексику. Курс італійської доступний українською мовою.

Уроки в Drops спрямовані на запам'ятовування нової лексики шляхом асоціацій слів та зображень. На рисунку 1.5 надано фрагмент уроку в Drops.



Рис. 1.5 Фрагмент уроку в Drops

Крім звичайних уроків Drops пропонує студентам режим вікторини у якому ставиться питання, що має кілька правильних варіантів відповіді. Цей режим допомагає краще засвоювати лексику завдяки зв'язуванню нової інформації з уже відомою. Фрагмент вікторини у Drops зображено на рисунку 1.6.

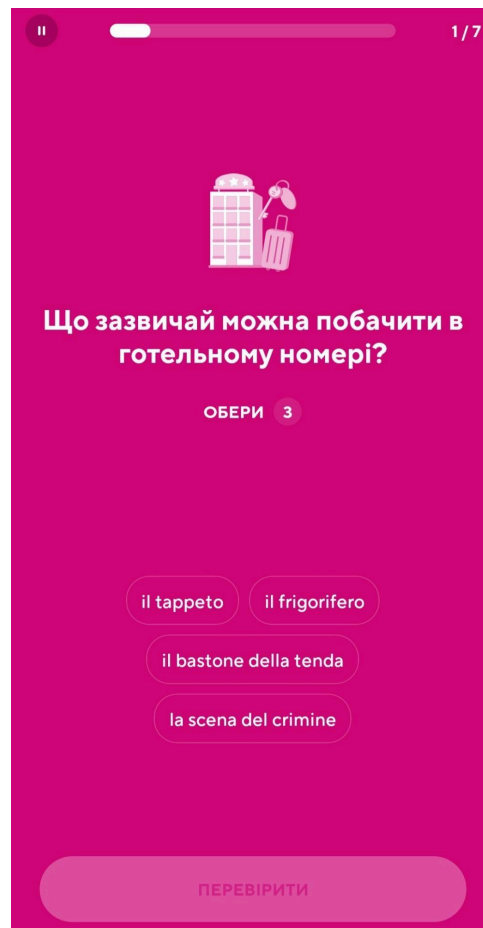


Рис. 1.6 Фрагмент вікторини у Drops

Drops має вебверсію, а також додатки для Android та iOS.

Переваги вивчення італійської мови у Drops:

- наявність української версії курсу італійської мови;
- наявність вікторин, які допомагають підтримувати словниковий запас;
- наявність словника, що має функцію пошуку.

До недоліків Drops можна віднести:

- додаток призначений лише для поповнення словникового запасу.

1.2.4 Busuu

Busuu [11] – це платформа для вивчення іноземних мов, що пропонує короткі уроки, які допомагають засвоїти нову лексику, містять граматичні правила, покращують сприйняття речень на слух та тренують вимову; є можливість перевірки завдань носіями. Курс італійської доступний англійською мовою.

Уроки у Busuu згруповані за темами, що відкриваються поступово, але є можливість скласти іспит за темою, щоб перестрибнути у наступну. Уроки у додатку мають різні типи завдань: поповнення словникового запасу, практика читання, перевірка сприйняття на слух. Наявна теоретична частина у якій пояснюються граматичні та фонетичні правила італійської мови. Слова і фрази записані носіями мови. На рисунку 1.7 надано фрагмент уроку у вебверсії Busuu.

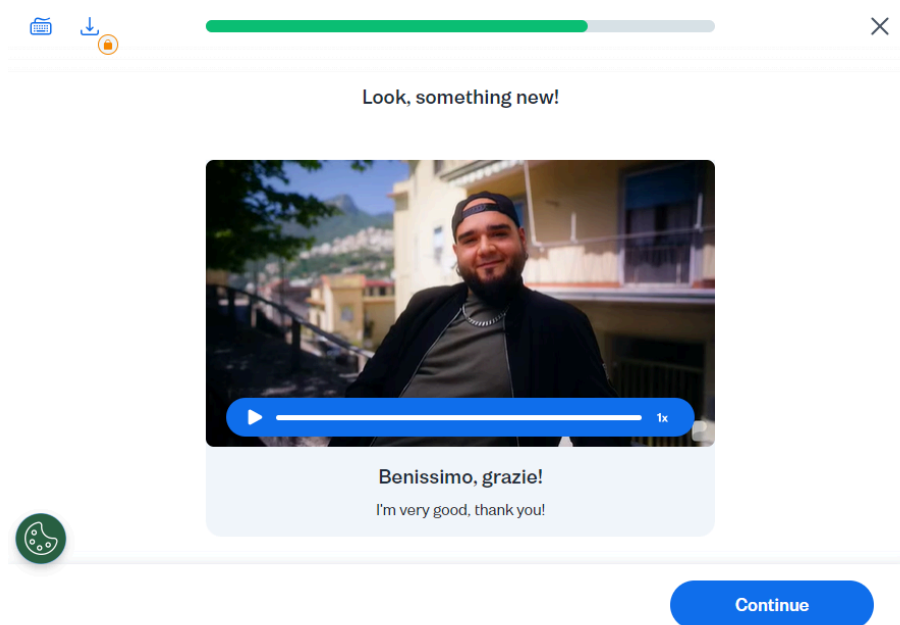


Рис. 1.7 Фрагмент уроку в Busuu

Busuu пропонує студентам деякі завдання, які дають можливість залишити свою відповідь на форумі, щоб носії мови перевірили його та залишили свій відгук. На рисунку 1.8 зображено приклад роботи цієї функції.

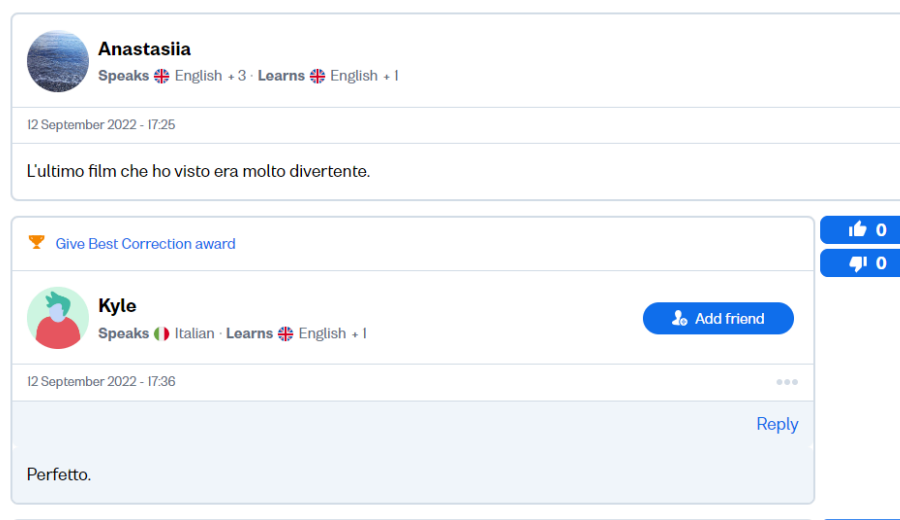


Рис. 1.8 Форум у Busuu

Busuu має вебверсію, а також додатки для Android та iOS. Вебдодаток розроблено з використанням React, для редагування контенту використовується Grav CMS. [12]

Переваги вивчення італійської мови у Busuu:

- наявність граматичних правил в уроках;
- наявність курсу з фонетичними правилами (у преміум версії);
- запис слів та фраз носіями мови;
- наявність форуму, який дозволяє носіям мови перевіряти завдання.

До недоліків Busuu можна віднести:

- відсутність української версії курсу італійської мови.

1.2.5 Memrise

Memrise [13] – це платформа для вивчення іноземних мов, що допомагає вивчити нові слова та фрази, які озвучені носіями. Курс італійської доступний англійською мовою.

Уроки в Memrise націлені на вивчення нової лексики та мають завдання на перевірку сприйняття мови на слух. Вправи в додатку бувають з вибором однієї правильної відповіді, складанням фрази у правильному порядку за допомогою кнопок або вводу перекладу слова італійською з клавіатури. На рисунку 1.9 зображено фрагмент уроку у вебверсії Memrise.



Рис. 1.9 Фрагмент уроку в Memrise

Memrise має вебверсію, а також додатки для Android та iOS.

Переваги вивчення італійської мови у Memrise:

- запис слів та фраз носіями мови;
- можливість тренувати сприйняття мови на слух.

До недоліків Memrise можна віднести:

- відсутність української версії курсу італійської мови;
- додаток призначений лише для поповнення словникового запасу.

1.2.6 uTalk

uTalk [14] – додаток для вивчення іноземних мов, що допомагає вивчити нову лексику та тренувати вимову. Курс італійської доступний українською мовою.

uTalk має курс італійської, який складається з шести ігор з картками, які є перекладом до слів. На рисунку 1.10 надано перелік ігор, які пропонує додаток:

- Вправа із фразами пропонує перелік слів та фраз з перекладом, які можна прослухати та записати власним голосом (без автоматичної перевірки).
- Гра (початківець) складається з чотирьох карток-зображень, які озвучуються на початку раунду, а потім перемішуються. Завданням користувача є прослухати фразу та знайти одну правильну картку.
- Розмовна гра пропонує користувачу записати фрази власним голосом, а потім шукати правильну картку. Додаток не здійснює перевірку вимови.
- Гра (профі) містить кілька карток-зображень, які кожен раунд збільшуються на одну. Завданням користувача є прослухати фразу та обрати правильну картку (одна з карток зайва).
- Гра на запам'ятовування складається з кількох карток, які потрібно запам'ятати після чого вони закриваються. Необхідно прослухати фразу та обрати правильну картку (одна з карток зайва). Кожен раунд кількість карток збільшується на одну. Після появи семи карток починається друга половина гри під час якої картки відкриті, треба слухати фрази та обирати правильні картки (одна зайва).

- Гра-пригадування показує картку та пропонує гравцю записати переклад голосом після чого програється правильна відповідь. Запис не перевіряється автоматично, а позначається правильним чи ні користувачем самостійно.

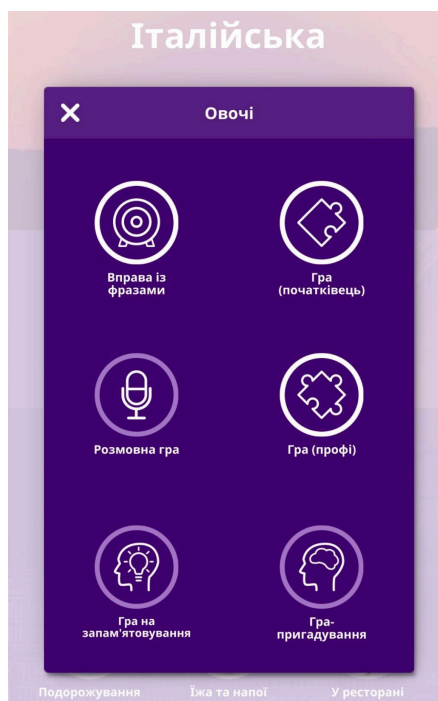


Рис. 1.10 Перелік ігор в uTalk

uTalk має додатки для Android та iOS.

Переваги вивчення італійської мови в uTalk:

- наявність української версії курсу італійської мови;
- можливість тренувати асоціації слово-зображення.

До недоліків uTalk можна віднести:

- додаток призначений лише для поповнення словникового запасу;
- правильність вимови не перевіряється системою.

1.2.7 Wlingua

Wlingua [15] – це платформа для вивчення іноземних мов, що пропонує короткі уроки, які допомагають засвоїти нову лексику, містять граматичні правила та тренують сприйняття мови на слух. Курс має декілька рівнів, а уроки згруповані за темами. Курс італійської доступний англійською мовою.

Кожен урок містить нову лексику, граматичне правило та вправи на його засвоєння. Також наявні уроки, які направлені на роботу зі сприйняття тексту на слух. На рисунку 1.11 зображено фрагменти уроку у вебверсії Wlingua.

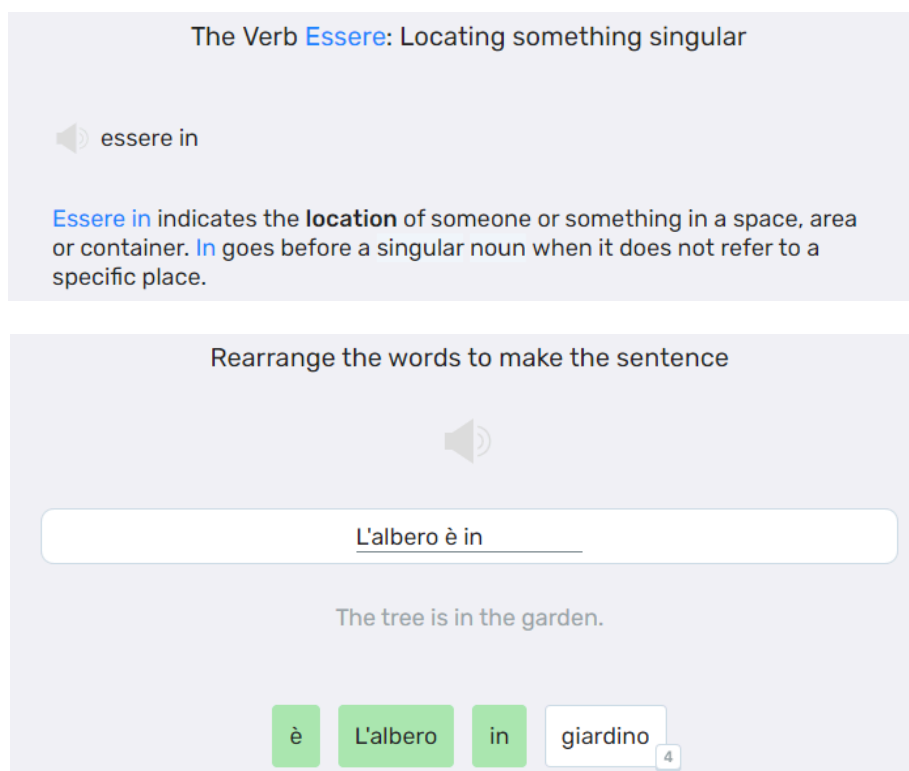


Рис. 1.11 Фрагменти уроку у Wlingua

Wlingua має вебверсію, а також додатки для Android та iOS.

