

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Бакалаврська кваліфікаційна робота

на тему: «Розробка вебзастосунку для вивчення абетки Морзе на основі
адаптивного персоналізованого підходу»

Виконав: студент 4 курсу
групи ІПІ-216
спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Кременчук А.І.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Ракитянська Г.Б.

(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. ЗІ Гарнага В.А.

(прізвище та ініціали)

Допущено до захисту

Завідувач кафедри ПЗ

д.т.н., проф. Романюк О.Н.

«09» 09 2025 р.

ВНТУ – 2025

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший бакалаврський
Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Освітньо-професійна програма – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О.Н.
«21» березня 2025 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Кременчук Аліні Ігорівні

1. Тема роботи – «Розробка вебзастосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу»

Керівник роботи: Ракитянська Ганна Борисівна., к.т.н., доц. кафедри ПЗ, затверджені наказом вищого навчального закладу від 20 березня 2025 р. №97.

2. Строк подання студентом роботи 2 червня 2025 р.

3. Вихідні дані до роботи: тип програми – вебзастосунок; модель розробки – інкрементна; засоби організації даних програми – фреймворки React та Node.js; вхідні дані: персональні дані користувача, інформація про знання абетки Морзе та досвід роботи з нею, цілі навчання; вихідні дані: звуки абетки Морзе, навчальні вправи, імерсивні сценарії, статистика прогресу вивчення абетки Морзе; середовище розробки – Visual Studio Code; мова програмування – TypeScript.

4. Зміст розрахунково-пояснювальної записки: вступ; аналіз стану розвитку застосунку для вивчення абетки Морзе; розробка структури та алгоритмів програмного застосунку; розробка застосунку для вивчення абетки Морзе; тестування програмного забезпечення; висновки; список використаних джерел; додатки.

5. Перелік графічного матеріалу: титульний слайд; актуальність теми; мета, об'єкт та предмет дослідження; задачі дослідження, наукова новизна, практична цінність одержаних результатів; порівняльний аналіз аналогів, розробка методу когнітивно-адаптивної прогресії з аналізом патернів помилок; розробка методу навчання з імерсивними контекстними сценаріями; тестування застосунку; висновки; апробація та публікації.

6. Консультанти розділів роботи

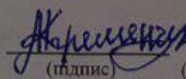
Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Ракитянська Г.Б., к.т.н., доцент кафедри ПЗ	24.03.25	01.06.25

7. Дата видачі завдання 24 березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

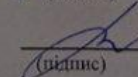
№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз стану розвитку застосунку для вивчення абетки Морзе	25.03.25 – 04.04.25	Вик.
2	Розробка структури та алгоритмів програмного застосунку	05.04.25 – 18.04.25	Вик.
3	Варіантний аналіз та вибір мови програмування розробки	19.04.25 – 22.04.25	Вик.
4	Розробка вебзастосунку для вивчення абетки Морзе	23.04.25 – 17.05.25	Вик.
5	Тестування програмного забезпечення	18.05.25 – 25.05.25	Вик.
6	Оформлення матеріалів до захисту БКР	26.05.25 – 30.05.25	Вик.

Студент



Кременчук А.І.
(прізвище та ініціали)

Керівник бакалаврської кваліфікаційної роботи



Ракитянська Г.Б.
(прізвище та ініціали)

АНОТАЦІЯ

УДК 004:005.9

Кременчук А.І. Розробка вебзастосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу : бакалаврська кваліфікаційна робота зі спеціальності 121 Інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця : ВНТУ, 2025. 95 с.

У бакалаврській кваліфікаційній роботі було проведено аналіз сучасного стану застосунків для вивчення абетки Морзе, доведено актуальність розробки такого застосунку. Проведено порівняльний аналіз аналогів з розроблюваним вебзастосунком, продемонстровано його переваги та таким чином доведено доцільність розробки.

Обґрунтовано вибір мови програмування JavaScript та відповідних технологій для реалізації вебзастосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу.

Отримані в бакалаврській кваліфікаційній роботі результати можна використовувати для вивчення абетки Морзе.

Ключові слова: абетка Морзе, персоналізація, навчання.

ABSTRACT

UDC 004:005.9

Kremenчук А.І. Development of a web application for learning Morse code based on an adaptive personalized approach: bachelor's qualification work in the specialty 121 Software Engineering, educational program - Software Engineering. Vinnytsia: VNTU, 2025. 95 p.

The bachelor's thesis analyzes the current state of applications for learning Morse code and proves the relevance of developing such an application. A comparative analysis of analogs with the web application under development was carried out, its advantages were demonstrated, and thus the feasibility of development was proved.

The choice of the JavaScript programming language and appropriate technologies for the implementation of a web application for learning Morse code based on an adaptive personalized approach is substantiated.

The results obtained in the bachelor's qualification work can be used to learn Morse code.

Keywords: Morse code, personalization, learning.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ СТАНУ РОЗВИТКУ ЗАСТОСУНКІВ ДЛЯ ВИВЧЕННЯ АБЕТКИ МОРЗЕ	6
1.1 Аналіз стану питання розробки застосунків для вивчення абетки Морзе	6
1.2 Порівняльний аналіз аналогів.....	7
1.3 Аналіз методів розв’язання задачі	13
1.4 Постановка задач дослідження	15
1.5 Висновки	16
2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ЗАСТОСУНКУ	17
2.1 Аналіз інформаційного забезпечення.....	17
2.2 Розробка архітектури застосунку	19
2.3 Розробка методу когнітивно-адаптивної прогресії з аналізом патернів помилок.....	21
2.4 Розробка методу навчання з імерсивними контекстними сценаріями	23
2.5 Висновки	26
3 РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АБЕТКИ МОРЗЕ	27
3.1 Варіантний аналіз та вибір мови програмування розробки	27
3.2 Розробка бази даних	29
3.3 Розробка модулів системи	36
3.4 Висновки	45
4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	46
4.1 Аналіз методів тестування	46
4.2 Тестування вебзастосунку для вивчення абетки Морзе	47
4.3 Висновки	60
ВИСНОВКИ	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТКИ.....	64
Додаток А – Технічне завдання	65
Додаток Б – Протокол перевірки на наявність текстових запозичень	68
Додаток В – Лістинг програми	69
Додаток Г – Ілюстративна частина.....	90

ВСТУП

Обґрунтування вибору теми дослідження. Абетка Морзе, хоч і вважається застарілою для масового використання, досі відіграє важливу роль у низці специфічних сфер. Серед радіоаматорів (операторів любительського радіо) Морзе залишається популярним способом обміну повідомленнями. Багато радіоаматорів продовжують передавати повідомлення абеткою Морзе на швидкості 15–20 слів за хвилину, навіть попри те, що знання Морзе більше не є обов'язковим для отримання ліцензії. Характерний ритм кодів дозволяє таким сигналам пробиватися крізь перешкоди і шум ефіру краще, ніж голосові передачі.

Завдяки цьому з допомогою простого передавача малої потужності (на кілька ват) і дротяної антени оператори Морзе можуть встановлювати зв'язок на відстанях у тисячі кілометрів.

У сучасній авіації цифрові канали зв'язку та голосові радіосистеми майже повністю витіснили абетку Морзе з повсякденного використання. Втім, Морзе все ще використовується для ідентифікації навігаційних радіомаяків: наземні VOR/NDB маяки передають свої індикатори у вигляді звукових сигналів Морзе, що пілоти прослуховують для підтвердження правильності налаштування [1].

Навіть якщо більшість пілотів не знають Морзе досконало, навчальні програми все ж ознайомлюють їх з базовими сигналами. Деякі льотні школи досі включають абетку Морзе до навчального курсу як засіб останньої надії: у випадку відмови радіосистем пілот зможе подати сигнал SOS блиманням бортових вогнів чи ручним радіомаяком. У ряді країн (наприклад, Бразилія, Японія) навіть існують екзаменаційні вимоги на вміння розпізнавати основні сигнали Морзе для отримання авіаційних сертифікатів. Це свідчить, що авіаційна спільнота розглядає знання Морзе як корисний навик у непередбачених ситуаціях.

Морзе розглядається військовими як резервний канал зв'язку на випадок, якщо високотехнологічні системи будуть недоступні. Шифровані цифрові канали можуть зазнати збоїв чи глушіння, тоді як передача повідомлень абеткою Морзе потребує мінімального обладнання і може працювати в умовах, де інші засоби

неефективні. У спеціальних операціях навички Морзе можуть застосовуватися для прихованого зв'язку, адже вузькосмуговий сигнал Морзе важче виявити чи заглушити противнику. Таким чином, навіть у XXI столітті абетка Морзе залишається в арсеналі військових як запасний, але надійний спосіб комунікації.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Метою роботи є покращення процесу вивчення абетки Морзе шляхом розробки спеціалізованого програмного забезпечення, що використовує адаптивний персоналізований підхід.

Основними задачами дослідження є:

- розробити архітектуру застосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу;
- розробити інтерфейс користувача застосунку;
- розробити метод персоналізованого підходу;
- розробити алгоритми роботи застосунку;
- розробити функціонал застосунку;
- провести тестування розробленого застосунку.

Об'єкт дослідження – процес розробки застосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу.

Предмет дослідження – методи та засоби розробки застосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу.

Методи дослідження. У процесі досліджень використовувались:

- методи реалізації вебзастосунків;
- методи реалізації архітектури застосунків;
- методи персоналізованого підходу до навчання;
- методи реалізації інтерфейсу;
- методи тестування застосунків.

Наукова новизна отриманих результатів.

1. Подальшого розвитку отримав метод когнітивно-адаптивної прогресії з

аналізом патернів помилок, який, на відміну від відомих, використовує "матриці помилок" для детального відстеження специфічних помилок користувача, автоматично коригує частоту появи проблемних символів у навчальних вправах, адаптує швидкість передачі для різних груп символів на основі індивідуальних труднощів кожного користувача, та застосовує прогресивне розширення набору символів лише після досягнення 90% точності розпізнавання поточних символів.

2. Подальшого розвитку отримав метод навчання з імерсивними контекстними сценаріями, який, на відміну від відомих, занурює користувача у персоналізовані наративні контексти (морські подорожі, космічні місії, історії порятунку), де використання абетки Морзе має практичне сюжетне значення, застосовує ритмічно-музичну синхронізацію для перетворення абстрактних кодів Морзе.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає в можливості використання розробленої системи для вивчення абетки Морзе та регулярної перевірки рівня знань.

Особистий внесок здобувача. Усі наукові результати, викладені у бакалаврській кваліфікаційній роботі, отримані автором особисто. У роботі, виконаній у співавторстві, автору належить: актуальність розробки вебзастосунку для вивчення абетки Морзе, особливості реалізації вебзастосунку.

Апробація результатів роботи. Описані у бакалаврській кваліфікаційній роботі положення доповідались на конференції «LIV Всеукраїнська науково-технічна конференція підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2025)».

Публікації. Результати роботи були опубліковані в матеріалах конференції «LIV Всеукраїнська науково-технічна конференція підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2025)» [2].

Аналіз. У пояснювальній записці до бакалаврської кваліфікаційної роботи було розглянуто 4 розділи та було використано 22 літературних джерела.

1 АНАЛІЗ СТАНУ РОЗВИТКУ ЗАСТОСУНКІВ ДЛЯ ВИВЧЕННЯ АБЕТКИ МОРЗЕ

1.1 Аналіз стану питання розробки застосунків для вивчення абетки Морзе

Розробка вебзастосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу – це перспективне завдання в області електронної освіти, що об'єднує сучасні веб-технології та методики адаптивного навчання. Такий застосунок має потенціал не лише полегшити процес засвоєння нових знань, а й зробити його максимально зручним та ефективним завдяки персоналізованим навчальним траєкторіям, що враховують індивідуальні особливості кожного користувача [3].