Переваги вивчення італійської мови у Wlingua:

- курс італійської має кілька рівнів;
- курс має граматичні правила.

До недоліків Wlingua можна віднести:

- відсутність української версії курсу італійської мови;
- відсутність фонетичних правил.

Зведені результати аналізу характеристик розглянутих вебсайтів та додатків наведено у таблиці 1.1.

Таблиця 1.1

Зведені результати аналізу характеристик вебсайтів
та додатків для вивчення італійської мови

Функціонал	Duolingo	Mondly	Drops	Busuu	Memrise	uTalk	Wlingua
Платформи	Android, iOS, Web	Android, iOS, Web	Android, iOS, Web	Android, iOS, Web	Android, iOS, Web	Android, iOS	Android, iOS, Web
Мова курсу італійської	англійська	українська англійська	українська англійська	англійська	англійська	українська англійська	англійська
Нова лексика	+	+	+	+	+	+	+
Граматичні правила	+	-	-	+	-	-	+
Фонетичні правила	-	-	-	+(платно)	-	-	-
Тренування сприйняття на слух	+	+	+	+	+	+	+
Вправи на говоріння	+	+	-	+	-	+	-
Поділ курсу на рівні	+	-	-	+	-	-	+
Інші особливості	наявність історій; словник з вивченою лексикою	наявність чат-бота; можна вивчати мову маючи вільні руки	наявність вікторин; словник з лексикою; додаток лише для вивчення лексики	наявність форуму де завдання перевіряються носіями мови	фрази та слова озвучені носіями мови; тільки для вивчення лексики	додаток лише для вивчення лексики	дипломи за завершення рівнів (лише у преміум версії)

Беручи до уваги аналіз програмного забезпечення, яке може бути використано для вивчення італійської мови, можна виділити ключовий недолік наявних рішень: основна мета призначення – поповнення словникового запасу.

Більшість з ресурсів не мають граматичних та фонетичних правил, а також словника, куди зібрана лексика курсу.

1.3 Огляд засобів реалізації вебсайту для вивчення італійської мови

Вебсайт для вивчення італійської мови розробляється для веббраузера, що забезпечить багатоплатформове використання внаслідок розробки адаптивної версії.

Основними засобами для розробки вебсайтів є мова розмітки HTML, мова таблиць стилів CSS та мова програмування JavaScript. Нижче наведено їх короткий огляд.

1.3.1 HTML

HTML – це основний будівельний блок Інтернету, який визначає зміст і структуру вебконтенту. [16] На вебсайті відповідає за каркас сторінок.

Цеглинками, що закладають основу вебсторінки є HTML-елементи. Вони складаються з контенту, що обгорнутий у теги. На рисунку 1.12 зображено приклад HTML-елементу.

```
<div>This is a content of the div.</div>
```

Рис. 1.12 Приклад HTML-елементу

Теги бувають парними, тобто такими, що мають відкриваючий та закриваючий теги (<p></p>, <a>, <div></div>, ін.) та непарними (
, <hr>, ін.). Теги можуть мати атрибути, приклад якого зображено на рисунку 1.13: *type* та *name* для типу та назви полів форми.

```
<input type="text" name="username">  
<input type="email" name="email">
```

Рис. 1.13 Приклад атрибутів HTML-тегів

HTML-теги мають два атрибути, що використовуються для подальшої роботи з тегами, наприклад, для їх стилізації: *class* та *id*. Елементів з однаковими назвами класів може бути декілька, а для кожного ідентифікатора може бути лише один унікальний елемент. Приклад атрибутів *class* та *id* зображено на рис. 1.14.

```
<h2 class="div_title">Title</h2>
<p id="paragraph_3">This is the third paragraph.</p>
```

Рис. 1.14 Приклад атрибутів HTML-тегів: class та id

Деякі теги можна вкладати один в одного, наприклад, теги списку, але потрібно правильно розмістити відкриваючі та закриваючі теги, інакше браузер може відобразити вміст сторінки некоректно. Приклад вкладеності тегів зображено на рисунку 1.15.

```
<ul>
  <li>List element 1.</li>
  <li>List element 2.</li>
  <li>List element 3.</li>
</ul>
```

Рис. 1.15 Приклад вкладеності HTML-тегів

Кожна HTML-сторінка складається з кількох базових тегів:

- `<!DOCTYPE html>` це оголошення стандарту документа;
- парних тегів `<html></html>`, що можуть мати атрибут *lang*, який вказує на основну мову контенту сторінки.
- парного тегу `<head></head>`, що містять наступні дані:
 - `<meta charset="UTF-8">` відповідає за кодування символів сторінки;
 - `<meta name="viewport" content="width=device-width, initial-scale=1.0">` відповідає за відображення сторінки відповідно до розмірів екрана пристрою. Використовується при адаптації сторінки під різні пристрої;
 - `<title>Document</title>` це назва сторінки, яка буде відображатись у браузері, в цьому випадку це *Document*;

- зазвичай перед закриваючим тегом `</head>` розміщується посилання на файл CSS зі стилями сторінки.
Приклад: `<link rel="stylesheet" href="style.css">`, де *style.css* це назва файлу стилів.
- парного тегу `<body></body>` всередині яких розміщено вміст сторінки.
Перед закриваючим тегом `</body>` можуть розміщуватися посилання на файл зі скриптом створеним за допомогою мови програмування JavaScript.
Приклад: `<script src="script.js"></script>`, де *script.js* це назва файлу JS-скрипту.

На рисунку 1.16 зображено базову структуру HTML-сторінки. Під час перегляду цієї сторінки у браузері відобразиться заголовок першого рівня з текстом “Title” та абзац тексту *lorem ipsum*, що використовується для проектування макетів сайтів.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Title</h1>
  <p>Lorem ipsum dolor sit amet consectetur adipiscing elit. Nunc, nobis
  eligendi sapiente pariatur quasi unde quidem corrupti ab quaerat rerum atque
  consectetur laboriosam maxime nam, nulla repudiandae natus incidunt. Aperiam.</
  p>
</body>
</html>
```

Рис. 1.16 Базова структура HTML-сторінки

Для створення звичайної вебсторінки достатньо використати лише HTML-розмітку. Проте така сторінка буде виглядати майже як звичайний текстовий документ, адже браузер підлаштує відображення контенту сторінки під базові стилі, які має кожен HTML-елемент. Щоб вигляд вебсайту був іншим, красивішим та унікальним використовують CSS.

1.3.2 CSS

CSS – це мова таблиць стилів, яка використовується для опису представлення документа, написаного на HTML. CSS описує, як елементи повинні відображатися на екрані. [17] Допомагає змінити стандартні стилі браузера та зробити вебсайт привабливіше.

Щоб змінити вигляд HTML-елементу зі стандартного необхідно прописати CSS-правило. Кожне правило складається з селектора, властивості (або кількох властивостей) та значення властивості:

- селектор дозволяє звернутися до HTML-елемента, що потрібно стилізувати. Селектори бувають кількох типів, докладніше у таблиці 1.2;
- кожен HTML-елемент має властивості, які можна змінювати за допомогою CSS, наприклад, властивість `font-size` відповідає за розмір шрифту.
- прикладом значення властивості може бути колір, який можна вказати словом *blue*, або вказавши код кольору, наприклад, HEX-код: `#00008b` для синього кольору.

Таблиця 1.2

Типи селекторів

Тип селектора	Приклад	Що обирає
Селектор елемента	a	Обирає всі елементи <a>
Селектор ідентифікатора	#element_id	Обирає HTML-елемент з id="element_id"
Селектор класу	.class_name	Обирає всі елементи з class="class_name"
Унікальний селектор	*	Обирає всі елементи
Селектор елемента, що має клас	a.class_name	Обирає всі елементи <a> з class="class_name"
Селектор атрибута	div[lang]	Обирає всі елементи <div> з атрибутом [lang]

Типи селекторів

Тип селектора	Приклад	Що обирає
Селектор псевдокласу	a:hover	Обирає всі елементи <a> на які наведено курсор.
Селектор псевдоелемента	a::after	Обирає всі псевдоелементи ::after елементів <a>

Щоб мінімізувати код однакові властивості кількох елементів групуються завдяки селектору групування. На рисунку 1.17 зображено приклад групування селекторів. Всі елементи <h1> та <h2> будуть відцентровані, змінять свій колір на сірий та будуть підкреслені.

```
h1, h2 {
  text-align: center;
  color: #3b3b3b;
  text-decoration: underline;
}
```

Рис. 1.17 Приклад групування селекторів

Селектори також можуть мати комбінатори, які подані у таблиці 1.3. Завдяки їх використанню створюються складні селектори – це використання комбінаторів між двома простими селекторами.

Таблиця 1.3

Комбінатори

Назва	Приклад	Для чого потрібен
Комбінатор нащадків	div a	Обирає всі елементи <a>, які містяться всередині HTML-елемента <div>
Дочірній комбінатор	div > a	Обирає всі елементи <a>, які є дочірніми для HTML-елемента <div>

Комбінатори

Назва	Приклад	Для чого потрібен
Загальний батьківський комбінатор	div ~ a	Обирає всі елементи <a>, які є родичами для HTML-елемента <div>
Сусідній родинний комбінатор	div + a	Обирає всі елементи <a>, які є суміжними, тобто розміщені наступними після елемента <div>

На рисунку 1.18 зображено приклад CSS-правил. Під час перегляду вебсторінки, до якої застосовано файл з цими стилями, заголовок першого рівня відцентрується та змінить свій колір зі стандартного на блакитний, а абзац тексту займе положення по центру, змінить свою ширину з базової на 750px, а розмір тексту зі стандартного зміниться на 18px.

```

h1 {
  text-align: center;
  color: #00008b;
}
p {
  margin: 0 auto;
  width: 750px;
  font-size: 18px;
}

```

Рис. 1.18 Приклад CSS-правил

Зв'язати стилі CSS з HTML-документом можна двома способами: за допомогою посилання у хедері на файл зі стилями (найпопулярніший спосіб) або прописувати CSS-правила безпосередньо у HTML-документі всередині парного тегу <style></style>, який розміщується у хедері документа. Також можна записувати CSS-властивості всередині HTML-тегів, використовуючи атрибут *style*, але цей спосіб рідко використовується (наприклад, у випадках, коли пріоритет правила має бути найвищим).

Вебсайт, який створено лише з використанням HTML та CSS є статичним, тобто незмінним. Такі вебсайти можуть змінювати лише розробники, адже мають доступ до файлів початкового коду сайту. Відвідувачі мають можливість лише переглядати вміст цих ресурсів: не тільки текстову інформацію, а й мультимедійну (зображення, відео, аудіо).

Для більшої взаємодії користувача з сайтом він має бути динамічним. Контент таких ресурсів може змінюватися залежно від певних умов або дій, наприклад, оновлення стрічки новин при натисканні на кнопку. Для розробки динамічних вебсайтів використовується JavaScript.

1.3.4 JavaScript

JavaScript – це багатопарадигмальна, однопоточна, динамічна мова, що базується на прототипах, підтримує об'єктноорієнтовані, імперативні та декларативні (наприклад, функціональне програмування) стилі. [18] Додає на вебсайт інтерактивні елементи: форми реєстрації або входу, модальні вікна, каруселі зображень, вікторини, пошук елементів у списку, аудіоелементи, які відтворюються після клацання по картинці, показ прихованого контенту при натисканні на кнопку та ін.; оброблює події. Також за допомогою JavaScript можна працювати з вмістом HTML-елементів, наприклад, змінити текст на кнопці за певної умови.

Для того, щоб мати можливість працювати з HTML-елементами у функціях JavaScript необхідно оголосити змінну за допомогою ключового слова *const* та присвоїти їй як значення один з методів об'єкта Document для роботи з об'єктом Element, наприклад, метод *querySelector()*. На рисунку 1.19 зображено приклад отримання HTML-елемента з класом *class-name*. Надалі цей елемент можна використовувати у функціях або подіях JS звертаючись за назвою змінної *element*.

```
const element = document.querySelector(".class-name");
```

Рис. 1.19 Приклад отримання HTML-елемента з класом *class-name*

Завдяки властивості *style* об'єкта *Element* існує можливість взаємодіяти з CSS-властивостями та змінювати їх за допомогою JavaScript. На рисунку 1.20 зображено зміну кольору HTML-елемента на синій, отриманого за допомогою оголошеної раніше змінної *element*.

```
element.style.color = "blue";
```

Рис. 1.20 Приклад зміни CSS-властивості HTML-елемента

Якщо потрібно змінити вміст HTML-елемента, наприклад, текст кнопки після її натискання, можна використати властивість *innerHTML* об'єкта *Element*. Текст, який був на кнопці до виконання певної умови, зміниться на *Show More*. Приклад коду для зміни вмісту кнопки зображено на рисунку 1.21.