На сучасному етапі існують різні додатки для вивчення абетки Морзе, проте більшість з них орієнтовані на стандартні навчальні плани та статичний контент. Застосування адаптивного підходу дозволяє аналізувати показники користувача, такі як швидкість реагування, точність розпізнавання сигналів та частоту помилок, і відповідно регулювати складність завдань. Така методика широко використовується в інших сферах освіти, наприклад, у мовних додатках чи платформах для підготовки до іспитів, і її впровадження у вивчення абетки Морзе відкриває нові можливості для персоналізації процесу навчання [4].

Сучасні вебтехнології, такі як HTML5, CSS3 та сучасні JavaScript-фреймворки (React, Angular, Vue), дозволяють створити інтуїтивно зрозумілий та адаптивний інтерфейс, що працює на різних пристроях – від ПК до смартфонів і планшетів [5]. Інтеграція аудіо- та відеоелементів забезпечує можливість відтворення звукових сигналів абетки Морзе з високою точністю, що є критично важливим для правильного засвоєння аудиторного аспекту цього коду. Завдяки можливості інтеграції з хмарними сервісами, можна ефективно збирати та аналізувати дані про успішність користувачів, що дозволяє в режимі реального часу адаптувати навчальний контент [6].

Адаптивний персоналізований підхід вимагає використання алгоритмів, які

аналізують дані про поведінку користувача та формують індивідуальний навчальний маршрут. Це можуть бути методи машинного навчання, статистичний аналіз або навіть елементи методики просторового повторення (spaced repetition). На основі зібраної інформації система може визначати, які символи чи комбінації користувач засвоює повільніше, і пропонувати додаткові вправи для їх закріплення. Такий підхід дозволяє не лише прискорити процес навчання, але й підвищити рівень засвоєння матеріалу за рахунок постійного моніторингу та адаптації контенту до потреб конкретного користувача [7].

Одним із викликів розробки є забезпечення високої точності та надійності аудіо-відтворення сигналів, оскільки неправильне відтворення може негативно вплинути на процес навчання [8]. Крім того, необхідно ретельно продумати алгоритми адаптації, щоб вони не були занадто агресивними чи, навпаки, надто пасивними, що може призвести до втрати мотивації користувача. Інтеграція з різними платформами та забезпечення високої продуктивності в умовах великого потоку даних – це ще один аспект, який потребує особливої уваги.

Використання сучасних вебтехнологій у поєднанні з алгоритмами машинного навчання дозволяє створити інструмент, що не лише адаптується до рівня знань кожного користувача, а й сприяє ефективному засвоєнню матеріалу [9]. Ретельно спроектована система може стати корисним інструментом як для початківців, так і для тих, хто бажає вдосконалити свої навички, забезпечуючи індивідуальний підхід до кожного етапу навчання.

Отже актуальною є розробка вебзастосунку для вивчення абетки Морзе.

1.2 Порівняльний аналіз аналогів

На сьогодні існує чимало ресурсів, призначених для самостійного вивчення абетки Морзе. Це і веб-сайти, і мобільні додатки, і класичні програмні тренажери. Розглянемо деякі з них: Morse-It, Morse Mania, Morse Code Flash Cards.

Morse-It [10] – це універсальний застосунок для роботи з абеткою Морзе, який дозволяє зручно конвертувати текст у сигнали Морзе та навпаки. Завдяки інтуїтивно зрозумілому інтерфейсу, додаток підходить як для початківців, так і для

досвідчених користувачів. Він надає можливість налаштувати швидкість передачі, а також вибрати аудіо- та візуальні індикатори, що робить процес навчання гнучким та адаптивним до індивідуальних потреб.

Інтерфейс Morse-It наведено на рисунку 1.1.

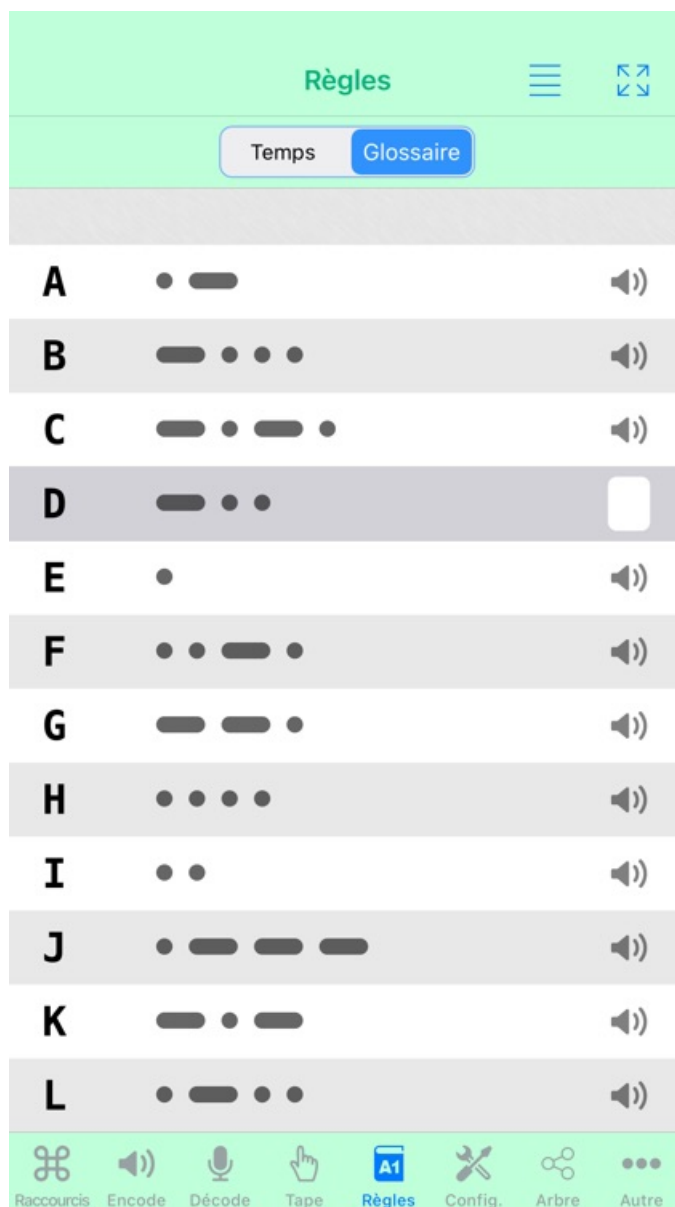


Рисунок 1.1 – Інтерфейс Morse-It

Крім базового перетворення тексту, Morse-It пропонує інтерактивні функції, які стимулюють регулярну практику та вдосконалення навичок сприйняття сигналів. Додаток дозволяє проводити тестування, що допомагає закріпити знання про символи абетки Морзе, та відслідковувати прогрес користувача. Завдяки

зручному дизайну та можливості персоналізації, навчання абетки Морзе стає не лише ефективним, а й захоплюючим процесом.

Morse Mania [11] – це інтерактивна гра, яка перетворює процес вивчення абетки Морзе на справжню пригоду. Завдяки ігровому підходу, кожен рівень та завдання стимулюють користувачів до постійного вдосконалення своїх навичок. Застосунок створено таким чином, що вивчення абетки Морзе стає не лише корисним, але й надзвичайно захоплюючим.

Інтерфейс Morse Mania наведено на рисунку 1.2.

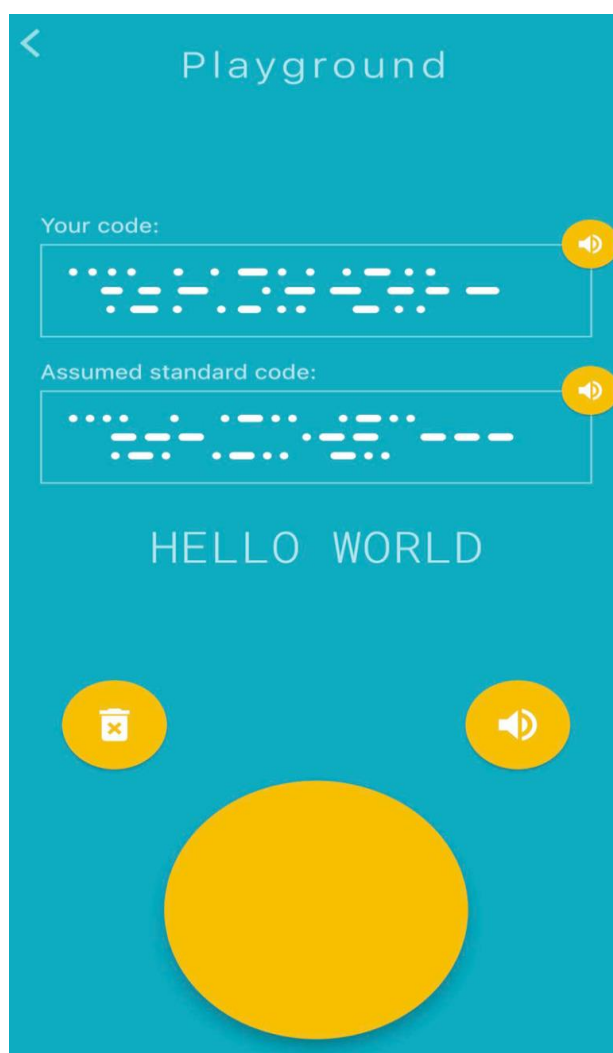


Рисунок 1.2 – Інтерфейс Morse Mania

Ігрові елементи, такі як система нагород і рейтингові таблиці, мотивують користувачів до регулярних тренувань і допомагають відчувати прогрес у навчанні.

Додаткові опції налаштувань, включаючи варіанти швидкості передачі сигналів та різні режими тренувань, дозволяють гнучко адаптувати програму до індивідуальних потреб кожного користувача.

Morse Code Flash Cards [12] – це застосунок, який використовує метод флеш-карток для ефективного запам'ятовування символів абетки Морзе. Кожна картка містить візуальне зображення символу та відповідний аудіосигнал, що дозволяє створити міцний зв'язок між звуком і знаком. Такий підхід допомагає користувачам активно повторювати та закріплювати матеріал, перетворюючи процес навчання на інтерактивну та захоплюючу гру.

Інтерфейс застосунку Morse Code Flash Cards наведено на рисунку 1.3.

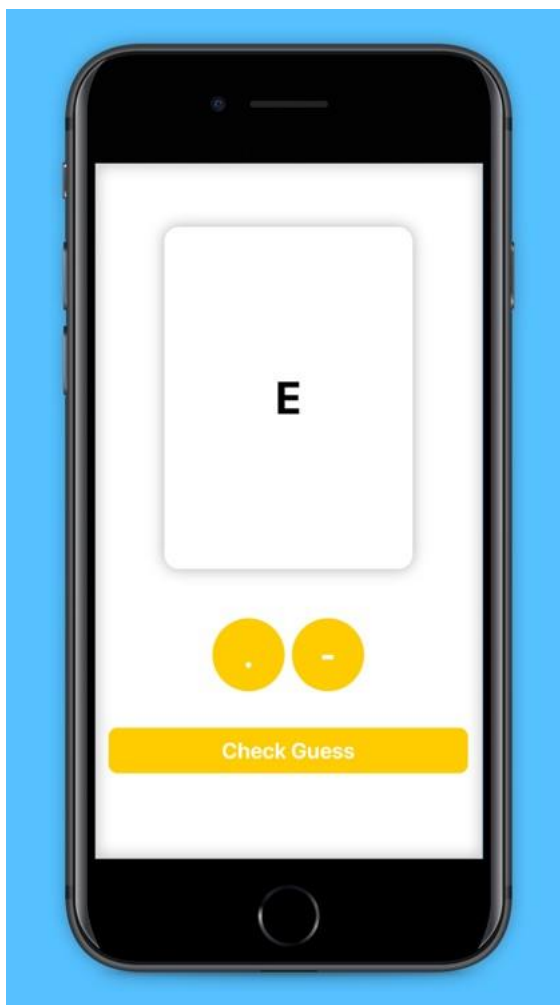


Рисунок 1.3 – Інтерфейс Morse Code Flash Cards

Крім базового перегляду карток, додаток пропонує інтерактивні тести та

вправи, які дозволяють перевіряти та аналізувати рівень засвоєння знань. Налаштовувані параметри складності та режим повторення забезпечують персоналізований підхід до вивчення, що робить Morse Code Flash Cards корисним як для початківців, так і для тих, хто прагне вдосконалити свої навички у використанні абетки Морзе.

Для порівняння характеристик і можливостей цих застосунків для вивчення абетки Морзе з власною розробкою, було сформовано таблицю 1.1.

Таблиця 1.1 – Порівняння характеристик застосунків

Характеристика	Morse-It	Morse Mania	Morse Code Flash Cards	Власна розробка
Аудіо тренування	+	+	+	+
Візуальні підказки	+	+	+	+
Режими тренування	+	+	+	+
Відслідковування прогресу	+	+	-	+
Інструкції та підказки	+	+	+	+
Адаптація навчання під темп користувача	-	-	-	+
Загальний бал	5	5	4	6

Не зважаючи на різноманіття доступних платформ, у них простежуються спільні обмеження, котрі відкривають можливість для покращення. Більшість існуючих застосунків пропонують всім користувачам однакову траєкторію навчання. Наприклад, додаток за замовчуванням йде за певним наперед визначеним списком символів (часто за методом Коха або просто по алфавіту), і саме користувач має підлаштовуватися під темп програми, а не навпаки. Якщо людина не встигає – їй доводиться самостійно повторювати рівні чи зменшувати швидкість у налаштуваннях (якщо така опція є). Якщо ж навпаки все дається легко – часто немає механізму автоматичного перескоку вперед, доводиться

проходити всі проміжні етапи. Деякі сучасніші додатки дозволяють ручну кастомізацію – наприклад, вибрати конкретний набір літер для тренування. Проте цей вибір цілком покладається на учня, тоді як програма не рекомендує нічого від себе. Це свідчить про те, що сьогоденні інструменти ще не задовольняють повністю потреби щодо адаптивності. Контент і вправи у більшості застосунків є статичними. Це може проявлятися у лінійному наборі рівнів або фіксованому переліку уроків. Користувач проходить заздалегідь підготовлені етапи, які не змінюються в залежності від його успіхів. Наприклад, якщо програма містить 50 уроків, то незалежно від того, як учень справляється, він муситиме пройти всі 50 у заданому порядку. Такий підхід не враховує, що деякі теми можуть бути вже відомі або, навпаки, потребують додаткової уваги. Статичність також означає, що відсутнє адаптивне регулювання складності: швидкість аудіо-сигналів, тривалість пауз, довжина слів – усе зафіксовано наперед або змінюється лише вручну за бажанням користувача. Відтак новачок може зіткнутися з різким стрибком складності і втратити мотивацію, а досвідченішому учню, можливо, доведеться витрачати час на зайві повторення простих елементів. Хоч деякі інструменти, реалізують елементи гри (рівні, бали), багато традиційних курсів є доволі сухими у плані подачі. Вони, як правило, обмежуються повідомленням, чи правильно користувач почув/ввів сигнал, і фіксують відсоток правильних відповідей. Немає системи нагород або веселих викликів, які б заохочували продовжувати займатися регулярно. Інтерфейс деяких програм може бути застарілим і нудним, не використовуючи сучасних можливостей графіки чи інтерактивності, що теж впливає на залученість. Крім того, часто відсутній елемент соціальної взаємодії або конкуренції – навчання відбувається повністю сам на сам з програмою, що підходить не всім.

Замість фіксованого курсу «для всіх» кожен користувач отримуватиме власний шлях освоєння абетки Морзе. Алгоритм, проаналізувавши початковий рівень (наприклад, проходження діагностичного тесту), може одразу підлаштувати програму: когось почне вчити з базових літер, а комусь запропонує відразу складніші комбінації, якщо бачить, що просте вже знайоме. Надалі траєкторія буде

видозмінюватися залежно від успіхів – це гнучкість, якої бракує існуючим застосункам.

На відміну від статичних рівнів, тут кожна вправа може автоматично підбиратися під поточний прогрес. Якщо учень довго не може засвоїти якийсь символ, програма не поспішатиме додавати нові, доки не покращиться результат із проблемним елементом. З іншого боку, якщо система фіксує, що за останні десятки спроб не зроблено жодної помилки, вона прискорить темп або додасть новий виклик (наприклад, почне змішувати цифри та літери, введе знаки пунктуації чи ускладнить повідомлення). Таке “розумне” регулювання автоматично тримає учня у зоні найбільш ефективного навчання, де завдання є достатньо важкими, але досяжними.

1.3 Аналіз методів розв’язання задачі

Для розробки вебзастосунків існують такі методи розробки:

1. Клієнтський метод орієнтований на реалізацію основної логіки безпосередньо на стороні користувача, використовуючи HTML, CSS та JavaScript [13]. Такий підхід дозволяє створити високо інтерактивний інтерфейс, який відразу реагує на дії користувача, не чекаючи відповіді від сервера. Основною перевагою є можливість роботи застосунку в офлайн-режимі, особливо коли реалізовано додаткові функції, як-от кешування даних через Service Workers. Однак, в цьому випадку виникають певні обмеження, зокрема – неможливість зберігати великі обсяги інформації або забезпечити централізований контроль над даними користувачів без допомоги серверних компонентів. Тому для завдань, що вимагають складної обробки даних або їх збереження на сервері, клієнтський підхід може бути доповнений іншими рішеннями.

2. Серверний метод передбачає розподіл логіки застосунку між клієнтом та сервером, що дозволяє централізовано управляти даними, контролювати процеси та забезпечувати персоналізований досвід для кожного користувача [14]. На серверній стороні можуть бути реалізовані алгоритми адаптивного навчання, аналітика та зберігання історії роботи користувача, що сприяє більшій гнучкості

застосунку. Інструменти, такі як Node.js, Python (з фреймворками Django чи Flask), PHP чи Ruby on Rails, дозволяють розробити RESTful API для комунікації з фронтендом. Проте використання серверного підходу вимагає більш складної інфраструктури, підвищеної уваги до питань безпеки та постійного підключення до інтернету, що може бути викликом для деяких користувачів.

3. Гібридний метод, який базується на розробці односторінкових додатків (SPA) та використанні технологій прогресивних вебзастосунків (PWA), об'єднує переваги як клієнтського, так і серверного підходів [15]. Завдяки цьому методу застосунок може працювати з високою інтерактивністю та швидкістю завдяки виконанню більшої частини логіки на стороні клієнта, а також отримувати необхідні дані з сервера для централізованого зберігання інформації. Можливості PWA, такі як робота в офлайн-режимі та інтеграція з мобільними пристроями, роблять цей підхід особливо привабливим для навчальних програм. Незважаючи на це, реалізація гібридного рішення вимагає ретельного тестування на різних пристроях і браузерах, а також складнішої архітектури для забезпечення належного функціонування сервісних компонентів.

4. Використання спеціалізованих бібліотек, які можуть значно скоротити час розробки та підвищити якість аудіо та візуальних ефектів [16]. Наприклад, бібліотеки для роботи зі звуком, як Howler.js, дозволяють просто і ефективно генерувати аудіо сигнали, що імітують точки та тире абетки Морзе. Для створення анімацій та візуальних ефектів можна використовувати Anime.js або GreenSock, що забезпечують плавність анімацій та покращують візуальне сприйняття інформації. Використання таких бібліотек дозволяє розробникам зосередитися на основному функціоналі застосунку, а не на вирішенні технічних деталей реалізації аудіо/відео компонентів. Проте, варто враховувати можливі проблеми сумісності з обраним стеком технологій і залежність від зовнішніх рішень, які можуть змінюватися з часом.

Для порівняння можливостей описаних методів та вибору найкращого та найефективнішого методу розробки, було побудовано таблицю 1.2 порівняння характеристик методів розв'язання задачі з розробки вебзастосунку для вивчення

абетки Морзе.

Таблиця 1.2 – Порівняння характеристик методів розробки вебзастосунків

Характеристика	Клієнтський метод	Серверний метод	Гібридний метод (SPA/PWA)	Спеціалізовані бібліотеки
Висока інтерактивність	+	-	+	+
Підтримка офлайн режиму	+	-	+	-
Централізоване зберігання даних	-	+	+	-
Швидке розгортання	+	-	-	+
Гнучкість та масштабованість	-	+	+	-
Загальний бал	2	2	4	2

Враховуючи наведені характеристики, було прийнято рішення об'єднати останні два методи – гібридний метод та спеціалізовані бібліотеки. Можливості гібридного методу дозволять працювати в офлайн-режимі та зберігати дані централізовано, а використання спеціалізованих бібліотек пришвидшить розробку в частині роботи зі звуками абетки Морзе. Однак, варто зазначити, що таке поєднання негативно вплине на гнучкість та масштабованість через залежність від зовнішніх бібліотек, що розробляються сторонніми розробниками, їх обмеження та можливі зміни кодової бази цих бібліотек.

1.4 Постановка задач дослідження

Провівши аналіз методів розв'язання поставленої задачі, аналіз стану

застосунків для вивчення абетки Морзе, порівняння аналогів було поставлено такі задачі дослідження:

- розробити архітектуру застосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу;
- розробити інтерфейс користувача застосунку;
- розробити метод персоналізованого підходу;
- розробити алгоритми роботи застосунку;
- розробити функціонал застосунку;
- провести тестування розробленого застосунку.

Технічне завдання на розробку наведено в додатку А.

1.5 Висновки

В цьому розділі бакалаврської кваліфікаційної роботи було проведено аналіз сучасного стану застосунків для вивчення абетки Морзе, доведено актуальність розробки такого застосунку. Проведено порівняльний аналіз аналогів з розроблюваним вебзастосунком, продемонстровано його переваги та таким чином доведено доцільність розробки. Проведено аналіз методів розв'язання задачі, обрано гібридний метод з використанням бібліотек для роботи зі звуками. Проведено постановку задач дослідження з розробки вебзастосунку з вивчення абетки Морзе на основі адаптивного персоналізованого підходу.

2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ЗАСТОСУНКУ

2.1 Аналіз інформаційного забезпечення

При роботі вебзастосунку для вивчення абетки Морзе необхідний певний набір даних. Також, певні дані генеруються внаслідок взаємодії користувача з ним.

Дані навчального прогресу формують динамічну складову профілю користувача, що безперервно оновлюється в процесі навчання. Ця категорія включає детальну інформацію про вивчені символи Морзе з точними показниками рівня засвоєння для кожного з них. Особливе значення мають матриці помилок, що відображають проблемні пари символів та типи помилок, характерні для конкретного користувача. Система зберігає історію тривалості та частоти навчальних сесій, аналізуючи оптимальні режими роботи. Відстежуються динамічні показники швидкості декодування, що дозволяє системі адаптувати темп подачі матеріалу та визначати готовність до переходу на наступний рівень складності.