```
const button = document.querySelector("#btn");
button.innerHTML = "Show More";
```

Рис. 1.21 Приклад зміни вмісту HTML-елемента

На рисунку 1.22 зображено приклад JavaScript коду для показу більшого обсягу тексту після натискання кнопки. Скрипт має дві змінні, що отримують елементи, які є кнопкою та додатковим блоком тексту, за id-ідентифікаторами з HTML-документа. Функція скрипту має умову, яка перевіряє відображення додаткового блоку тексту, показує його після натискання на кнопку, якщо текст було приховано та навпаки, а також змінює напис на кнопці відповідно до умови.

```
const btn = document.querySelector('#showMore-btn');
const moreText = document.querySelector('#showMore-text');

function showMore() {
  if (moreText.style.display === "none") {
    moreText.style.display = "block";
    btn.innerHTML = "Показати менше";
  } else {
    moreText.style.display = "none";
    btn.innerHTML = "Показати більше";
  }
}
```

Рис. 1.22 Приклад JavaScript коду для показу більшого обсягу тексту

Для запуску функції до HTML-тегу кнопки необхідно додати подію onclick, код якої зображено на рисунку 1.23.

```
<button id="showMore-btn" onclick="showMore()">Показати більше</button>
```

Рис. 1.23 Подія onclick для запуску функції showMore()

1.4 Визначення вимог до вебсайту для вивчення італійської мови

Після проведення аналізу особливостей процесу вивчення італійської мови та визначення ключових недоліків існуючих рішень для самостійного вивчення італійської мови можна сформулювати вимоги до майбутнього програмного продукту.

Вебсайт для вивчення італійської мови повинен бути реалізованим для веббраузера та забезпечувати наступні функції:

- вивід результатів тестів одразу після проходження;
- слова та фрази озвучуються після натискання по іконці вушка;
- функція пошуку перекладу слів у словнику.

Також було сформовано наступні нефункціональні вимоги:

- можливість переглядати вебсайт на різних пристроях, що мають веббраузер: комп'ютерах, планшетних ПК або смартфонах;
- українська локалізація інтерфейсу вебсайту;
- наявність уроків, що мають теорію та тести;
- наявність уроків, що мають граматичні правила;
- наявність уроків, що мають фонетичні правила;
- наявність діалогів, що забезпечують розуміння мови в контексті;
- можливість переглядати переклад фраз у діалогах.

РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБСАЙТУ

2.1 Аналіз варіантів використання вебсайту

Діаграма варіантів використання (або use-case діаграма) використовується для графічного представлення взаємодії між користувачами та системою.

Складовими елементами use-case діаграми є:

- актор, який позначає користувача або пристрій, що використовує систему;
- прецедент або варіант використання – це еліпс, який використовується для позначення окремого варіанту використання: назва дії, яку може виконати користувач або система;
- межа системи – це прямокутник з назвою системи розміщеною у верхній частині, що позначає систему, яка моделюється;
- примітка – це прямокутник, що має загнутий верхній правий куточок. Використовується для розміщення додаткової інформації, яка є уточненням до конкретних варіантів використання;
- відношення (зв'язки) бувають різних типів, вказують на взаємодію акторів з варіантами використання. Також кілька прецедентів можуть бути зв'язані між собою, але при цьому не мати зв'язків з акторами.

Типи відношень на діаграмі варіантів використання:

- асоціація вказує на зв'язок актора та варіанту використання. Позначається суцільною лінією;
- включення вказує на те, що поведінка одного варіанту використання є складовим компонентом послідовності поведінки іншого прецеденту (дочірній варіант використання є обов'язковим для базового). Позначається пунктирною лінією зі стрілкою та має стереотип <<include>>. Лише два прецеденти можуть бути зв'язані відношенням включення між собою;

- розширення вказує на взаємозв'язок варіанту використання з іншим, який не є обов'язковим, а здійснюється при виконанні певних додаткових умов. Позначається пунктирною лінією зі стрілкою та має стереотип <<extend>>;
- узагальнення вказує на зв'язки батьківських прецедентів з дочірніми. Позначається суцільною лінією з пустим трикутником на кінці.

На рисунку 2.1 зображено діаграму варіантів використання вебсайту зі сторони Користувача. Користувач може переглядати список уроків, які бувають трьох типів: уроки з теорією, фонетичні правила та граматичні правила, обирати урок зі списку, проходити тест та переглядати результати одразу після проходження, шукати переклад слова у словнику, переглядати список діалогів, переглядати діалог, переглядати переклад фраз діалогу, а також слухати як звучить слово або фраза в уроці, словнику або діалозі.

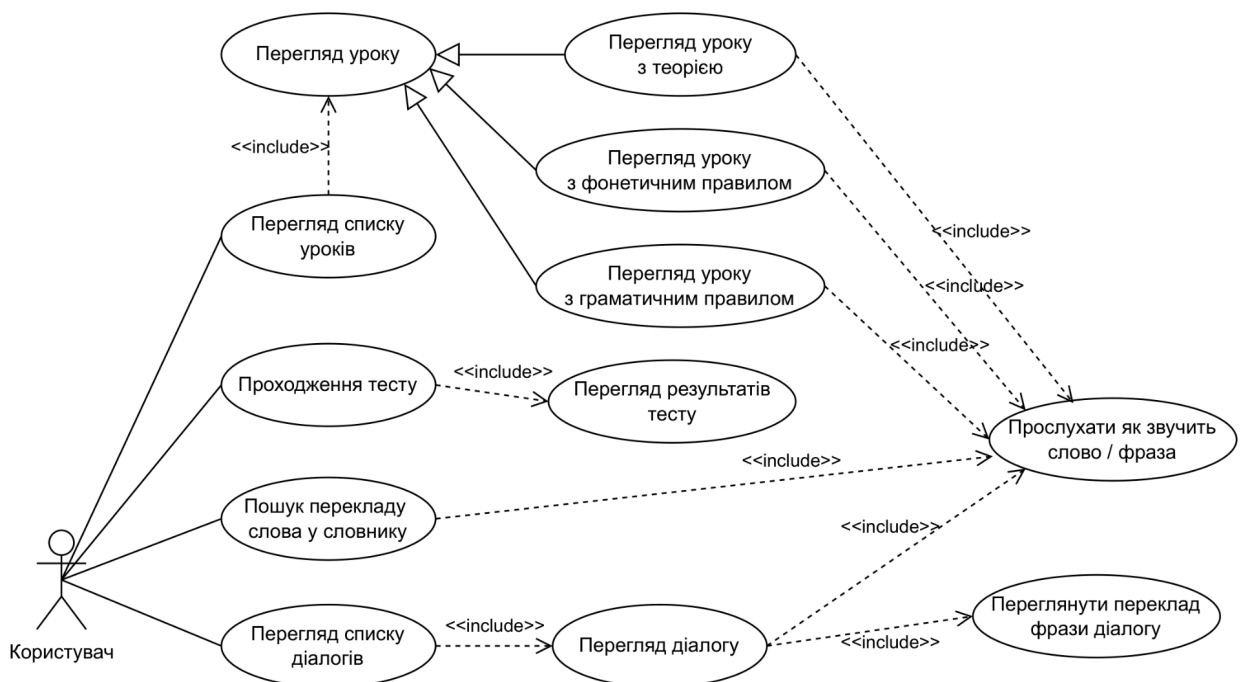


Рис. 2.1 Діаграма варіантів використання вебсайту

2.2 Проектування структури вебсайту

Структура вебсайту – це схема розташування сторінок вебсайту. Впливає на зручність ресурсу для користувача, а також розподіл посилальної ваги сторінок.

Виділяють наступні типи структур вебсайтів:

- ієрархічна або деревоподібна. Від головної сторінки йдуть посилання на розділи. Своєю чергою, розділи можуть поділятися на підрозділи. Цей варіант підходить для великих багатосторінкових сайтів, наприклад, інтернет-магазинів;
- лінійна. Кожна сторінка має одне посилання на наступну. Перегляд сторінок відбувається по ланцюжку: користувач відвідує сторінки одну за одною. Цей варіант підійде для простих вебсайтів, наприклад, сайтів-візиток;
- матрична або павутинна. Кожна сторінка має посилання на наступні. Надає користувачу повну свободу відвідування пов'язаних між собою сторінок. Цей варіант часто використовують для великих сайтів з взаємопов'язаним контентом, наприклад, e-commerce або новинних порталів.
- база даних або динамічна. Навігація по вебсайтах з такою структурою часто відбувається за допомогою пошуку або фільтрів. Цей тип структури використовують для соціальних мереж, онлайн форумів, новинних порталів, великих e-commerce проектів або для сервісів онлайн-банкінгу.
- гібридна. Комбінує різні типи структур вебсайтів. Прикладом такого поєднання може бути одночасне використання ієрархічної та лінійної структури на вебсайті.

Було обрано деревоподібну структуру вебсайту, яку зображено на рисунку 2.2:

- головна сторінка вебсайту (міститься у файлі index.html);
- розділ вебсайту зі списком уроків з теорією (lessons.html);

- сторінки вебсайту з уроками, що мають теорію (lesson_01.html ... lesson_n.html);
- розділ зі списком уроків з фонетичними правилами (phonetics.html);
- сторінки вебсайту з уроками з фонетичними правилами (phonetics_01.html ... phonetics_n.html);
- розділ зі списком уроків з граматичними правилами (grammar.html);
- сторінки вебсайту з уроками з граматичними правилами (grammar_01.html ... grammar_n.html);
- розділ вебсайту зі словником курсу (dictionary.html);
- розділ вебсайту зі списком діалогів (dialogues.html);