Для проведення тестування та навчання абетці Морзе у застосунок наперед завантажено повний набір символів міжнародної системи Морзе з різними формами їх представлення. Кожен символ зберігається у вигляді аудіозаписів з різною швидкістю передачі (від початкової 5 WPM до професійної 20+ WPM), що дозволяє адаптувати складність сприйняття залежно від рівня користувача. Також завантажено візуальне представлення символів у вигляді послідовностей крапок і тире. Символи Морзе групуються за кількома параметрами: за складністю (від найпростіших E, T до складних комбінацій), за частотою використання в реальних повідомленнях, за рівнем акустичної схожості (для виявлення потенційно проблемних пар). Така класифікація дозволяє системі формувати оптимальні набори символів для вивчення на кожному етапі, мінімізуючи інтерференцію між схожими символами на початкових етапах та поступово вводячи складніші диференціації.

Навчальні вправи представляють другий великий блок контенту, що включає

структуровані набори тренувальних завдань різної складності. Для методу когнітивно-адаптивної прогресії розробляються спеціалізовані вправи для подолання конкретних типів помилок, наприклад, для розрізнення акустично схожих символів або покращення сприйняття певної тривалості сигналів. Для методу нейроморфного навчання створюються імерсивні сценарії – контекстуалізовані історії з елементами гейміфікації, де вивчення Морзе інтегровано в сюжетну лінію. Кожен сценарій розробляється з кількома рівнями складності та можливістю динамічної адаптації до показників успішності користувача.

Система також містить бібліотеку тестових матеріалів для оцінки засвоєння, включаючи стандартизовані тести для перевірки швидкості та точності декодування, а також контекстуальні перевірки для оцінки практичного застосування знань. Навчальний контент розподіляється на модулі прогресії, забезпечуючи поступове нарощування складності та введення нових символів у відповідності до індивідуальних показників користувача.

Навчальна аналітика є критичним компонентом інформаційної системи, що забезпечує адаптивність навчального процесу. Система накопичує та аналізує детальну статистику успішності за різними категоріями символів, виявляючи як загальні тенденції, так і індивідуальні особливості кожного користувача. Особлива увага приділяється аналізу патернів помилок – система відстежує не лише факт помилки, але й її тип, контекст виникнення, час реакції користувача, періодичність повторення конкретного типу помилок.

Інформаційна система зберігає та аналізує дані про ефективність різних методів навчання для кожного користувача, визначаючи, яка комбінація навчальних стратегій дає найкращі результати в конкретному випадку. Система також збирає детальну інформацію про час, затрачений на засвоєння різних символів та груп символів, що дозволяє прогнозувати темп навчання та адаптувати навчальний план відповідно до індивідуальних показників прогресу.

Прогностичні моделі формуються на основі аналізу накопичених даних та генерують індивідуалізовані рекомендації для подальшого навчання. Одним із

ключових елементів є побудова персоналізованих кривих забування на основі теорії Еббінгауза, адаптованих до індивідуальних особливостей пам'яті користувача. Система використовує ці криві для розрахунку оптимальних інтервалів повторення вивченого матеріалу, запобігаючи забуванню та максимізуючи ефективність навчання.

Аналітична система також генерує прогнози щодо очікуваного часу повного засвоєння матеріалу, виявляє потенційні "вузькі місця" у навчальному процесі та розробляє стратегії їх подолання. Постійно відстежується ефективність персоналізованих рекомендацій, що дозволяє системі самовдосконалюватися та підвищувати точність прогнозування з накопиченням даних про користувача.

2.2 Розробка архітектури застосунку

Архітектура вебзастосунку для вивчення абетки Морзе побудована за сучасним підходом клієнт-сервер, що забезпечує ефективний розподіл відповідальності між різними компонентами системи. Такий підхід дозволяє досягти високої масштабованості, надійності та зручності в розробці й супроводі [17]. Застосунок розділений на сторону клієнта та сторону сервера які комунікують між собою через RESTful API.

Сторона клієнта розроблена на базі фреймворка React, який забезпечує створення інтерфейсу користувача. Для управління станом застосунку використовується Redux, що дозволяє централізовано зберігати дані та керувати їх оновленням у всіх компонентах. React-компоненти організовані за принципом модульності та повторного використання, що спрощує розробку нових функцій та підтримку існуючих.

Для роботи зі звуком, що є необхідним для застосунку з вивчення абетки Морзе, використовується Web Audio API. Цей компонент відповідає за генерацію аудіосигналів Морзе з правильним таймінгом для крапок, тире та пауз, а також забезпечує налаштування швидкості передачі. Сервіси для роботи з аудіо відокремлені від компонентів інтерфейсу, що дозволяє легко модифікувати їх без впливу на інші частини системи.

Сторона сервера реалізована на Node.js з використанням фреймворку Express, що забезпечує швидку роботу, асинхронну обробку запитів та ефективну взаємодію з базою даних. Серверна частина відповідає за аутентифікацію користувачів, обробку та збереження даних про їхній прогрес, керування навчальними сценаріями та генерацію персоналізованих навчальних планів.

Для зберігання даних використовується MongoDB – нереляційна база даних, що забезпечує гнучкість у зберіганні складних структур даних, таких як матриці помилок, прогрес користувача та деталізовані сценарії навчання. Схеми Mongoose використовуються для структурування даних та забезпечення їх валідації перед збереженням. Архітектура бази даних оптимізована для швидкого доступу до персоналізованої інформації про навчальний прогрес користувачів.

Комунікація між клієнтом і сервером здійснюється через RESTful API, що забезпечує стандартизований інтерфейс для обміну даними. API-ендпоїнти структуровані за функціональними модулями: аутентифікація, профіль користувача, навчальний прогрес, сценарії, тестування. Для безпечної передачі даних використовується HTTPS, а аутентифікація запитів забезпечується через JSON Web Tokens (JWT).

Схема архітектури системи наведена на рисунку 2.1.

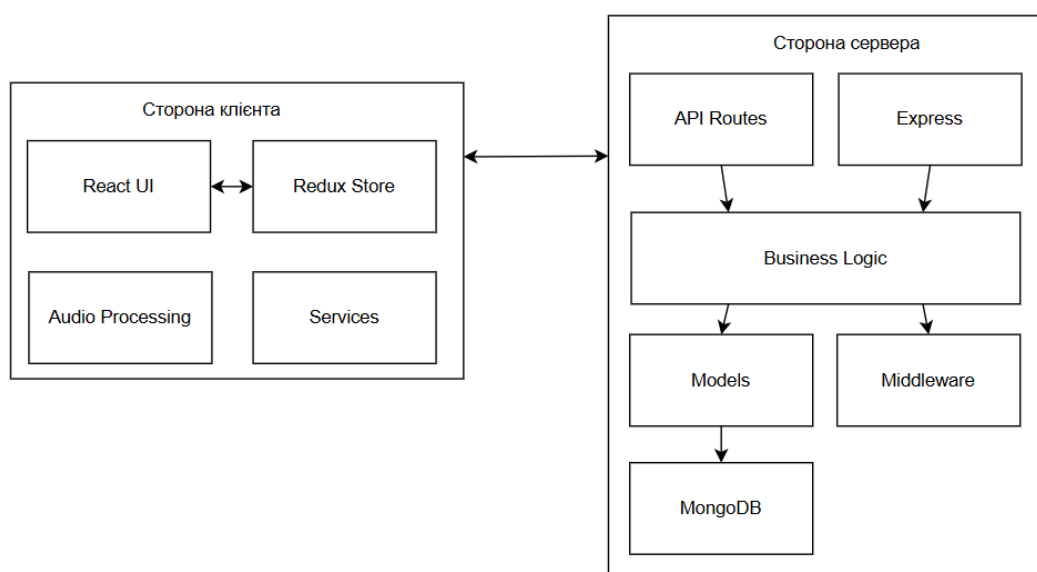


Рисунок 2.1 – Архітектура системи

Така архітектура дозволяє системі масштабуватися та адаптуватися до зростаючої кількості користувачів і розширення функціональності, та забезпечує реалізацію усього необхідного функціоналу для вивчення абетки Морзе.

2.3 Розробка методу когнітивно-адаптивної прогресії з аналізом патернів помилок

Когнітивно-адаптивна прогресія представляє підхід, що базується на глибокому аналізі індивідуальних помилок користувача та динамічній адаптації навчального процесу. Цей метод фокусується на розпізнаванні специфічних патернів помилок, які робить користувач, та використанні цієї інформації для формування персоналізованої навчальної траєкторії.

Основою методу є детальне відстеження та аналіз помилок користувача через створення "матриці помилок" – спеціальної структури даних, що відображає, які символи користувач плутає між собою та з якою частотою. Наприклад, система може виявити, що користувач часто плутає літери "A" (·—) та "H" (—·), або має труднощі з розрізненням символів із схожими ритмічними структурами.

На основі аналізу помилок система динамічно коригує навчальний план, змінюючи частоту появи проблемних символів у вправах, створюючи спеціалізовані тренувальні послідовності для відпрацювання складних комбінацій, та адаптуючи швидкість передачі сигналів для різних груп символів. Цей підхід дозволяє зосередити навчальні зусилля саме на тих аспектах, які викликають найбільші труднощі у конкретного користувача. Реалізовано прогресивне розширення набору символів – нові символи вводяться тільки після досягнення порогового значення точності з поточним набором. Система враховує не лише загальну точність, але й стабільність розпізнавання символів у часі та при різних швидкостях передачі.

Алгоритм роботи методу :

1. Початкова оцінка користувача. Сюди входить: проведення короткого тесту для визначення початкового рівня сприйняття аудіо та швидкості реакції, збір

даних про попередній досвід з Морзе, створення початкового профілю користувача (рисунок 2.2).

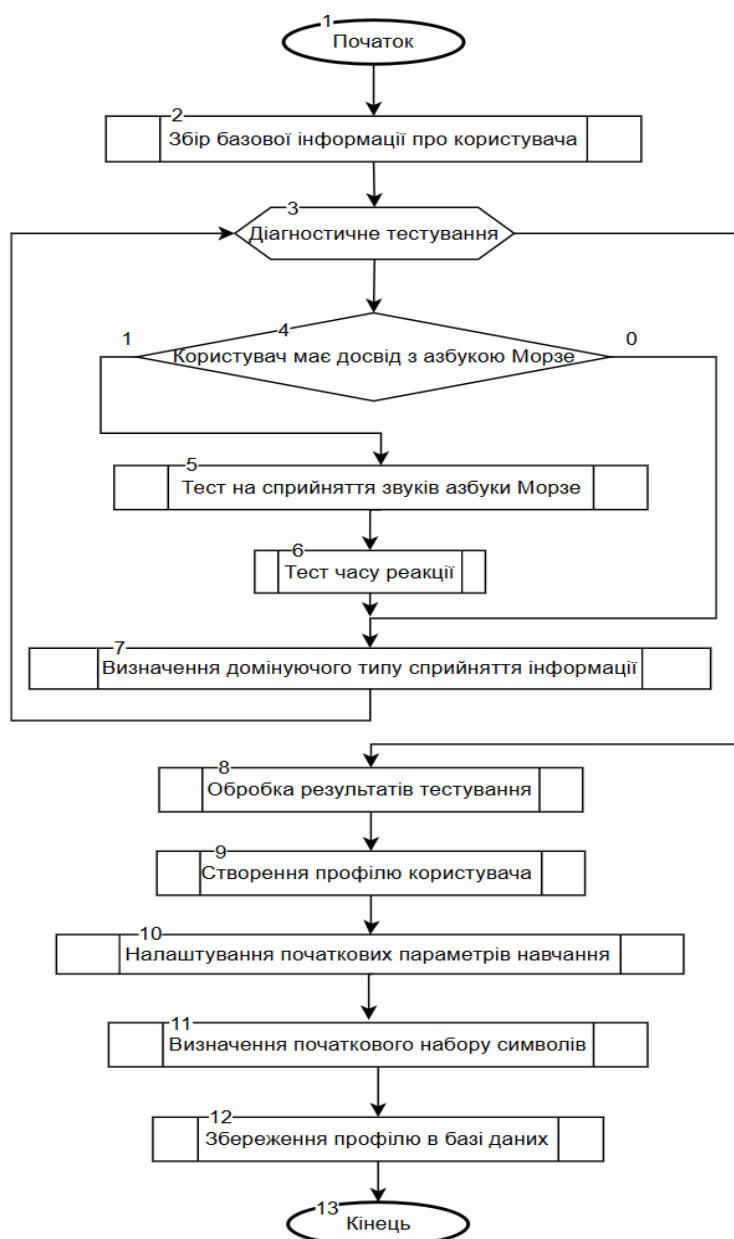


Рисунок 2.2 – Блок-схема алгоритму початкової оцінки користувача

2. Представлення символів через аудіо та візуальний супровід, починаючи з повільної швидкості передачі символів з поступовим збільшення швидкості на основі показників засвоєння.

3. Відстеження специфічних пар символів, які користувач плутає, визначення типових помилок, на основі яких створюється «матриця помилок» для

кожного користувача.

4. Поступове збільшення частоти появи проблемних символів та автоматичне коригування швидкості WPM для різних груп символів.

5. Після досягнення 90% точності у визначення проблемних символів, система розпізнає це як успішне їх опанування, та додає нові символи.

6. На основі успішності навчання, формуються звіти про прогрес з візуалізацією даних, які представлені у вигляді графіків та діаграм. На основі отриманого прогресу у навчанні, формуються автоматичні рекомендації щодо оптимальної тривалості та частоти тренувань.

Розроблений метод забезпечує адаптивний процес вивчення абетки Морзе, формує персоналізований підхід до вивчення на основі наявних про користувача даних, що дозволяє швидше та ефективніше опанувати абетку Морзе.

2.4 Розробка методу навчання з імерсивними контекстними сценаріями

Нейроморфне навчання з імерсивними контекстними сценаріями представляє собою метод вивчення абетки Морзе, що занурює користувача в наративні контексти, де використання Морзе має практичне значення. Ключовою ідеєю методу є перетворення абстрактних символів та звуків на осмислену комунікацію в сюжетно-обумовлених ситуаціях.

На основі інформації про користувача, система формує персоналізовану модель навчання та пропонує відповідні тематичні сценарії.

Імерсивні контекстні сценарії, такі як морська подорож, космічна місія чи історія порятунку, створюють змістовний наратив, в якому використання абетки Морзе стає невід'ємною частиною процесу. В кожному сценарії користувач виконує послідовність сюжетних завдань, де необхідно декодувати повідомлення, передавати координати, відповідати на сигнали про допомогу, тощо. Ці сценарії адаптуються до інтересів користувача та рівня його навичок. Важливим компонентом методу є ритмічно-музична синхронізація, яка використовує генерування ритмічних патернів, що відповідають кодам Морзе, та налаштування темпу і тону сигналів відповідно до індивідуальних параметрів

сприйняття користувача. З прогресом навчання система поступово наближає ці ритми до стандартних швидкостей передачі Морзе.

Алгоритм роботи методу:

1. Аналіз типу сприйняття інформації користувачем.
2. Створення динамічної моделі навчання шляхом формування персоналізованої карти складності для кожного символу Морзе.
3. Створення тематичних сценаріїв з Адаптацією сценаріїв на основі інтересів користувача.
4. Генерування ритмічних патернів, що відповідають кодам Морзе, налаштування темпу та тону сигналів відповідно до індивідуальних параметрів сприйняття.
5. Динамічне коригування складності на основі опанування користувачем абетки Морзе (рисунок 2.3).

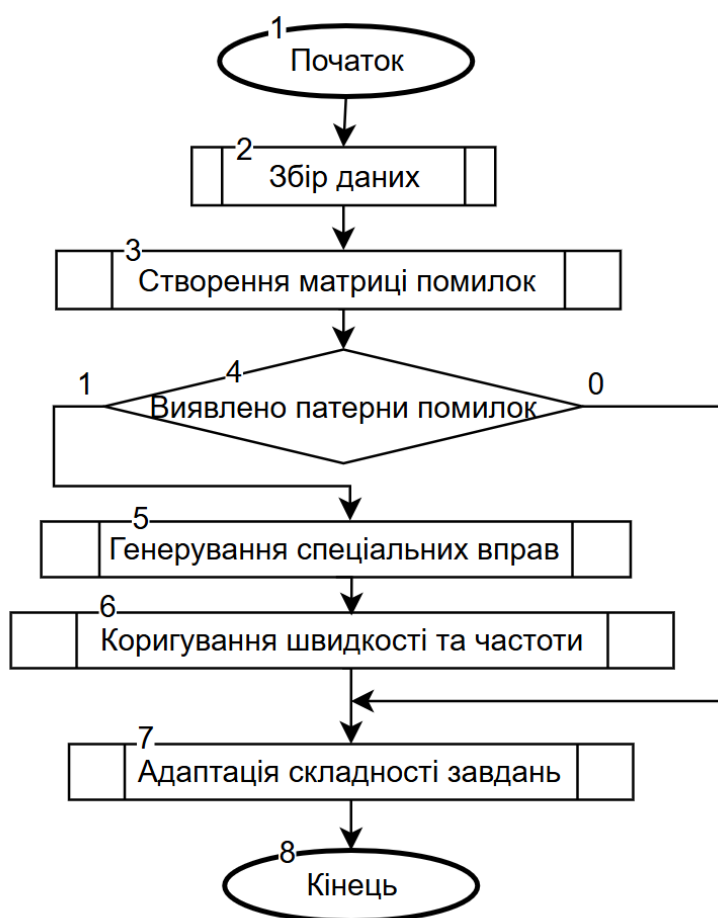


Рисунок 2.3 – Блок-схема алгоритму коригування складності

6. Автоматичне виявлення символів, що "забуваються" з часом та проактивне повторення матеріалу на основі передбачення можливого забування (рисунок 2.4).

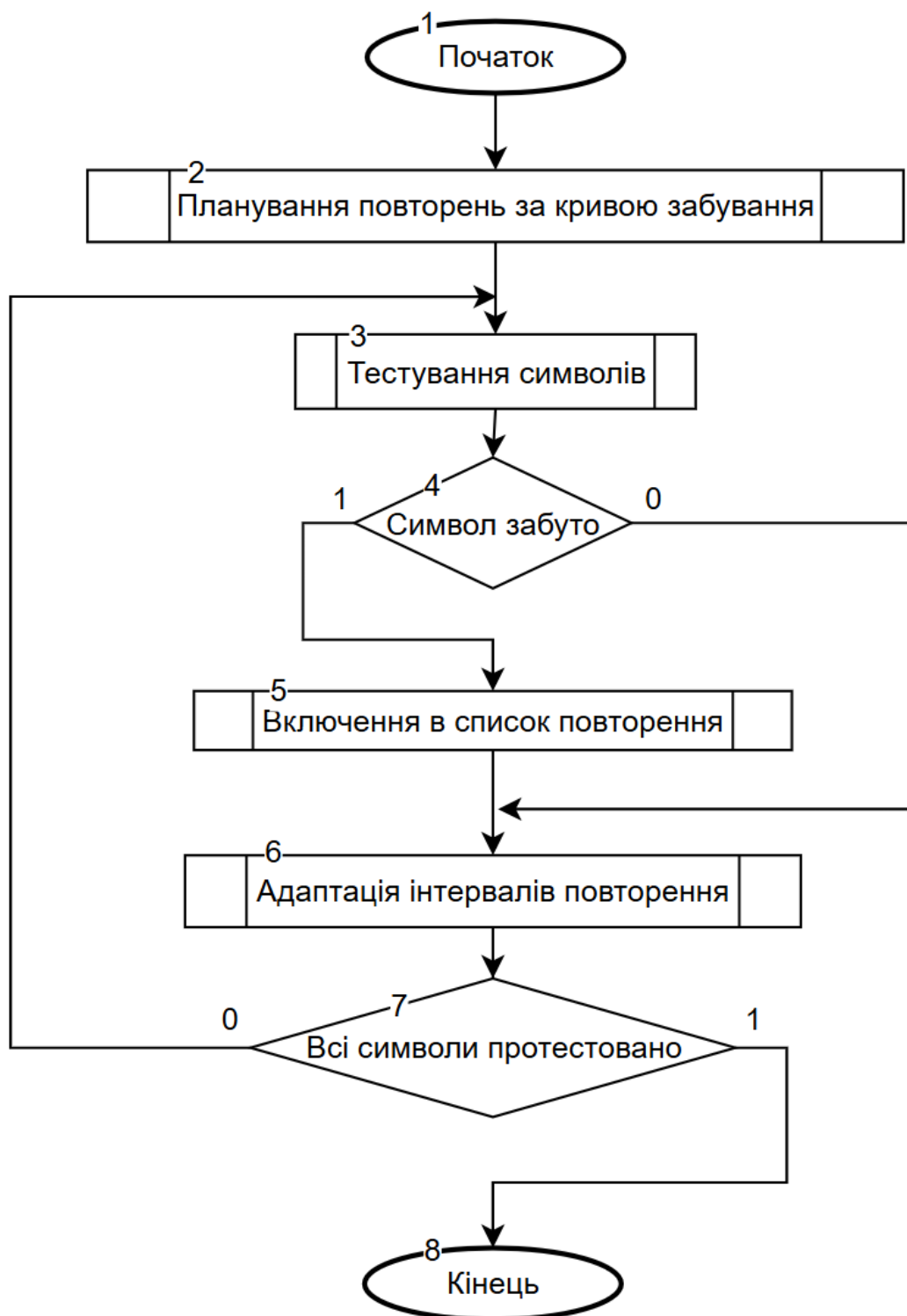


Рисунок 2.4 – Блок-схема алгоритму відстеження довготривалого запам'ятовування

Розроблений метод дозволяє практикувати отримані навички з абетки Морзе у ситуаціях, близьких до реальних, з елементами гейміфікації. Це дозволяє розвивати отримані початкові навички, та краще їх закріплювати, знаходити їм практичне призначення.

2.5 Висновки

У другому розділі було проведено аналіз інформаційного забезпечення системи для вивчення абетки Морзе, описано дані, які у ній використовуються та формуються. Розроблено архітектуру системи за клієнт-серверним підходом, описано складові архітектури. Розроблено метод когнітивно-адаптивної прогресії з аналізом патернів помилок та метод навчання з імерсивними контекстними сценаріями, побудовано блок-схеми алгоритмів складових цих методів.

3 РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АБЕТКИ МОРЗЕ

3.1 Варіантний аналіз та вибір мови програмування розробки

Для вибору мови програмування розробки, порівняємо такі мови програмування, що використовуються у веб-розробці: JavaScript, Python, PHP.

JavaScript [18] – динамічна мова програмування, яка дозволяє створювати інтерактивні елементи на веб-сторінках. Для розробки застосунку з вивчення абетки Морзе JavaScript надає можливість реалізувати відтворення звукових сигналів, візуальну анімацію та миттєвий зворотний зв'язок користувачу без перезавантаження сторінки.

JavaScript працює безпосередньо у браузері користувача, що забезпечує швидку реакцію інтерфейсу. Завдяки багатій екосистемі бібліотек та фреймворків (React, Vue, Angular), можна швидко розробити адаптивний та зручний інтерфейс для вивчення абетки Морзе з різними інтерактивними вправами та тренажерами.