- сторінки з діалогами (dialogue_01.html ... dialogue_n.html).

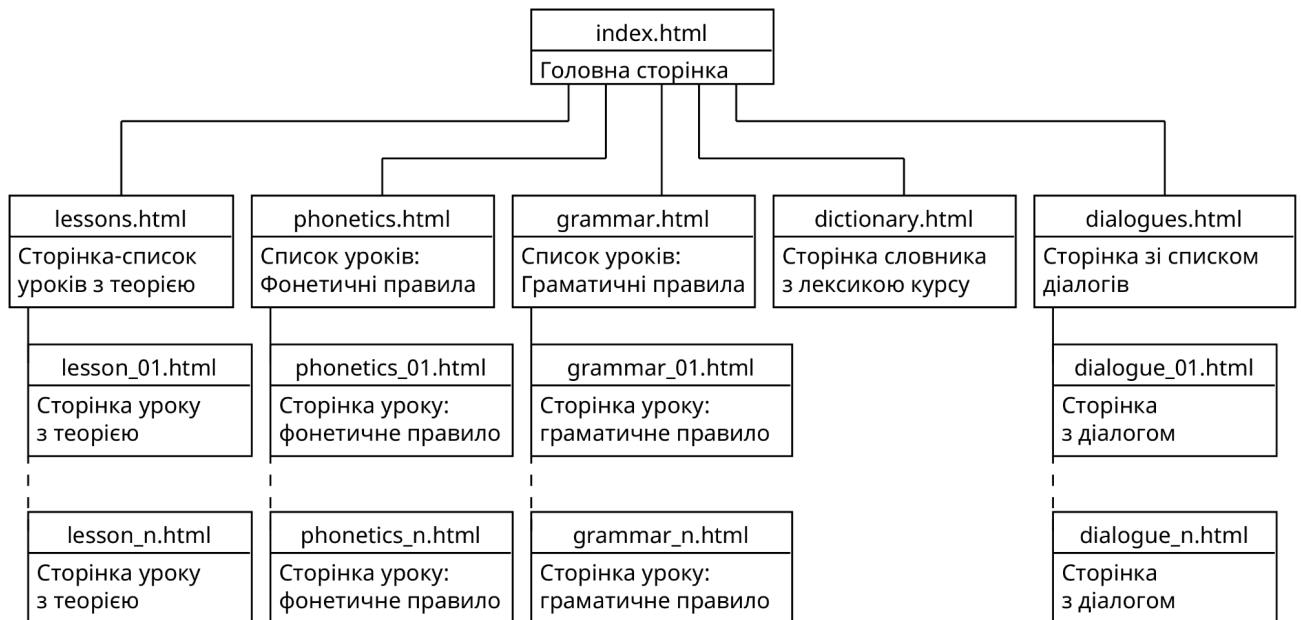


Рис. 2.2 Схема структури вебсайту

2.3 Проєктування графічного інтерфейсу вебсайту

Для проєктування графічного інтерфейсу вебсайту було використано графічний редактор Figma. Було створено макети головної сторінки, сторінки зі списком уроків, сторінки уроку, сторінки словника та сторінки діалогу. На рисунку 2.3 зображено загальний вигляд макета вебсайту.

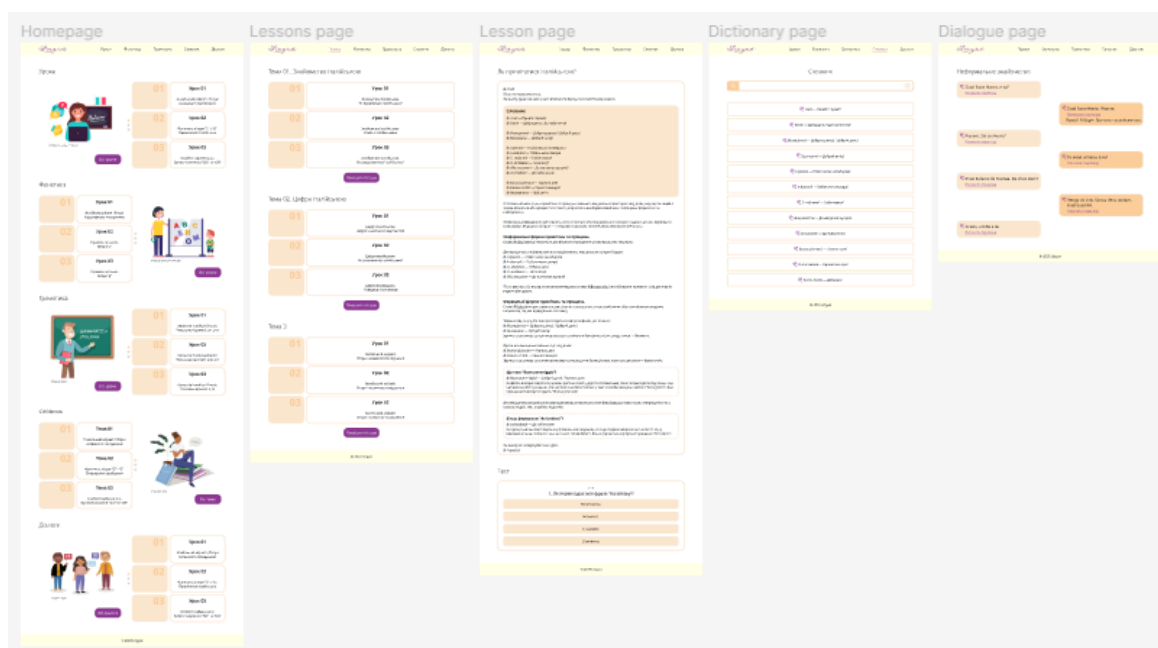


Рис. 2.3 Макет вебсайту у графічному редакторі Figma

Основні сторінки вебсайту зображено на рисунках 2.4 - 2.8:

- головна сторінка вебсайту (рис. 2.4);
- сторінка уроку (рис. 2.5) та тест на сторінці уроку (рис. 2.6);
- сторінка словника (рис. 2.7);
- сторінка діалогу (рис. 2.8).

Головна сторінка вебсайту (рис. 2.4) має 5 слайдшопу для кожного розділу слайди яких містять посилання на уроки з теорією, уроки з фонетичними правилами, уроки з граматичними правилами, словник та діалоги. Також кожна секція головної сторінки містить посилання у вигляді кнопки на відповідні розділи вебсайту для своєї секції, наприклад, посилання на сторінку зі списком діалогів для секції діалоги.

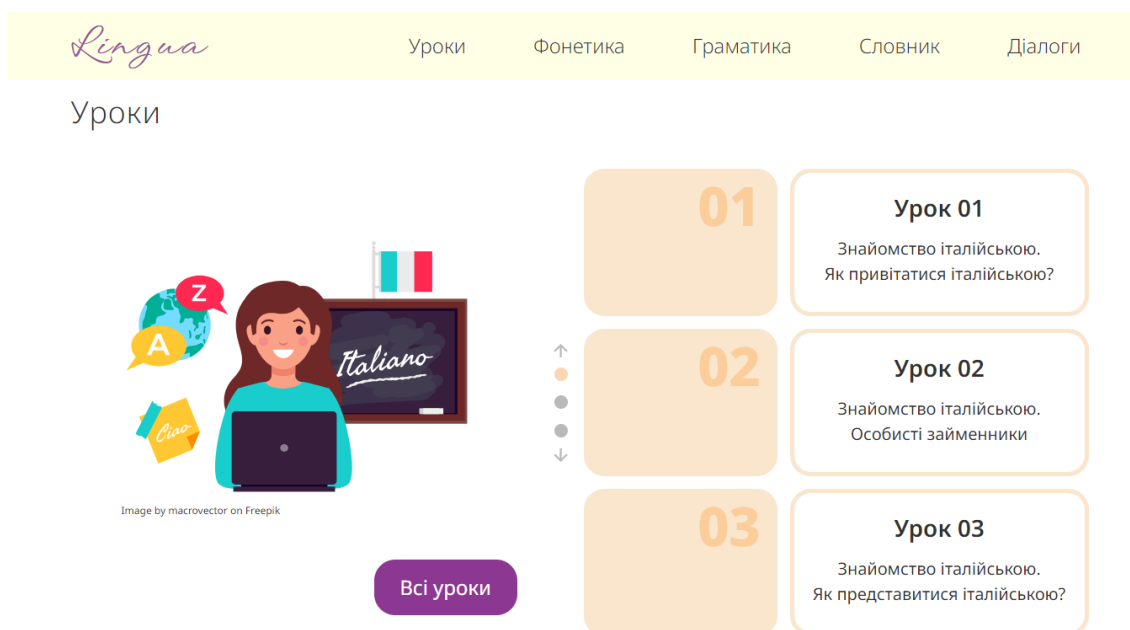


Рис. 2.4 Фрагмент головної сторінки вебсайту

Кожен урок на вебсайті складається з двох головних секцій: теоретичної частини та тесту для закріплення пройденого матеріалу. Нижче наведено фрагмент теоретичної частини уроку (рис. 2.5) та тест до цього уроку (рис. 2.6). Кожна сторінка уроку з теорією містить словник, що має слова, використання яких буде розглядатися під час уроку. Слова окрім перекладу мають аудіозапис звучання, який відтворюється після натискання іконки вушка, що передує тексту.

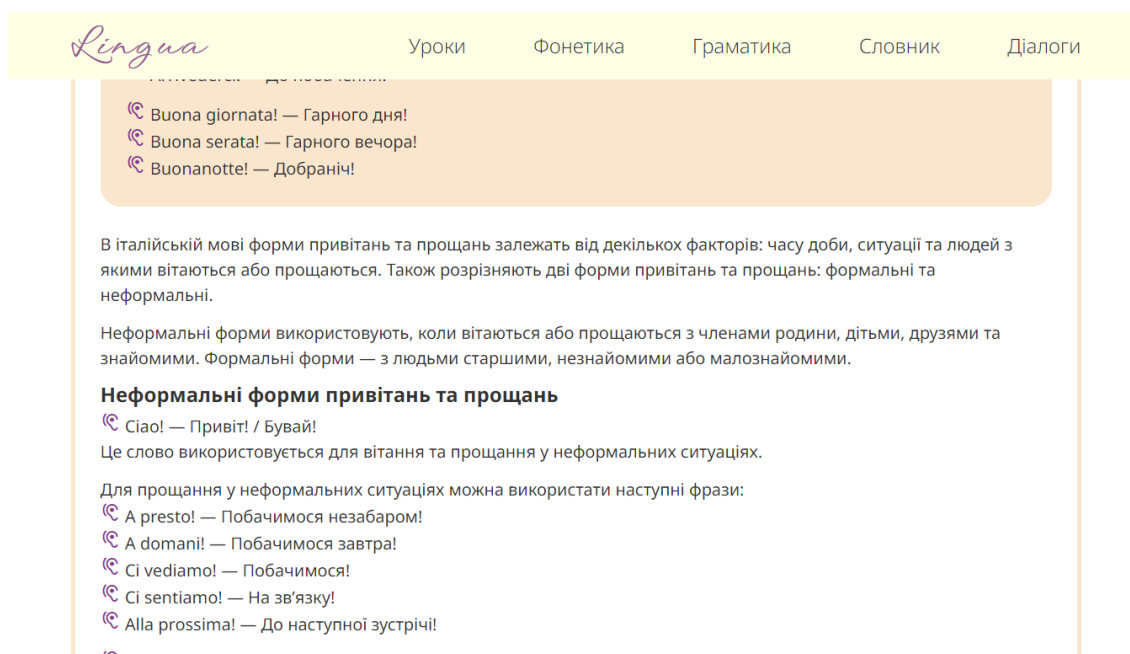


Рис. 2.5 Фрагмент теоретичної частини уроку

Кожен тест має чотири варіанти відповідей одна з яких є правильною. При виборі правильного варіанту кнопка з текстом відповіді виділяється зеленим кольором. При виборі неправильного варіанту обрана кнопка буде червоного кольору, а правильний варіант виділиться зеленим. Одразу після проходження тесту користувачу оголошується кількість правильних відповідей.

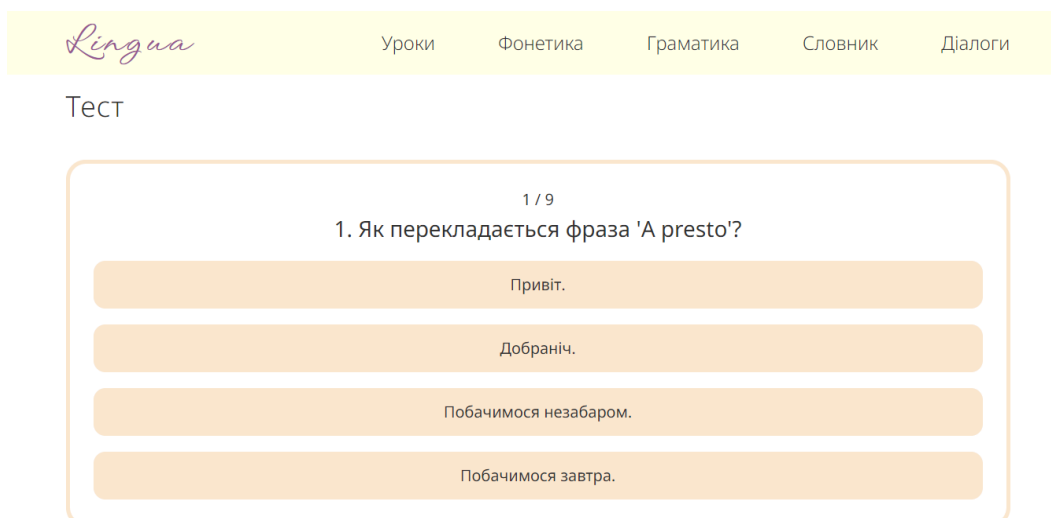


Рис. 2.6 Тест на сторінці уроку

Словник (рис. 2.7) на вебсайті має функцію пошуку. За замовченням на сторінці є лише назва розділу та поле пошуку, а слова та фрази приховані. Під час вводу буквосполучень, що має слово італійською або українською, якщо буде знайдено збіг з наявними на сторінці словами або фразами, вони відобразяться.

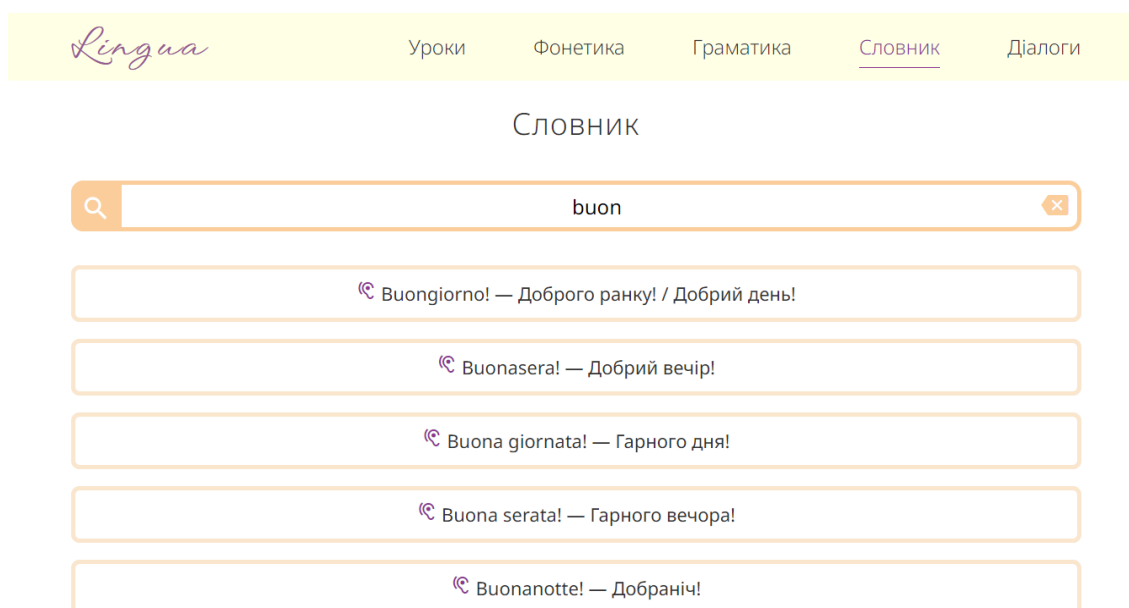


Рис. 2.7 Сторінка словника: пошук слів за запитом *buon*

Діалоги мають переклад фраз, який можна показати по кліку на посилання.

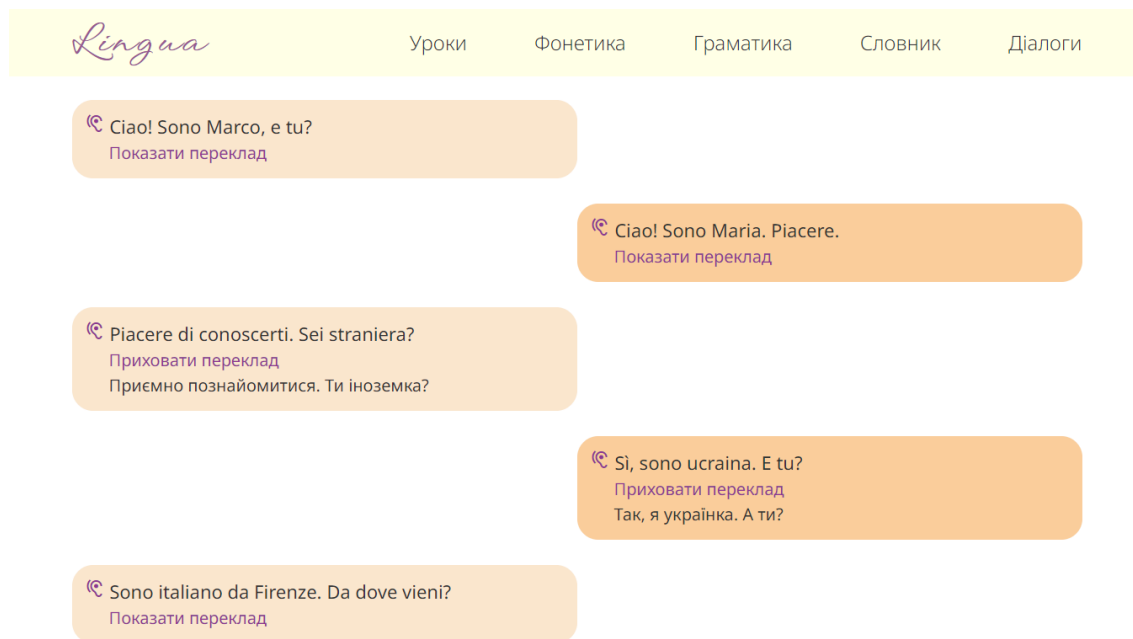


Рис. 2.8 Фрагмент сторінки діалогу

2.4 Розробка вебсайту

Для розробки вебсайту було використано мову розмітки HTML, мову таблиць стилів CSS та мову програмування JavaScript. Середовищем розробки було обрано Visual Studio Code.

Схему структури вебсайту на якій надано перелік розроблених HTML-сторінок зображено на рисунку 2.2 (див. підрозділ 2.2).

Було створено наступні файли CSS-стилів:

- скидання стандартних стилів (reset.css);
- головний файл стилів (style.css);
- стилі сторінки зі списком уроків (lessons.css);
- стилі сторінки уроку (lesson.css);
- стилі сторінки словника (dictionary.css);
- стилі сторінки діалогу (dialogue.css);
- стилі для адаптивної версії вебсайту (media.css).

Розроблено наступні js-скрипти, діаграму пакетів яких надано на рисунку 2.9:

- слайдшоу на головній сторінці, що містить посилання на уроки з теорією, уроки з фонетичними правилами, уроки з граматичними правилами, словник з лексикою та діалоги (файл `lessons_slider.js`);
- скрипт для показу прихованих блоків з посиланнями на уроки на сторінках зі списками уроків (`showMore_content.js`);
- функція для відтворення озвучки слів та фраз після натискання по іконці вушка (`playSound.js`);
- скрипти для роботи тестів на сторінках уроків (`test_ciao.js`, `test_alphabet.js`, `test_articles_un.js`, ..., `test_n.js`);
- скрипт для пошуку перекладу слів на сторінці словника (`search_words.js`);
- скрипт бургер-меню для адаптивної версії вебсайту (`burger_menu.js`).

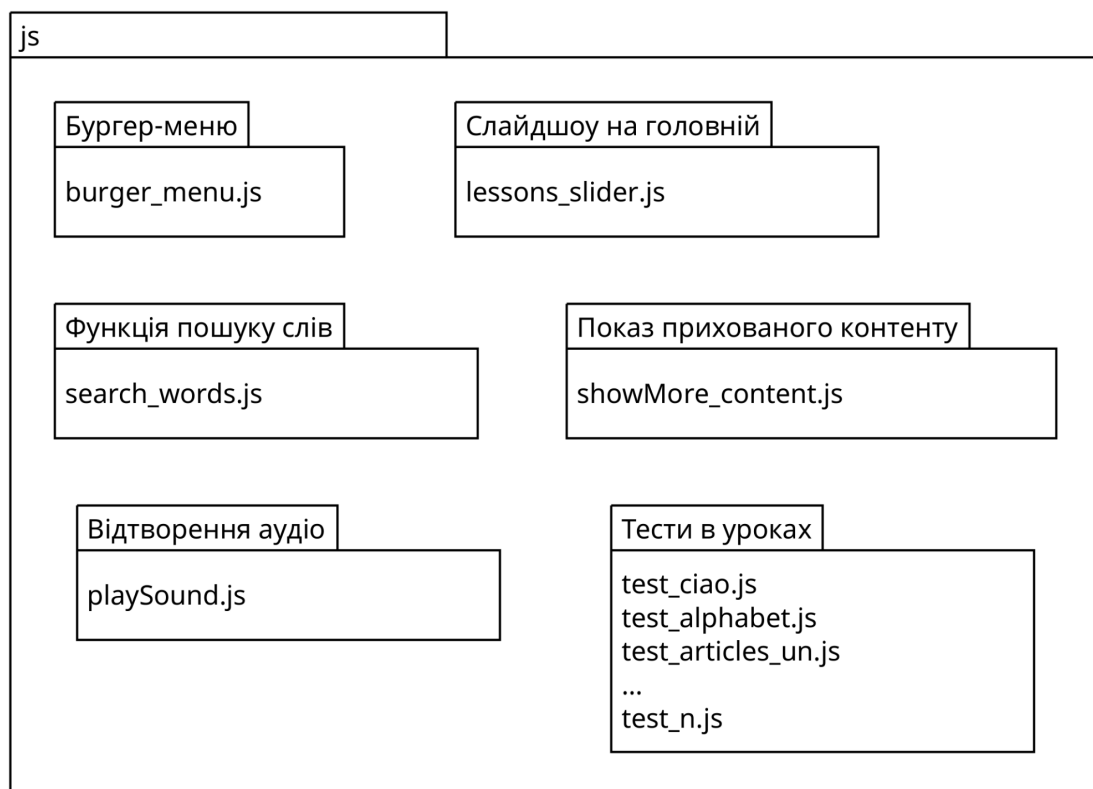


Рис. 2.9 Діаграма пакетів скриптів вебсайту.

Скрипт слайдшоу розроблявся для одночасної роботи з кількома слайдшоу, розміщеними на головній сторінці вебсайту, які мають стрілки та крапки для керування перемиканням слайдів. Слайди перемикаються елементами керування своїх батьківських контейнерів-слайдшоу.

Скрипт для показу прихованого контенту розроблявся для сторінок зі списками уроків. Уроки на цих сторінках згруповані за темами. За замовченням у кожній темі відображається по три блоки-посилання на уроки, а під ними розміщена кнопка “Показати більше”. При натисканні по ній відображаються інші приховані блоки-посилання на уроки, а текст на кнопці змінюється на “Показати менше”. Після повторного натискання кнопки додаткові блоки знов приховуються.

Слова та фрази в уроках, словнику та діалогах мають озвучення, яке відтворюється після клацання по іконці вушка, що їм передує. Для цього було створено функцію `playSound()`, яка отримує елемент з HTML-документа за ідентифікатором та викликає подію `play()`.

Тести, розміщені на кожній сторінці уроків, складаються з кількох питань, які мають лише один правильний варіант відповіді з чотирьох. Питання тестів перемішуються, тобто відображаються у випадковому порядку, при кожному оновленні сторінки. Після вибору варіанту відповіді не можна обрати іншу: кнопки з відповідями стають неактивними. Якщо обрано правильний варіант відповіді, то кнопка цієї відповіді буде зеленого кольору, а якщо обрано неправильний – червоного та буде виділено кнопку з правильним варіантом зеленим кольором. Перехід до наступного питання відбувається після натискання на кнопку з текстом “Далі”. Після проходження тесту з’являються рядок з кількістю правильних відповідей та кнопка “Пройти знову”. На рисунку 2.10 зображено тест в уроці в якому обрано неправильний варіант відповіді.

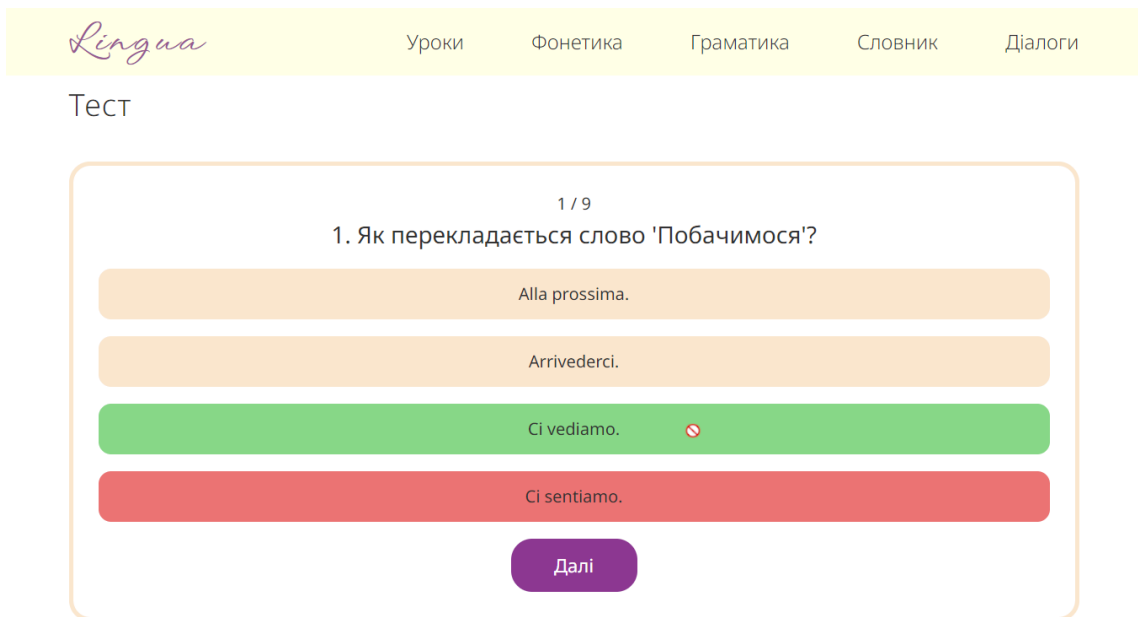


Рис. 2.10 Тест в уроці з обраним неправильним варіантом відповіді

Розроблено функцію пошуку перекладу слів для сторінки словника. За замовченням на сторінці окрім хедера та футера відображається лише назва сторінки та поле пошуку. Користувач починає вводити у поле пошуку слово українською або італійською і якщо буде виявлено збіг з буквосполученнями наявних на сторінці слів, вони відобразяться під полем пошуку. При натисканні на іконку `backspace`, яку розміщено в правому кутку поля пошуку, запит очищується.

Для адаптивної версії вебсайту було розроблено бургер-меню для роботи якого використовується JavaScript. Пункти меню розташовані блоком для пристроїв з шириною екрана менше 860px та приховані за межами екрана. При натисканні на іконку меню до списку меню додається клас `showNav`, який відповідає за показ меню, а сама іконка змінюється на хрестик. При клацанні по іконці хрестика меню знов приховується за межами екрана.

2.5 Розробка адаптивної версії вебсайту

Щоб вебсайт виглядав привабливо, тобто компоненти сторінок коректно відображалися і на комп'ютері, і на менших розмірах екранів, необхідно було розробити адаптивну версію. Для її розробки було використано медіазапити CSS.

На рисунку 2.11 зображено слайдшоу на головній сторінці вебсайту у планшетній версії. Якщо у комп'ютерній версії зображення розміщено в одну лінію зі слайдшоу, то при адаптуванні ці елементи розміщуються один під одним.



Рис. 2.11 Слайдшоу на головній сторінці у планшетній версії вебсайту

При зменшенні ширини екрана пункти меню вебсайту більше не могли розмішуватися в межах свого контейнера: вони виїжджали вбік та створювали горизонтальну прокрутку на сторінці. Для того, щоб виправити цю ситуацію було розроблено бургер-меню, яке зображено на рисунку 2.12.

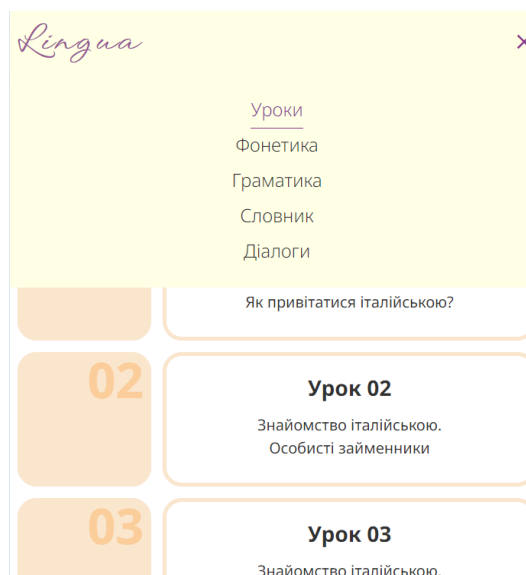


Рис. 2.12 Бургер-меню у планшетній версії вебсайту

Для того, щоб на менших розмірах екранів вебсайт не мав зайвого пустого місця, а текст не займав більшу частину екрана було зменшено відступи між елементами та розміри шрифтів. На рисунку 2.13 зображено сторінку словника для мобільної версії вебсайту на якій відображено результати пошуку перекладу слів, що мають буквсполучення *giorn*.

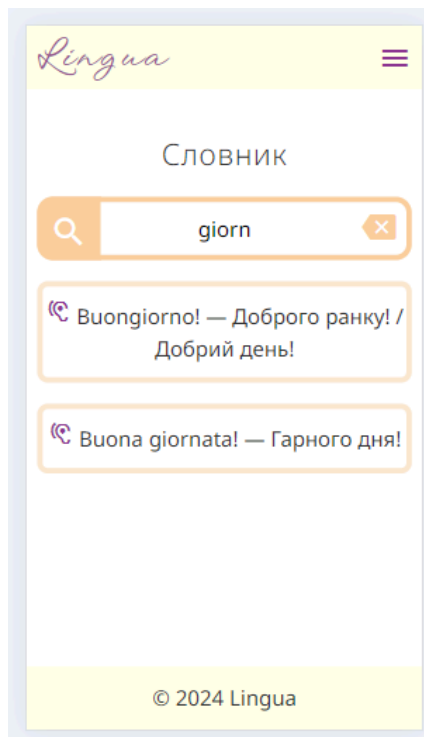


Рис. 2.13 Сторінка словника у мобільній версії вебсайту

РОЗДІЛ 3 ТЕСТУВАННЯ ВЕБСАЙТУ

Тестування є одним з важливих етапів життєвого циклу програмного забезпечення. Під час тестування системи відбувається виконання тестових випадків фактичний результат яких порівнюється з очікуваним. Тестування проводять з метою виявлення дефектів.

Система тестується на різних етапах свого життєвого циклу. Чим раніше буде виконано тестування, тим бюджетнішим вийде проєкт, адже дефекти будуть виявлені та усунені на ранніх стадіях розробки програмного забезпечення.

Існує два типи тестування: функціональне (перевірка коректності роботи функцій системи) та нефункціональне (продуктивності, безпеки, інтерфейсу користувача, зручності використання, інсталяційне, сумісності). Також виділяють різні методи тестування: “чорної скриньки”, “сірої скриньки” та “білої скриньки”.

Тестування “чорної скриньки” виконується без знань про внутрішню структуру коду програмного забезпечення. Принципом цього методу є оцінка продукту з погляду користувача.

Тестування “сірої скриньки” виконується з частковим знанням про внутрішню структуру системи. Метою цього методу є пошук дефектів, які з’являються через неправильне написання коду.

Тестування “білої скриньки” виконується з повним доступом до коду продукту. Метою цього методу є перевірка внутрішньої структури, будови та коду програмного забезпечення.

Тестування програмного забезпечення має чотири рівні:

- модульне тестування. Перевірка роботи окремих модулів системи;
- інтеграційне тестування. Перевірка взаємодії модулів між собою;
- системне тестування. Перевірка роботи повної системи;
- приймальне тестування. Перевірка програмного продукту на відповідність до вимог.

Основні види тестування вебсайтів:

- функціональне тестування. Перевірка роботи функцій вебсайту;
- тестування інтерфейсу користувача. Перевірка графічних елементів вебсайту (кольору тексту, розміру шрифту, кнопок, посилань, зображень та ін.) на відповідність до вимог проєкту;
- usability тестування. Перевірка зручності використання компонентів, перевірка доступності інформації;
- тестування продуктивності. Перевірка навантаженості яку може витримати вебсайт;
- тестування безпеки. Перевірка вебсайту на вразливість до атак.

3.1 Тестування інтерфейсу користувача вебсайту

Тестування графічного інтерфейсу користувача проводиться з метою перевірки правильності відображення компонентів, наприклад, зображення не повинно перекривати текст. Також перевіряється коректність взаємодії компонентів один з одним. При тестуванні інтерфейсу імітується поведінка користувача: натискання кнопок, перехід по посиланнях, введення даних у поля форми, ін.

Результати тестування графічного інтерфейсу користувача вебсайту надано у таблиці 3.1.

Таблиця 3.1

Результати тестування графічного інтерфейсу користувача вебсайту

№ тесту	Назва тесту	Очікуваний результат	Статус тесту
1	Перевірка хедеру сайту	У верхньому лівому кутку розташовано логотип. Пункти меню не виходять за рамки свого контейнера. При зменшенні ширини екрана пункти меню замінюються меню-бургером.	Пройдено

Продовження таблиці 3.1

Результати тестування графічного інтерфейсу користувача вебсайту

№ тесту	Назва тесту	Очікуваний результат	Статус тесту
2	Перевірка секцій зі слайдшоу на головній сторінці	Контент розміщений у слайдшоу не виходить за межі свого контейнера. При зменшенні ширини екрана розміщені лінійно елементи (блок з зображенням та блок зі слайдшоу) розміщуються один під одним.	Пройдено
3	Перевірка блоків посилань на уроки	Текст у блоках розміщено по центру, він не виходить за межі блоків. При зменшенні ширини екрана блоки стискаються (зменшують свою ширину).	Пройдено
4	Перевірка контенту на сторінці уроку	Контент сторінки не виходить за межі свого контейнера. При зменшенні ширини екрана розмір шрифту зменшується.	Пройдено
5	Перевірка поля пошуку на сторінці словника	Елементи пошукового рядка (іконка лупи та кнопка <code>backspace</code>) розміщені у його межах. При зменшенні ширини екрана зменшується розмір шрифту підказки та тексту пошукового рядка.	Пройдено
6	Перевірка елементів діалогу	Текст фраз діалогу не виходить за межі свого контейнера. При зменшенні ширини екрана розмір шрифту зменшується.	Пройдено
7	Перевірка читабельності тексту	Кольори тексту та фону на якому вони розміщені мають контрастність рівня AA, яка є рекомендованою.	Пройдено

За результатами тестування можна зробити висновок, що графічний інтерфейс користувача вебсайту є зручним: візуальні елементи розміщені правильно, тобто не перекривають один одного; при адаптуванні компоненти відображаються коректно, тобто розміщуються в межах своїх контейнерів.

Горизонтальна прокрутка на вебсайті не з'являється при зменшенні ширини екрана. Текст є читабельним на своєму фоні, адже має рекомендований рівень контрастності.

Також було перевірено наявність орфографічних помилок у тексті. Знайдені помилки було виправлено.

3.2 Тестування зручності використання вебсайту

Тестування зручності використання проводиться з метою оцінки зручності використання вебсайту. Під час usability тестування перевіряється зручність навігації, інтуїтивність інтерфейсу, тобто наскільки користувачу зрозуміло навіщо потрібен той чи інший компонент, а також зручність використання елементів керування, наприклад, кнопок. Під час тестування зручності використання також може проводитися тестування інтерфейсу користувача.