Python [19] – високорівнева мова програмування із зрозумілим синтаксисом, яка широко використовується для веб-розробки завдяки фреймворкам Django та Flask. Для створення застосунку з вивчення абетки Морзе Python дозволяє розробити надійну серверну частину, яка може зберігати прогрес користувача, аналізувати його успішність та надавати персоналізовані рекомендації.

Використання Python для вебзастосунку передбачає обробку логіки на сервері, що може уповільнити реакцію інтерфейсу для користувача. Проте Python має потужні можливості для роботи з даними та аналітикою, що дозволяє реалізувати складні алгоритми для адаптивного навчання та відстеження прогресу вивчення абетки Морзе.

PHP [20] є серверною мовою програмування, спеціально розробленою для веб-розробки. Для застосунку з вивчення абетки Морзе PHP дозволяє створити систему авторизації користувачів, зберігати їхній прогрес у базі даних та генерувати динамічний контент на основі рівня користувача.

PHP має низький поріг входження для початківців та широку підтримку різними хостинг-провайдерами. Однак для створення інтерактивних елементів,

таких як звукове відтворення сигналів Морзе чи анімовані візуалізації, PHP потребує додаткового використання JavaScript на стороні клієнта, що ускладнює архітектуру застосунку.

З метою виявлення мови програмування, що найкраще підходить для розробки вебзастосунку для вивчення абетки Морзе, було сформовано таблицю 3.1, що порівнює характеристики JavaScript, Python та PHP.

Таблиця 3.1 – Порівняння мов програмування

Характеристика	JavaScript	Python	PHP
Нативна підтримка у веб-браузерах	+	-	-
Нативний доступ до Web Audio API	+	-	-
Підтримка асинхронних операцій	+	+	+
Об'єктно-орієнтоване програмування	+	+	+
Підтримка REST API	+	+	+
Доступ до DOM-елементів без додаткових бібліотек	+	-	-
Нативна підтримка JSON	+	+	+
Підтримка WebSockets	+	+	+
Сумарний бал	8	5	5

Таким чином, JavaScript має вищий сумарний бал за Python та PHP на 37,5% ($100\% - 5/8 \cdot 100\% = 37,5\%$). Отже, JavaScript є найкращою мовою розробки вебзастосунку для вивчення абетки Морзе. Вона надає нативний доступ до Web Audio API, що необхідний для відтворення звуків абетки Морзе, має нативну підтримку у веб-браузерах, що сприяє легкій розробці, впровадженню використанню застосунку, та надає доступ до DOM-елементів без додаткових бібліотек.

3.2 Розробка бази даних

З метою збереження інформації про користувачів для подальшої їх обробки та формування персоналізованого підходу до навчання, збереження матриці помилок, сформованих іммерсивних сценаріїв, тестів та результатів тестів необхідне джерело збереження даних. Для цього найкраще підходить реалізація власної бази даних [21].

Таблиця `users` зберігає основну інформацію про зареєстрованих користувачів системи. Вона має такі поля:

- `id`: унікальний ідентифікатор користувача;
- `username`: унікальне ім'я для входу в систему;
- `email`: електронна адреса користувача для комунікації та відновлення доступу;
- `password`: зашифрований пароль;
- `created_at`: дата створення облікового запису;
- `updated_at`: дата останнього оновлення даних.

Таблиця `user_profiles` зберігає розширену особисту інформацію про користувачів. Вона має такі поля:

- `id`: унікальний ідентифікатор профілю;
- `user_id`: зв'язок з таблицею користувачів;
- `full_name`: повне ім'я користувача;
- `age`: вік користувача;
- `experience`: попередній досвід з абеткою Морзе (`none`, `beginner`, `intermediate`, `advanced`);
- `learning_goal`: мета вивчення абетки Морзе;
- `preferred_method`: обраний метод навчання (`cognitive`, `immersive`, `hybrid`);
- `created_at`: дата створення профілю;
- `updated_at`: дата останнього оновлення.

Таблиця `cognitive_profiles` зберігає інформацію про когнітивні особливості користувача для персоналізації навчання. Вона має такі поля:

- id: унікальний ідентифікатор когнітивного профілю;
- user_id: зв'язок з таблицею користувачів;
- perception_type: домінуючий тип сприйняття (visual, auditory, kinesthetic);
- rhythmic_ability: оцінка ритмічних здібностей (0-100);
- reaction_time: середній час реакції в мілісекундах;
- memory_score: оцінка пам'яті користувача (0-100);
- interests: список інтересів у форматі JSON для підбору тематичних сценаріїв;
- created_at: дата створення когнітивного профілю;
- updated_at: дата останнього оновлення.

Таблиця morse_symbols зберігає інформацію про всі символи абетки Морзе. Вона має такі поля:

- id: унікальний ідентифікатор символу;
- symbol: сам символ (літера, цифра, знак пунктуації);
- morse_code: представлення символу в абетці Морзе (крапки та тире);
- category: категорія символу (літера, цифра, знак пунктуації, спеціальний символ);
- complexity: рівень складності символу (1-10);
- audio_path: шлях до аудіофайлу для цього символу.

Таблиця learning_progress відстежує прогрес вивчення кожного символу для кожного користувача. Вона має такі поля:

- id: унікальний ідентифікатор запису;
- user_id: користувач, чий прогрес відстежується;
- symbol_id: символ, для якого відстежується прогрес;
- correct_attempts: кількість правильних спроб розпізнавання;
- total_attempts: загальна кількість спроб розпізнавання;
- avg_response_time: середній час відповіді в секундах;
- last_practiced: дата останньої практики з цим символом.

Таблиця error_matrices зберігає інформацію про помилки користувачів для

адаптивного навчання. Вона має такі поля:

- id: унікальний ідентифікатор запису;
- user_id: користувач, чий помилки відстежуються;
- expected_symbol_id: символ, який мав бути розпізнаний;
- actual_symbol_id: символ, який помилково назвав користувач;
- error_count: кількість таких помилок;
- last_error_at: дата останньої такої помилки.

Таблиця overall_progress відстежує загальний прогрес навчання користувача.

Вона має такі поля:

- id: унікальний ідентифікатор запису;
- user_id: користувач, чий загальний прогрес відстежується;
- active_morse_speed: поточна швидкість передачі в WPM (слів за хвилину);
- overall_progress: загальний прогрес у відсотках (0-100);
- total_practice_time: загальний час практики у хвилинах;
- last_activity: дата останньої активності.

Таблиця learning_plans зберігає персоналізовані плани навчання, створені системою для користувачів. Вона має такі поля:

- id: унікальний ідентифікатор плану навчання;
- user_id: користувач, для якого створено план;
- recommended_symbols: список символів, рекомендованих для вивчення (JSON);
- problematic_symbols: список символів, які викликають труднощі (JSON);
- created_at: дата створення плану;
- updated_at: дата останнього оновлення плану.

Таблиця scenarios зберігає інформацію про імєрсивні сценарії для вивчення Морзе. Вона має такі поля:

- id: унікальний ідентифікатор сценарію;
- title: назва сценарію;
- description: опис сценарію;

- `image_path`: шлях до зображення для сценарію;
- `tags`: теги для категоризації сценарію (JSON);
- `min_level`: мінімальний рівень користувача для сценарію;
- `max_level`: максимальний рівень користувача для сценарію;
- `created_at`: дата створення сценарію;
- `updated_at`: дата останнього оновлення.

Таблиця `scenario_stages` зберігає етапи кожного сценарію. Вона має такі поля:

- `id`: унікальний ідентифікатор етапу;
- `scenario_id`: сценарій, до якого належить етап;
- `sequence_number`: порядковий номер етапу в сценарії;
- `title`: назва етапу;
- `narrative`: текстовий наратив етапу;
- `objectives`: цілі етапу у форматі JSON;
- `created_at`: дата створення етапу;
- `updated_at`: дата останнього оновлення.

Таблиця `scenario_exercises` зберігає вправи з Морзе для кожного етапу сценарію. Вона має такі поля:

- `id`: унікальний ідентифікатор вправи;
- `stage_id`: етап, до якого належить вправа;
- `exercise_type`: тип вправи (декодування, кодування, розпізнавання);
- `title`: назва вправи;
- `instructions`: інструкції до вправи;
- `content`: зміст вправи (код Морзе або текст);
- `answer`: правильна відповідь;
- `base_wpm`: базова швидкість передачі в WPM;
- `target_level`: цільовий рівень складності;
- `show_morse`: чи показувати код Морзе під час вправи;
- `created_at`: дата створення вправи;

Таблиця `scenario_rewards` зберігає інформацію про нагороди, які можна отримати за проходження етапів. Вона має такі поля:

- `id`: унікальний ідентифікатор нагороди;
- `stage_id`: етап, за який надається нагорода;
- `reward_type`: тип нагороди (досвід, відзнака, розблокування);
- `value`: значення нагороди;
- `created_at`: дата створення запису про нагороду.

Таблиця `scenario_progress` відстежує прогрес користувачів у проходженні сценаріїв. Вона має такі поля:

- `id`: унікальний ідентифікатор запису прогресу;
- `user_id`: користувач, чий прогрес відстежується;
- `scenario_id`: сценарій, прогрес якого відстежується;
- `active_stage_id`: поточний активний етап сценарію;
- `completed_stages`: список завершених етапів у форматі JSON;
- `created_at`: дата початку проходження сценарію;
- `updated_at`: дата останнього оновлення прогресу.

Таблиця `stage_results` зберігає результати проходження етапів сценаріїв. Вона має такі поля:

- `id`: унікальний ідентифікатор результату;
- `user_id`: користувач, чий результат зберігається;
- `stage_id`: етап сценарію, якого стосується результат;
- `success`: чи успішно пройдено етап;
- `completion_time`: час проходження в секундах;
- `answers`: відповіді користувача у форматі JSON;
- `completed_at`: Дата завершення етапу.

Таблиця `user_rewards` зберігає нагороди, отримані користувачами за різні досягнення. Вона має такі поля:

- `id`: унікальний ідентифікатор запису;
- `user_id`: користувач, який отримав нагороду;

- reward_type: тип нагороди (досвід, відзнака, розблокування);
- value: значення нагороди;
- scenario_id: сценарій, за який отримано нагороду (якщо застосовно);
- stage_id: етап, за який отримано нагороду (якщо застосовно);
- obtained_at: дата отримання нагороди.

Таблиця tests зберігає інформацію про тести для оцінювання знань абетки Морзе. Вона має такі поля:

- id: унікальний ідентифікатор тесту;
- title: назва тесту;
- description: опис тесту;
- difficulty: рівень складності (beginner, intermediate, advanced);
- estimated_time_minutes: орієнтовний час проходження в хвилинах;
- passing_score: прохідний бал у відсотках;
- active: чи активний тест для проходження;
- created_at: дата створення тесту;
- updated_at: дата останнього оновлення.

Таблиця test_questions зберігає питання для тестів. Вона має такі поля:

- id: унікальний ідентифікатор питання;
- test_id: тест, до якого належить питання;
- text: текст питання;
- question_type: тип питання (вибір з варіантів, введення тексту, розпізнавання Морзе);
- options: варіанти відповідей для питань з вибором (JSON);
- correct_answer: правильна відповідь;
- morse_code: код Морзе для питань на розпізнавання;
- wpm: швидкість передачі коду в WPM;
- category: категорія питання для аналітики;
- points: кількість балів за правильну відповідь;
- created_at: дата створення питання.

Таблиця `test_results` зберігає результати проходження тестів користувачами. Вона має такі поля:

- `id`: унікальний ідентифікатор результату;
- `user_id`: користувач, який проходив тест;
- `test_id`: тест, який було пройдено;
- `score`: отриманий бал у відсотках;
- `correct_answers`: кількість правильних відповідей;
- `incorrect_answers`: кількість неправильних відповідей;
- `average_response_time`: середній час відповіді в секундах;
- `category_performance`: результати за категоріями у форматі JSON;
- `passed`: чи успішно пройдено тест;
- `details`: детальна інформація про відповіді у форматі JSON;
- `completed_at`: дата завершення тесту.

Таблиця `learning_sessions` відстежує навчальні сесії користувачів. Вона має такі поля:

- `id`: унікальний ідентифікатор сесії;
- `user_id`: користувач, чия сесія відстежується;
- `session_type`: тип сесії (`cognitive`, `immersive`, `practice`, `test`);
- `started_at`: час початку сесії;
- `ended_at`: час завершення сесії;
- `duration`: тривалість сесії в секундах;
- `symbols_practiced`: символи, які практикувалися під час сесії (JSON).

Спрощена схема бази даних, що включає основні її таблиці, наведена на рисунку 3.1.



Рисунок 3.1 – Схема бази даних застосунку для вивчення абетки Морзе

3.3 Розробка модулів системи

Розробка модулів системи для вивчення абетки Морзе передбачає реалізацію функціоналу, необхідного для вивчення абетки Морзе та практикування її використання, а також розробки відповідного інтерфейсу, який буде легким для

розуміння та використання користувачем.

На рисунку 3.2 наведено блок-схему загального алгоритму роботи системи.

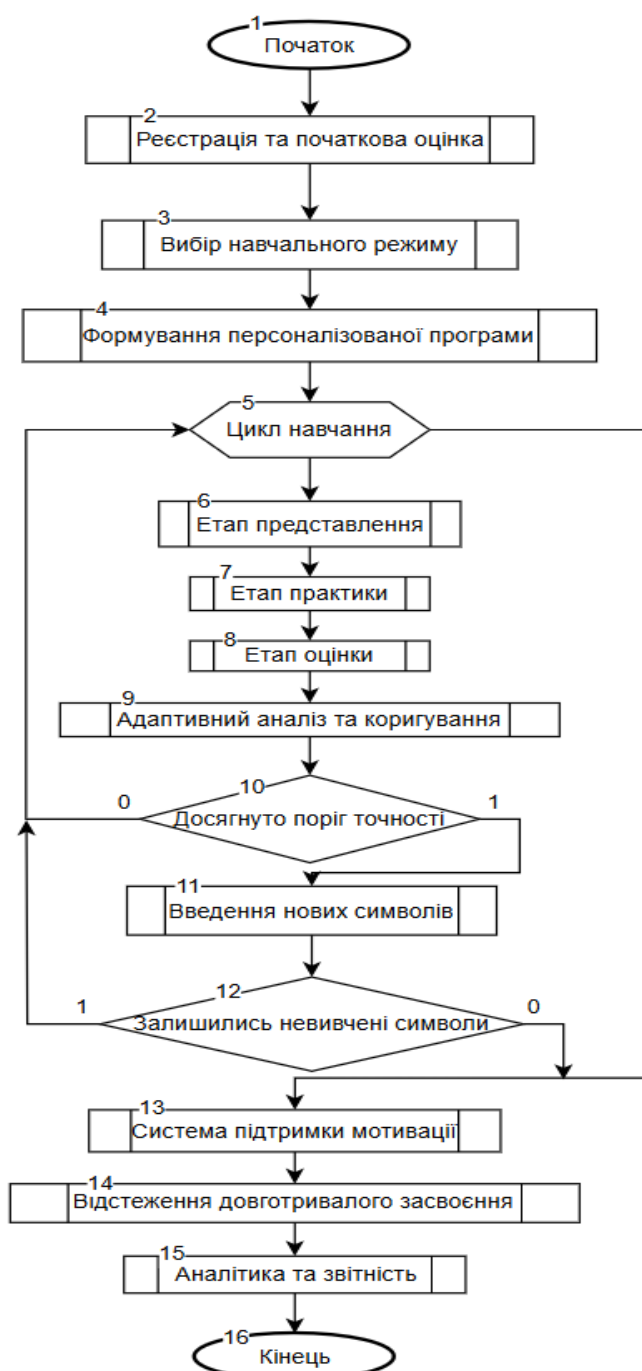


Рисунок 3.2 – Блок-схема загального алгоритму роботи системи

Процес взаємодії користувача із системою розпочинається з процесу реєстрації, в ході якого користувач вводить початкові дані про свої наявні знання абетки Морзе, відповідає на запитання, що дозволяють визначити його

особливості у навчанні та рівень складності, з якого необхідно розпочати навчання, та формується персоналізований підхід до навчання.

Після процесу реєстрації та формування персоналізованого підходу відбувається процес навчання, який являє собою етап представлення (демонстрація символів абетки Морзе), етап практики та оцінки (тестування та імерсивні сценарії), етап перевірки рівня знань.

Після вивчення символів, регулярно проводиться процес перевірки знань та виявлення забутих символів.

Для роботи такої системи, розроблено інтерфейс користувача. Розроблено головну сторінку застосунку, структура якої наведена на рисунку 3.3.

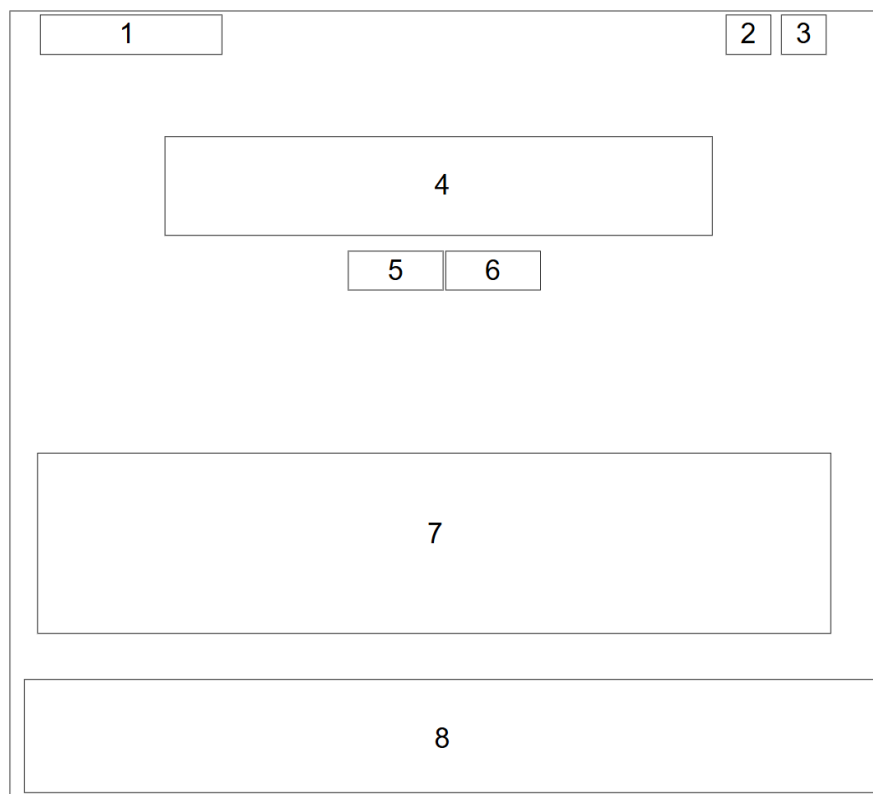


Рисунок 3.3 – Структурна схема інтерфейсу головної сторінки застосунку

Складові елементи головної сторінки застосунку:

1. Назва застосунку
2. Кнопка авторизації.
3. Кнопка реєстрації.

4. Опис застосунку.
5. Кнопка «Розпочати».
6. Кнопка авторизації.
7. Короткий опис методів навчання.
8. Футер сторінки.

Сторінка реєстрації – це сторінка, яку користувач бачить при першій взаємодії із системою. У ній користувач вводить свої особисті дані, та інформацію про свій досвід вивчення та використання абетки Морзе. Структура інтерфейсу сторінки реєстрації наведена на рисунку 3.4.

The diagram illustrates the layout of a registration page. It features a header area with three elements: a large box labeled '1' on the left, and two smaller boxes labeled '2' and '3' on the right. The main content area contains a vertical stack of input fields, numbered 4 through 13. Fields 4, 5, 6, 7, 8, 9, 10, and 11 are of uniform width. Field 12 is narrower, and field 13 is wider than field 12. At the bottom of the page is a wide footer box labeled '14'.

Рисунок 3.4 – Структурна схема інтерфейсу сторінки реєстрації

Складові елементи інтерфейсу сторінки реєстрації

1. Назва застосунку.
2. Кнопка авторизації.

3. Кнопка реєстрації.
4. Ім'я користувача.
5. Електронна адреса.
6. Пароль.
7. Повтор пароля.
8. Повне ім'я.
9. Вік.
10. Досвід роботи з кодами Морзе.
11. Мета навчання.
12. Домінуючий стиль навчання.
13. Кнопка «Зареєструватися».
14. Футер.

Сторінка початкового тестування має на меті зібрати необхідні дані для формування персоналізованого підходу шляхом отримання відповідей на відповідні запитання про знання та побажання до навчання абетки Морзе. Структура інтерфейсу сторінки початкового тестування наведена на рисунку 3.5.

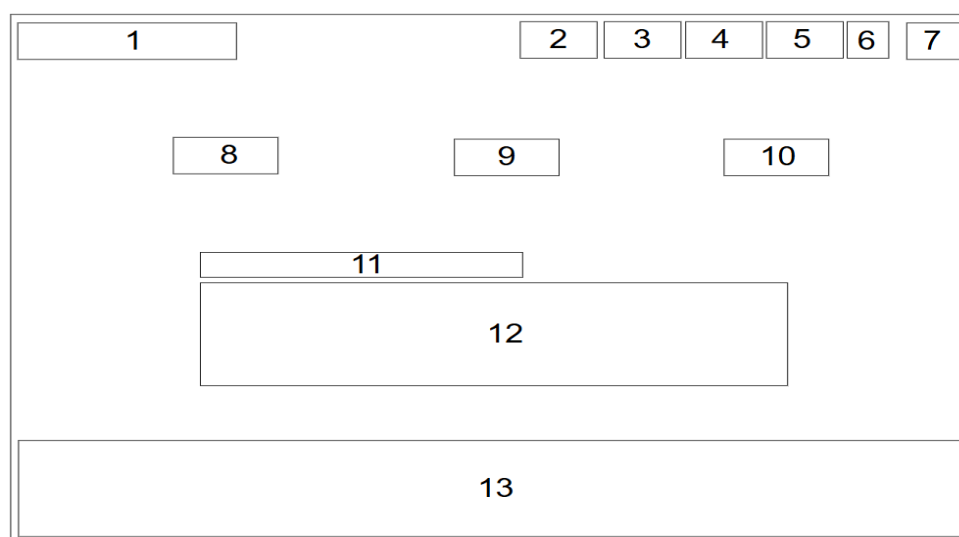


Рисунок 3.5 – Структурна схема інтерфейсу сторінки початкового тестування

Складові елементи інтерфейсу сторінки початкового тестування:

1. Назва застосунку.

2. Кнопка панелі інструментів.
3. Кнопка «Практика».
4. Кнопка «Сценарії».
5. Кнопка «Тестування».
6. Кнопка «Профіль».
7. Кнопка «Вийти з акаунту».
8. Етап тестування «Побажання щодо навчання».
9. Етап тестування «Попередній досвід».
10. Етап тестування «Цілі навчання».
11. Питання тесту.
12. Варіанти відповіді.
13. Футер застосунку.

Сторінка статистики відображає графіки та статистику, що відображає прогрес у навчанні, рівень володіння абеткою Морзе, рівень точності під час тестування та рівень помилок.

Структура інтерфейсу сторінки статистики наведена на рисунку 3.6.