У таблиці 3.2 надано результати тестування зручності використання вебсайту.

Таблиця 3.2

Результати тестування зручності використання вебсайту

№ тесту	Назва тесту	Очікуваний результат	Статус тесту
1	Перевірка навігаційних посилань у хедері вебсайту	Логотип вебсайту є посиланням на головну сторінку; на головній сторінці не виконує роль посилання. Логотип розміщено на кожній сторінці. Посилання у меню ведуть на відповідні розділи. Доступ до меню можливий з будь-якої сторінки.	Пройдено
2	Перевірка посилань на уроки	При наведенні курсора на блоки, що виконують роль посилань вказівник змінюється зі стандартного на руку. Посилання доступні для натискання.	Пройдено

Продовження таблиці 3.2

Результати тестування зручності використання вебсайту

№ тесту	Назва тесту	Очікуваний результат	Статус тесту
3	Перевірка зручності використання кнопок	Кнопки є зручними для натискання: їх розмір достатній для того, щоб мати можливість клацнути по них курсором миші або натиснути пальцем по екрану.	Пройдено
4	Перевірка логічності використання кольорів	Інтерактивні елементи виділено іншим кольором (на вебсайті це фіолетовий). Правильна відповідь у тесті виділяється зеленим, неправильна – червоним.	Пройдено
5	Перевірка полів форми	Поле пошуку на сторінці словника має підказку “Введіть слово тут...”.	Пройдено
6	Перевірка зображень	Зображення високої якості та мають пропорційні розміри (відсутня “сплюсненість”). Зображення на головній сторінці підібрані в одному стилі.	Пройдено

За результатами тестування можна зробити висновок, що вебсайт є зручним для користувачів: інтерфейс інтуїтивно зрозумілий, навігація доступна на кожній сторінці, а елементи керування є доступними для натискання.

3.3 Функціональне тестування вебсайту

Функціональне тестування проводиться з метою перевірки правильності роботи функцій програмного забезпечення, тобто порівняння очікуваного результату, що вказаний у вимогах до продукту, роботи функцій з фактичним.

Крім тестування основних функцій вебсайтів також перевіряються переходи за посиланнями, заповнення форм, реєстрація/авторизація, додавання контенту користувачами, ін.

Результати тестування основних функцій вебсайту надано у таблиці 3.3.

Таблиця 3.3

Результати тестування основних функцій вебсайту

№ тесту	Назва тесту	Очікуваний результат	Фактичний результат
1	Перевірка роботи слайдшоу на головній сторінці вебсайту	Слайди змінюються після натискання елементів керування, які належать відповідним слайдшоу.	Слайди змінюються після натискання елементів керування, які належать відповідним слайдшоу.
2	Перевірка роботи кнопки “Показати більше” на сторінках зі списками уроків	Після клацання по кнопці “Показати більше” показуються додаткові блоки з посиланнями на уроки. Кожна кнопка працює для своєї теми уроків.	Після клацання по кнопці “Показати більше” показуються додаткові блоки з посиланнями на уроки. Кожна кнопка працює для своєї теми уроків.
3	Перевірка відтворення аудіо при кліці по іконці вушка на сторінках уроків	При натисканні іконки вушка на сторінках уроків відтворюється аудіо запис, що відповідає слову або фразі, яким передує.	При натисканні іконки вушка на сторінках уроків відтворюється аудіо запис, що відповідає слову або фразі, яким передує.
4	Перевірка роботи тесту в уроці	Після вибору варіанту відповіді у тесті зеленим кольором виділяється правильна відповідь, червоним – неправильна. Після проходження тесту відображається кількість правильних відповідей і загальна.	Після вибору варіанту відповіді у тесті зеленим кольором виділяється правильна відповідь, червоним – неправильна. Після проходження тесту відображається кількість правильних відповідей і загальна.
5	Перевірка роботи пошуку слів у словнику	При вводі літер у поле пошуку на сторінці з’являються слова, якщо виявлено збіг з наявними словами. При натисканні іконки <i>backspace</i> поле пошуку очищується.	При вводі літер у поле пошуку на сторінці з’являються слова, якщо виявлено збіг з наявними словами. При натисканні іконки <i>backspace</i> поле пошуку очищується.

За результатами тестування можна зробити висновок, що основні функції вебсайту працюють відповідно до очікуваного результату, отже, вебсайт готовий до використання.

ВИСНОВКИ

Усі задачі, які було поставлено під час написання кваліфікаційної роботи бакалавра, було виконано в повному обсязі.

1. Досліджено особливості процесу вивчення італійської мови. Визначено основні навички, розвиток яких важливий для вивчення італійської мови.

2. Проведено аналіз існуючих рішень для вивчення італійської мови. Визначено переваги та недоліки цих рішень.

3. На основі результатів аналізу особливостей процесу вивчення італійської мови сформульовано вимоги до вебсайту для вивчення італійської мови з урахуванням недоліків існуючих рішень.

4. Проаналізовано варіанти використання вебсайту для вивчення італійської мови. Спроектовано структуру вебсайту та створено його дизайн.

5. Розроблено вебсайт для вивчення італійської мови з використанням мови розмітки HTML, мови таблиць стилів CSS та мови програмування JavaScript. Розроблено адаптивну версію вебсайту. Ключовими відмінностями вебсайту від існуючих рішень є наявність уроків, що мають граматичні та фонетичні правила, а також словник з функцією пошуку перекладу слів.

6. Проведено тестування вебсайту для вивчення італійської мови.

Розроблений вебсайт можна використовувати для спрощення процесу вивчення італійської мови.

ПЕРЕЛІК ПОСИЛАНЬ

1. Мінцифри, USAID, КМІС. Оцінка рівня володіння іноземними мовами дорослого населення України. *Звіт кількісного соціологічного дослідження*. Грудень 2022 – січень 2023.
2. Beinan Zhou, Andrea Krott. Bilingualism enhances attentional control in non-verbal conflict tasks – evidence from ex-Gaussian analyses. *Cambridge University Press. Bilingualism: Language and Cognition*. January 2018. Vol. 21, No. 1. P. 162-180.
3. Self-study of Italian as a foreign language. *Università di Pavia*.
URL: https://cla.unipv.it/?page_id=53468
4. Zhenyu Yang. A Study on Self-efficacy and Its Role in Mobile-assisted Language Learning. *Theory and Practice in Language Studies*. April 2020. Vol. 10, No. 4. P. 439-444.
5. Зелінська В. А. Роль цифрових технологій у підвищенні ефективності вивчення іноземних мов. *Міжнародна науково-практична конференція «Актуальні питання інтернаціоналізації вищої освіти в Україні в умовах сучасних освітніх реалій, 21-22 березня 2024 р., Біла Церква, Білоцерківський національний аграрний університет*. Збірник тез. Біла Церква: БНАУ. С. 169.
6. Duolingo. URL: <https://duolingo.com/>
7. André Kenji Horie. Rewriting Duolingo's engine in Scala. 31.01.2017.
URL: <https://blog.duolingo.com/rewriting-duolingos-engine-in-scala/>
8. Mondly. URL: <https://app.mondly.com/>
9. Mondly FAQ. What technologies do we use in the main app?
URL: <https://www.mondly.com/faq/technologies-we-use-in-main-app>
10. Drops. URL: <https://app.languagedrops.com/>
11. Busuu. URL: <https://busuu.com/>

12. Celia Usero Navarro. A Story About Dynamic SVG Loading. 15.10.2019.
URL: <https://tech.busuu.com/a-story-about-dynamic-svg-loading-82e97c3230d7>
13. Memrise. URL: <https://www.memrise.com/>
14. uTalk. URL: <https://utalk.com/>
15. Wlingua. URL: <https://italian.wlingua.com/>
16. MDN. HTML: HyperText Markup Language.
URL: <https://developer.mozilla.org/en-US/docs/Web/HTML>
17. MDN. CSS: Cascading Style Sheets.
URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>
18. MDN. JavaScript.
URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

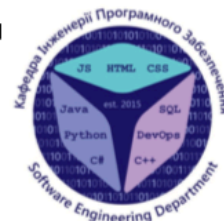
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка Web-сайту для вивчення італійської мови з використанням HTML, CSS, JS

Виконала студентка 5 курсу
групи ППЗ-51
Осадча Анастасія Андріївна
Керівник роботи

Доктор філософії, доцент кафедри ІПЗ Гребенюк Віктор Вікторович

Київ – 2024

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – спрощення процесу вивчення італійської мови за рахунок використання вебсайту, розробленого за допомогою HTML, CSS, JS.
- **Об'єкт дослідження** – процес вивчення італійської мови.
- **Предмет дослідження** – програмне забезпечення для вивчення італійської мови.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Дослідити особливості процесу вивчення італійської мови.
2. Проаналізувати існуючі рішення для вивчення італійської мови та визначити переваги і недоліки цих рішень.
3. Провести аналіз програмних засобів та вибрати найкращі для реалізації вебсайту для вивчення італійської мови.
4. Сформулювати вимоги до вебсайту для вивчення італійської мови з урахуванням недоліків існуючих рішень.
5. Спроектувати та розробити вебсайт для вивчення італійської мови.
6. Провести тестування вебсайту.

3

АНАЛІЗ АНАЛОГІВ

Функціонал	Duolingo	Mondly	Drops	Busuu	Memrise	uTalk	Wlingua	Lingua
Платформи	Android, iOS, Web	Android, iOS, Web	Android, iOS, Web	Android, iOS, Web	Android, iOS, Web	Android, iOS	Android, iOS, Web	Web
Мова курсу італійської	англійська	українська англійська	українська англійська	англійська	англійська	українська англійська	англійська	українська
Нова лексика	+	+	+	+	+	+	+	+
Граматичні правила	+	-	-	+	-	-	+	+
Фонетичні правила	-	-	-	+	-	-	-	+
Тренування сприйняття на слух	+	+	+	+	+	+	+	+
Тренування розуміння мови у контексті	+	+	+	+	-	-	+	+
Словник з функцією пошуку	-	-	+	-	-	-	-	+

4

ВИМОГИ ДО ВЕБСАЙТУ

Функціональні вимоги:

1. Вивід результатів тестів одразу після проходження.
2. Слова та фрази озвучуються після натискання по іконці вушка.
3. Функція пошуку перекладу слів у словнику.

Нефункціональні вимоги:

1. Можливість переглядати вебсайт на різних пристроях, що мають веббраузер: комп'ютерах, планшетних ПК або смартфонах.
2. Українська локалізація інтерфейсу вебсайту.
3. Наявність уроків, що мають теорію та тести.
4. Наявність уроків, що мають граматичні правила.
5. Наявність уроків, що мають фонетичні правила.
6. Наявність діалогів, що забезпечують розуміння мови в контексті.
7. Можливість переглядати переклад фраз у діалогах.

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Visual Studio Code



Figma

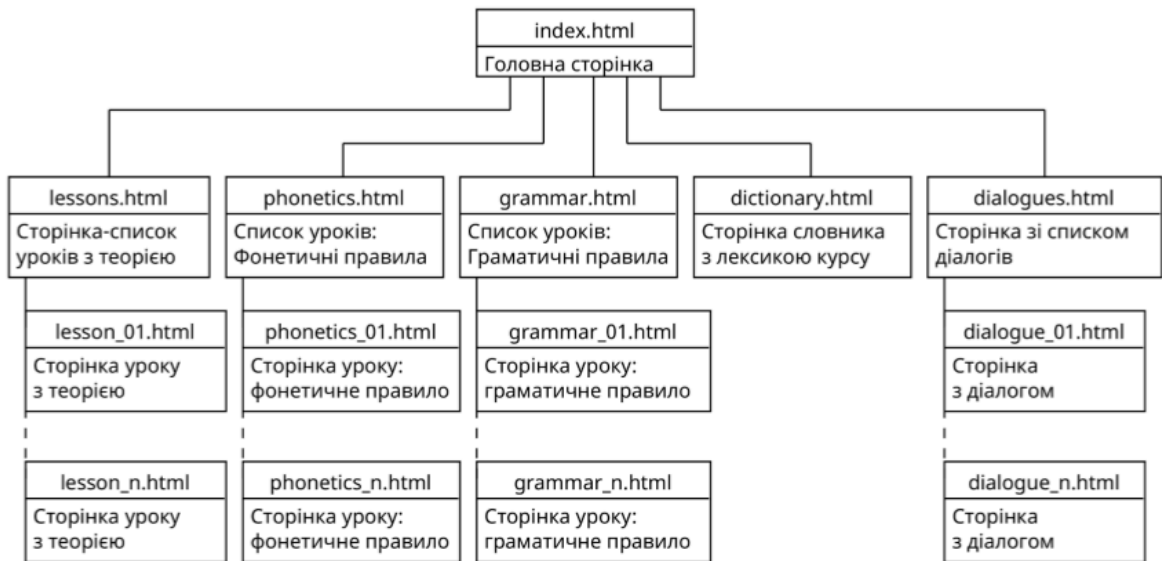
6

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



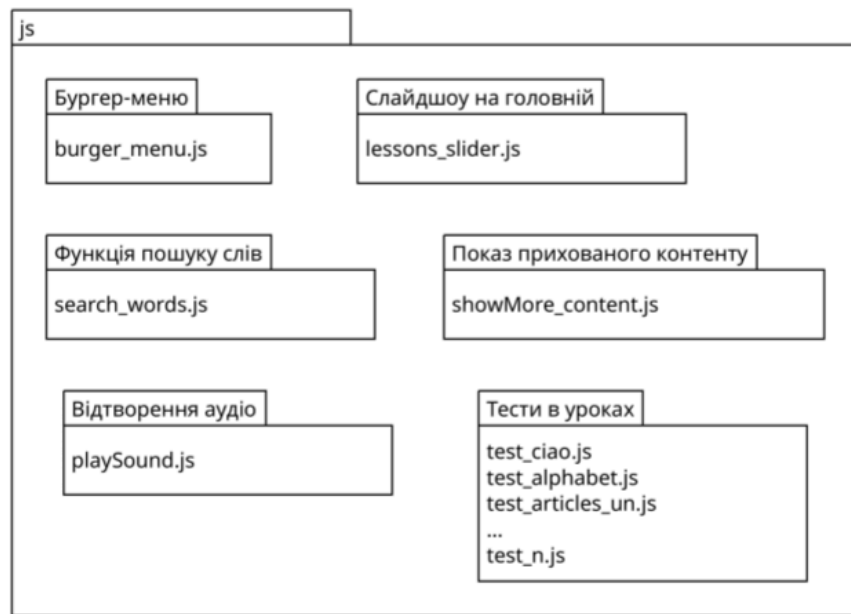
7

СХЕМА СТРУКТУРИ ВЕБСАЙТУ



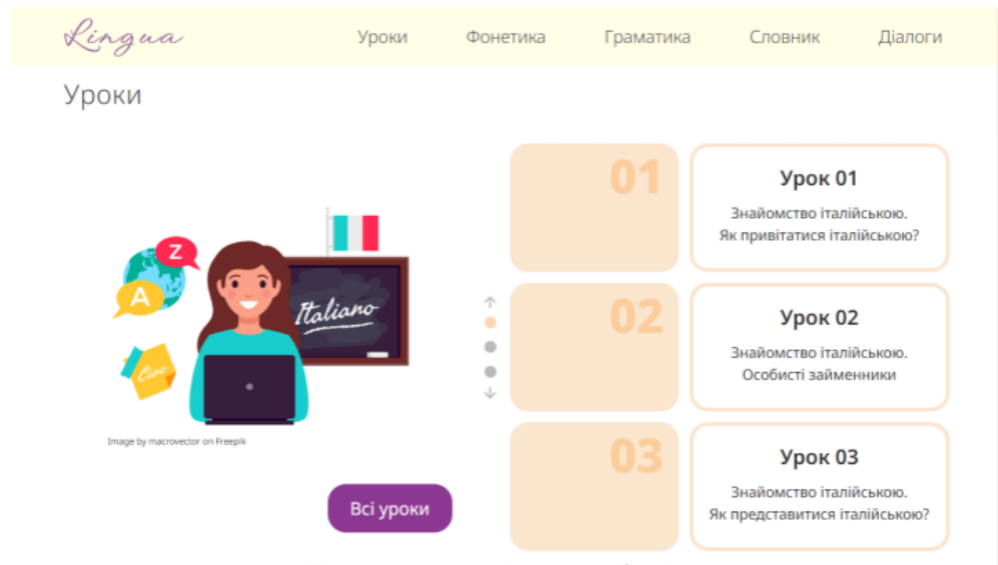
8

ДІАГРАМА ПАКЕТІВ СКРИПТІВ ВЕБСАЙТУ



9

ЕКРАННІ ФОРМИ



Головна сторінка вебсайту

10

ЕКРАННІ ФОРМИ

The screenshot shows the 'Lingua' website interface. At the top, there is a navigation bar with the logo 'Lingua' and menu items: 'Уроки', 'Фонетика', 'Граматика', 'Словник', and 'Діалоги'. The main content area is titled 'Привітання та прощання' and contains the following text:

- Buona giornata! — Гарного дня!
- Buona serata! — Гарного вечора!
- Buonanotte! — Добраніч!

В італійській мові форми привітань та прощань залежать від декількох факторів: часу доби, ситуації та людей з якими вітаються або прощаються. Також розрізняють дві форми привітань та прощань: формальні та неформальні.

Неформальні форми використовують, коли вітаються або прощаються з членами родини, дітьми, друзями та знайомими. Формальні форми — з людьми старшими, незнайомими або малознайомими.

Неформальні форми привітань та прощань

- Ciao! — Привіт! / Бувай!

Це слово використовується для вітання та прощання у неформальних ситуаціях.

Для прощання у неформальних ситуаціях можна використати наступні фрази:

- A presto! — Побачимося незабаром!
- A domani! — Побачимося завтра!
- Ci vediamo! — Побачимося!
- Ci sentiamo! — На зв'язку!
- Alla prossima! — До наступної зустрічі!

Фрагмент сторінки уроку

11

ЕКРАННІ ФОРМИ

The screenshot shows a test page on the 'Lingua' website. The navigation bar is the same as in the previous screenshot. The page is titled 'Тест' and shows a question: '1. Як перекладається слово 'Побачимося'?'. There are four options:

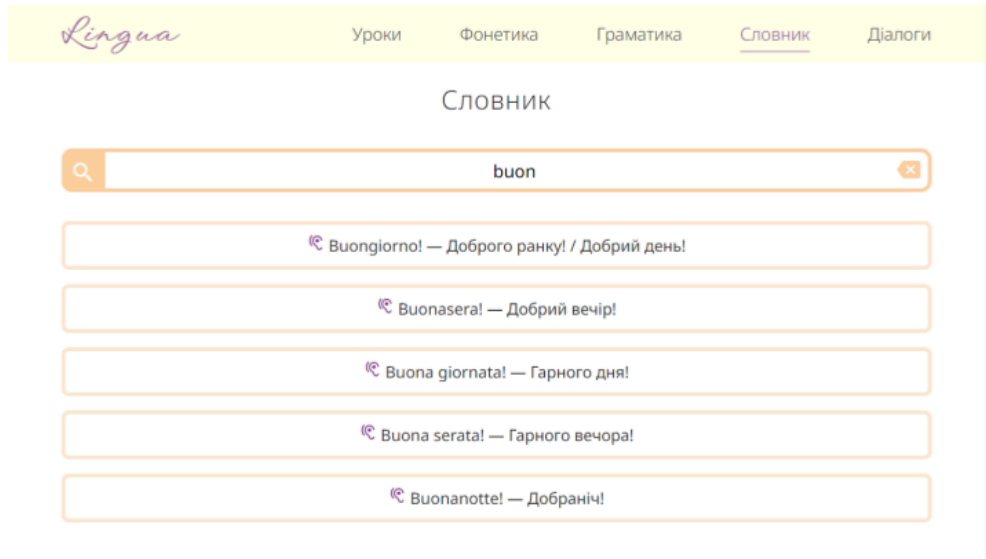
- Alla prossima.
- Arrivederci.
- Ci vediamo. (This option is highlighted in green, indicating it is the correct answer.)
- Ci sentiamo.

At the bottom of the question box, there is a purple button labeled 'Далі'.

Тест на сторінці уроку

12

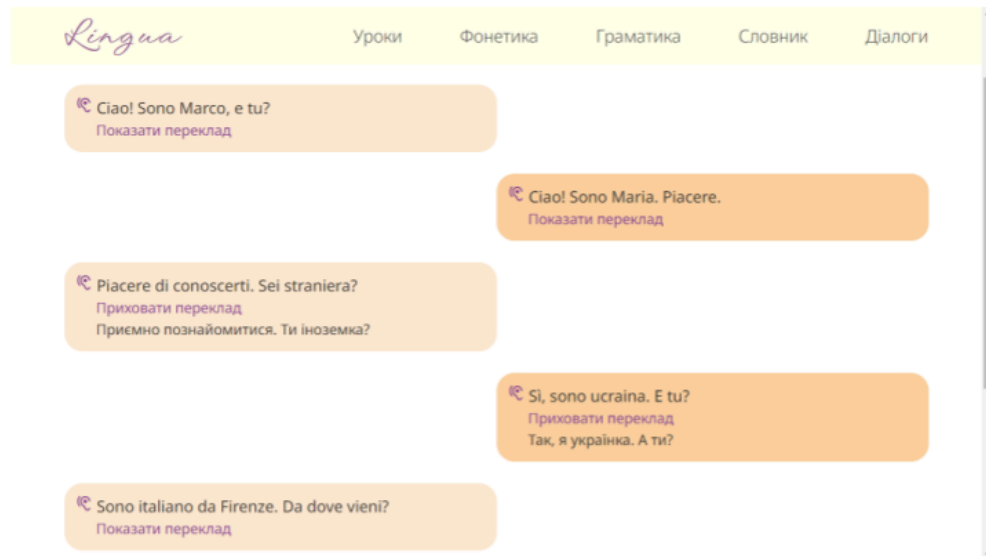
ЕКРАННІ ФОРМИ



Сторінка словника

13

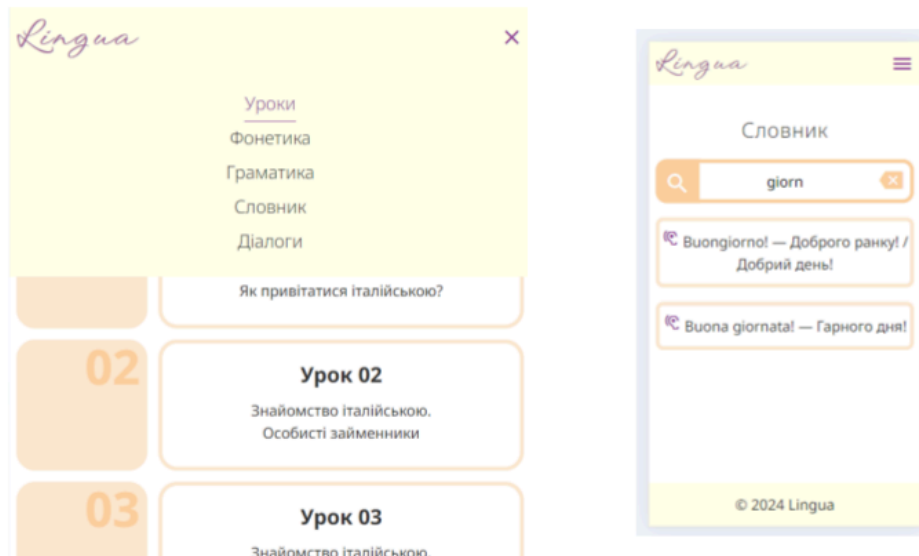
ЕКРАННІ ФОРМИ



Сторінка діалогу

14

ЕКРАННІ ФОРМИ



Адаптивна версія

15

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Осадча А.А. Аналіз існуючих рішень для вивчення італійської мови / А. А. Осадча, В.В. Залива // Застосування програмного забезпечення в інформаційно-комунікаційних технологіях: Матеріали всеукраїнської науково-технічної конференції. Збірник тез. 24.04.2024, ДУІКТ, м. Київ. К: ДУІКТ, 2024. С. 183.

2. Осадча А.А. Визначення вимог до вебсайту для вивчення італійської мови / А.А. Осадча, В.В. Залива // Застосування програмного забезпечення в інформаційно-комунікаційних технологіях: Матеріали всеукраїнської науково-технічної конференції. Збірник тез. 24.04.2024, ДУІКТ, м. Київ. К: ДУІКТ, 2024. С. 185.

16

ВИСНОВКИ

1. Досліджено особливості процесу вивчення італійської мови.
2. Проведено аналіз існуючих рішень для вивчення італійської мови та визначено переваги і недоліки цих рішень.
3. Проведено аналіз засобів розробки програмного забезпечення. Обрано HTML, CSS та мову програмування JavaScript.
4. Сформульовано вимоги до вебсайту для вивчення італійської мови з урахуванням недоліків існуючих рішень.
5. Спроектовано та розроблено вебсайт для вивчення італійської мови. Ключовими відмінностями вебсайту від існуючих рішень є наявність уроків, що мають граматичні та фонетичні правила, а також словник з функцією пошуку перекладу слів.
6. Проведено тестування вебсайту.

ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

lessons_slider.js

```
let slideIndex = 1;
let z = document.querySelector(".lessons_slider");

for (i = 0; i < z.length; i++) {
  z[i].setAttribute("data-currentSlide", 1);
  showDivs(z[i].getAttribute("data-currentSlide"), i);
}

function plusDivs(n, j) {
  slideIndex =
  parseInt(z[j].getAttribute("data-currentslide")[0]);
  showDivs(slideIndex += n, j);
}

function currentDiv(n, j) {
  showDivs(slideIndex = n, j);
}

function showDivs(n, j) {
  let i;
  let z = document.querySelector(".lessons_slider")[j];
  let slides = z.querySelector(".lessons_slide");
  let dots = z.querySelector(".lessons_dot");

  if (n > slides.length) {
    slideIndex = 1;
  }

  if (n < 1) {
    slideIndex = slides.length;
  }

  z.setAttribute("data-currentSlide", slideIndex);

  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }

  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace("
active", "");
  }

  slides[slideIndex - 1].style.display = "block";
  dots[slideIndex - 1].className += " active";
}
```

showMore_content.js

```
function showMore(button) {
  let content =
  button.parentElement.querySelector(".more-content");
  if (content.style.display === "none") {
    content.style.display = "block";
    button.innerHTML = "Показати менше";
  }
}
```

```
else {
  content.style.display = "none";
  button.innerHTML = "Показати більше";
}
}
```

```
const moreConent =
document.querySelectorAll(".more-content");
for (let i = 0; i < moreConent.length; i++) {
  moreConent[i].style.display = "none";
}
}
```

burger_menu.js

```
const nav = document.querySelector(".nav");
const navItems =
document.querySelectorAll(".nav_item");
const hamburger=
document.querySelector(".hamburger");
const closeIcon= document.querySelector(".close-icon");
const menuIcon =
document.querySelector(".menu-icon");
```

```
function toggleMenu() {
  if (nav.classList.contains("showNav")) {
    nav.classList.remove("showNav");
    closeIcon.style.display = "none";
    menuIcon.style.display = "block";
  }
```

```
else {
  nav.classList.add("showNav");
  closeIcon.style.display = "block";
  menuIcon.style.display = "none";
}
}
```

```
hamburger.addEventListener("click", toggleMenu);
```

```
navItems.forEach(
  function(navItem) {
    navItem.addEventListener("click", toggleMenu);
  }
)
```

search_words.js

```
document.querySelector('#search-input').addEventListener(
  'input', filterList);
```

```
function filterList() {
  const searchInput =
  document.querySelector('#search-input');
  const filter = searchInput.value.toLowerCase();
  const listItems = document.querySelectorAll('.word');
```

```
listItems.forEach((item) => {
  let text = item.textContent;
```

```

if(text.toLowerCase().includes(filter.toLowerCase())){
    item.style.display = 'block';
}

else {
    item.style.display = 'none';
}
});
}
}

```

test_ciao.js

```

const questions = [
    {
        question: "Як перекладається фраза 'A presto'?",
        answers: [
            { text: "Привіт.", correct: false},
            { text: "Добраніч.", correct: false},
            { text: "Побачимося незабаром.", correct: true},
            { text: "Побачимося завтра.", correct: false}
        ]
    },
    {
        question: "Як перекладається фраза 'A domani'?",
        answers: [
            { text: "Привіт.", correct: false},
            { text: "Добраніч.", correct: false},
            { text: "Побачимося незабаром.", correct:
false},
            { text: "Побачимося завтра.", correct: true}
        ]
    },
    {
        question: "Як перекладається слово
'Buonanotte'?",
        answers: [
            { text: "Привіт.", correct: false},
            { text: "Добраніч.", correct: true},
            { text: "Побачимося незабаром.", correct:
false},
            { text: "Побачимося завтра.", correct: false}
        ]
    },
    {
        question: "Як перекладається фраза 'На
зв'язку'?",
        answers: [
            { text: "Alla prossima.", correct: false},
            { text: "Arrivederci.", correct: false},
            { text: "Ci vediamo.", correct: false},
            { text: "Ci sentiamo.", correct: true}
        ]
    },
    {
        question: "Як перекладається слово
'Побачимося'?",
        answers: [
            { text: "Alla prossima.", correct: false},
            { text: "Arrivederci.", correct: false},
            { text: "Ci vediamo.", correct: true},
            { text: "Ci sentiamo.", correct: false}
        ]
    },
}

```

```

{
    question: "Як перекладається фраза 'Добрий
вечір'?",
    answers: [
        { text: "Buonasera.", correct: true},
        { text: "Buona giornata.", correct: false},
        { text: "Buona serata.", correct: false},
        { text: "Buongiorno.", correct: false}
    ]
},
{
    question: "Яким зі слів можна і привітатися, і
попрощатися?",
    answers: [
        { text: "Arrivederci.", correct: false},
        { text: "A presto.", correct: false},
        { text: "Ciao.", correct: true},
        { text: "Ci vediamo.", correct: false}
    ]
},
{
    question: "Яка з прощань належить до
формальної форми?",
    answers: [
        { text: "Buonanotte.", correct: false},
        { text: "Buona serata.", correct: true},
        { text: "A presto.", correct: false},
        { text: "Alla prossima.", correct: false}
    ]
},
{
    question: "Яка з фраз використовується для
вітання і прощання з малознайомими людьми?",
    answers: [
        { text: "Ciao.", correct: false},
        { text: "A domani.", correct: false},
        { text: "Buonasera.", correct: false},
        { text: "Salve.", correct: true}
    ]
}
]

```

```

const questionElement =
document.getElementById('question');
const answerButtons =
document.getElementById('answer-btns');
const nextButton =
document.getElementById('next-btn');
const testDetailsElement =
document.querySelector(".test_details");

```

```

let currentQuestionIndex = 0;
let score = 0;
let shuffledQuestions;

```

```

function startTest() {
    shuffledQuestions = questions.sort(() =>
Math.random() - .5);
    currentQuestionIndex = 0;
    score = 0;
    nextButton.innerHTML = "Далі";
}

```



```

    testDetailsElement.innerHTML =
`<p>${currentQuestionIndex} /
${questions.length}</p>`;
    testDetailsElement.style.display = "block";
    setNextQuestion();
}

function setNextQuestion() {
    resetState();

    showQuestion(shuffledQuestions[currentQuestionIndex])
;
}

function showQuestion() {
    resetState();
    let currentQuestion =
questions[currentQuestionIndex];
    let questionNo = currentQuestionIndex + 1;
    questionElement.innerHTML = questionNo + ". " +
currentQuestion.question;
    testDetailsElement.innerHTML = `<p>${questionNo}
/ ${questions.length}</p>`;

    currentQuestion.answers.forEach(answer => {
        const button = document.createElement("button");
        button.innerHTML = answer.text;
        button.classList.add("btn");
        answerButtons.appendChild(button);
        if (answer.correct) {
            button.dataset.correct = answer.correct;
        }
        button.addEventListener("click", selectAnswer);
    });
}

function resetState() {
    nextButton.style.display = "none";
    while(answerButtons.firstChild) {

answerButtons.removeChild(answerButtons.firstChild);
}

function selectAnswer(e) {
    const selectedButton = e.target;
    const isCorrect = selectedButton.dataset.correct ===
"true";
    if (isCorrect) {
        selectedButton.classList.add("correct");
        score++;
    } else {
        selectedButton.classList.add("incorrect");
    }
    Array.from(answerButtons.children).forEach(button
=> {
        if (button.dataset.correct === "true") {
            button.classList.add("correct");
        }
        button.disabled = true;
    });
    nextButton.style.display = "block";
}

```

```

function showScore() {
    resetState();
    questionElement.innerHTML = `Ви набрали ${score}
з ${questions.length}!`;
    nextButton.innerHTML = "Пройти znovu";
    nextButton.style.display = "block";
    testDetailsElement.style.display = "none";
}

function handleNextButton() {
    currentQuestionIndex++;
    if (currentQuestionIndex < questions.length) {
        showQuestion();
    } else {
        showScore();
    }
}

nextButton.addEventListener("click", ()=> {
    if (currentQuestionIndex < questions.length) {
        handleNextButton();
    } else {
        startTest();
    }
});

startTest();

```