Рисунок 3.6 – Структурна схема інтерфейсу сторінки статистики

Складові елементи інтерфейсу сторінки статистики:

1. Назва застосунку.
2. Кнопка панелі інструментів.
3. Кнопка «Практика».
4. Кнопка «Сценарії».
5. Кнопка «Тестування».
6. Кнопка «Профіль».
7. Кнопка «Вийти з акаунту».
8. Загальний прогрес навчання.
9. Кількість вивчених символів.
10. Кількість пройдених сценаріїв.
11. Час практикування.
12. Відсоток точності відповідей.
13. Діаграма методів навчання.
14. Рівень володіння символами.
15. Кнопка для дослідження іммерсивних сценаріїв.
16. Кнопка тестування.

Сторінка практики дозволяє перевіряти свої знання певних букв та звуків абетки Морзе. Користувач може обирати складність, швидкість програвання символів. Структура інтерфейсу цієї сторінки наведена на рисунку 3.7.

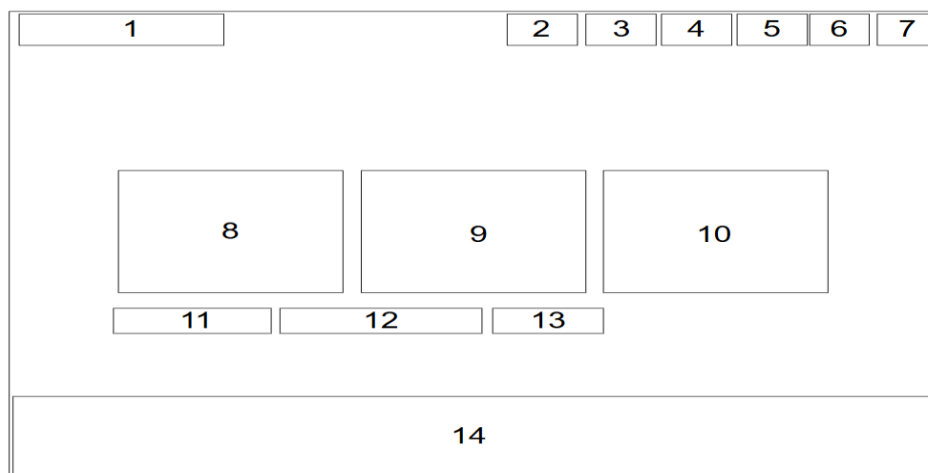


Рисунок 3.7 – Структурна схема інтерфейсу сторінки практики

Складові елементи інтерфейсу сторінки практики:

1. Назва застосунку.
2. Кнопка панелі інструментів.
3. Кнопка «Практика».
4. Кнопка «Сценарії».
5. Кнопка «Тестування».
6. Кнопка «Профіль».
7. Кнопка «Вийти з акаунту».
8. Меню «Розпізнавання букв».
9. Меню «Розпізнавання слів».
10. Меню «Формування кодів Морзе».
11. Рівень складності.
12. Швидкість програвання кодів Морзе.
13. Кнопка «Розпочати практику».
14. Футер сторінки.

Сторінка перевірки знань символів абетки Морзе містить простий інтерфейс, який дозволяє програвати звуки абетки Морзе та давати відповідь, яка буква була озвучена, або вводити символи абетки Морзе, які відповідають введеній букві чи числову. Структура інтерфейсу цієї сторінки наведена на рисунку 3.8.

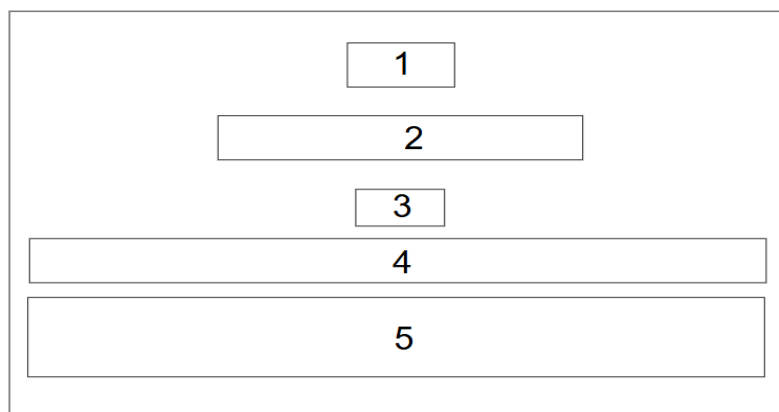


Рисунок 3.8 – Структурна схема інтерфейсу сторінки перевірки знань символів абетки Морзе

Складові елементи інтерфейсу сторінки перевірки знань символів абетки Морзе:

1. Кнопка «Програвання звуку».
2. Поле введення відповіді.
3. Кнопка «Відповісти».
4. Результат.
5. Статистика тестування.

Сторінка з переліком тестів містить тести для перевірки рівня знань абетки Морзе. Передбачається наявність різного характеру тестів та різного рівня знань. Структура інтерфейсу цієї сторінки наведена на рисунку 3.9.

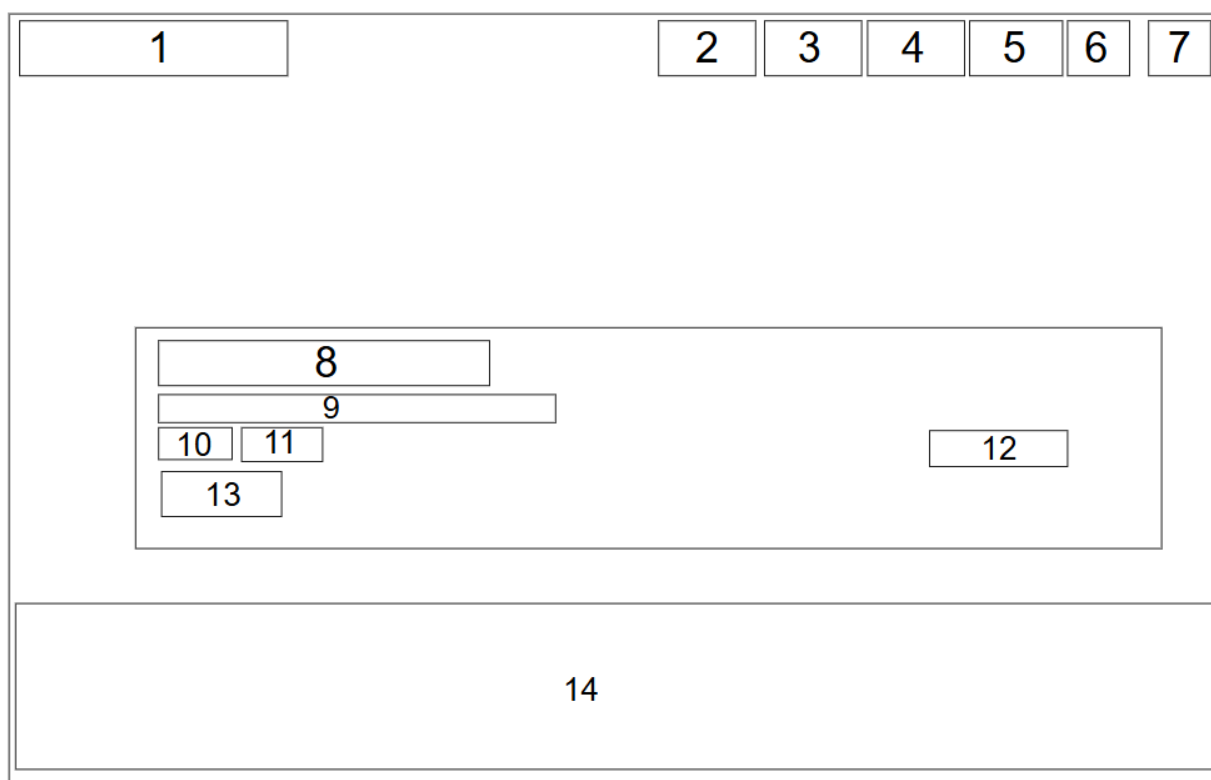


Рисунок 3.9 – Структурна схема інтерфейсу сторінки тестування навичок абетки Морзе

Складові елементи інтерфейсу сторінки тестування навичок абетки Морзе:

1. Назва застосунку.
2. Кнопка панелі інструментів.

3. Кнопка «Практика».
4. Кнопка «Сценарії».
5. Кнопка «Тестування».
6. Кнопка «Профіль».
7. Кнопка «Вийти з акаунту».
8. Назва тесту.
9. Опис тесту.
10. Рівень складності.
11. Кількість питань.
12. Приблизна тривалість тесту.
13. Кнопка «Розпочати тест».
14. Футер сторінки.

Таким чином, розроблено модулі системи для вивчення абетки Морзе. Описано алгоритм роботи системи, описано усі сторінки системи, що дозволяють користувачеві вивчати абетку Морзе, перевіряти свій рівень знань та практикуватися для підтримки навичок.

3.4 Висновки

У третьому розділі було проведено варіантний аналіз та обґрунтування вибору програмних засобів, обрано мову JavaScript для розробки системи для вивчення абетки Морзе, розроблено базу даних, що забезпечує збереження даних про користувача, тести та іншу супутню інформацію, розроблено модулі системи, реалізовано інтерфейс користувача та загальний алгоритм роботи системи.

4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Аналіз методів тестування

Тестування системи для вивчення абетки Морзе такого застосунку вимагає комплексного підходу, який охоплює як функціональні, так і нефункціональні вимоги різних компонентів. Кожен окремий компонент, від генератора аудіосигналів Морзе до алгоритмів аналізу патернів помилок, потребує ретельного ізольованого тестування. Особливої уваги вимагають дії та редуктори Redux, оскільки вони керують критичними переходами стану, що впливають на весь процес роботи застосунку.

Інтеграційне тестування необхідне для перевірки взаємодії між компонентами сторони клієнта та сервера. Кінцеві точки RESTful API застосунку повинні надійно обробляти автентифікацію, відстеження прогресу користувача та надсилання даних.

Інтерфейс користувача потребує тестування в кількох вимірах. Автоматизовані інструменти тестування повинні перевіряти базові можливості відображення та взаємодії, тоді як людські тестувальники мають оцінювати суб'єктивні аспекти досвіду. Інтерфейс повинен належним чином адаптуватися до різних розмірів пристроїв, підтримувати стандарти доступності та чітко представляти інформацію в усіх навчальних модулях.

Особливо важливим є тестування аудіокомпонентів, оскільки застосунок значною мірою покладається на точне генерування звуків абетки Морзе та часові параметри. Тести повинні перевіряти, що відтворення аудіо функціонує правильно в різних браузерях та пристроях, з правильним часуванням для крапок, тире та пауз при різних налаштуваннях швидкості. Це тестування має включати перевірку, що механізм аудіо-зворотного зв'язку точно відповідає візуальним представленням, коли вони представлені одночасно.

Важливим аспектом є перевірка матриць помилок та можливостей розпізнавання патернів, яка повинна відбуватися в різних часових масштабах, щоб забезпечити належне функціонування як короткострокових адаптацій, так і

довгострокових коригувань навчального шляху.

Отже, система для вивчення абетки Морзе потребує належного тестування, яке охоплює її унікальні особливості разом із звичайними проблемами якості програмного забезпечення.

4.2 Тестування вебзастосунку для вивчення абетки Морзе

Тестування розпочинається із головної сторінки системи, яка відображається при першому візиті користувача. Вона містить загальну інформацію про застосунок, його призначення (допомога у вивченні абетки Морзе з персоналізованим підходом) та можливості. Кнопка “Get started” дозволяє перейти до наступної сторінки для реєстрації користувача. Головна сторінка системи наведена на рисунку 4.1.

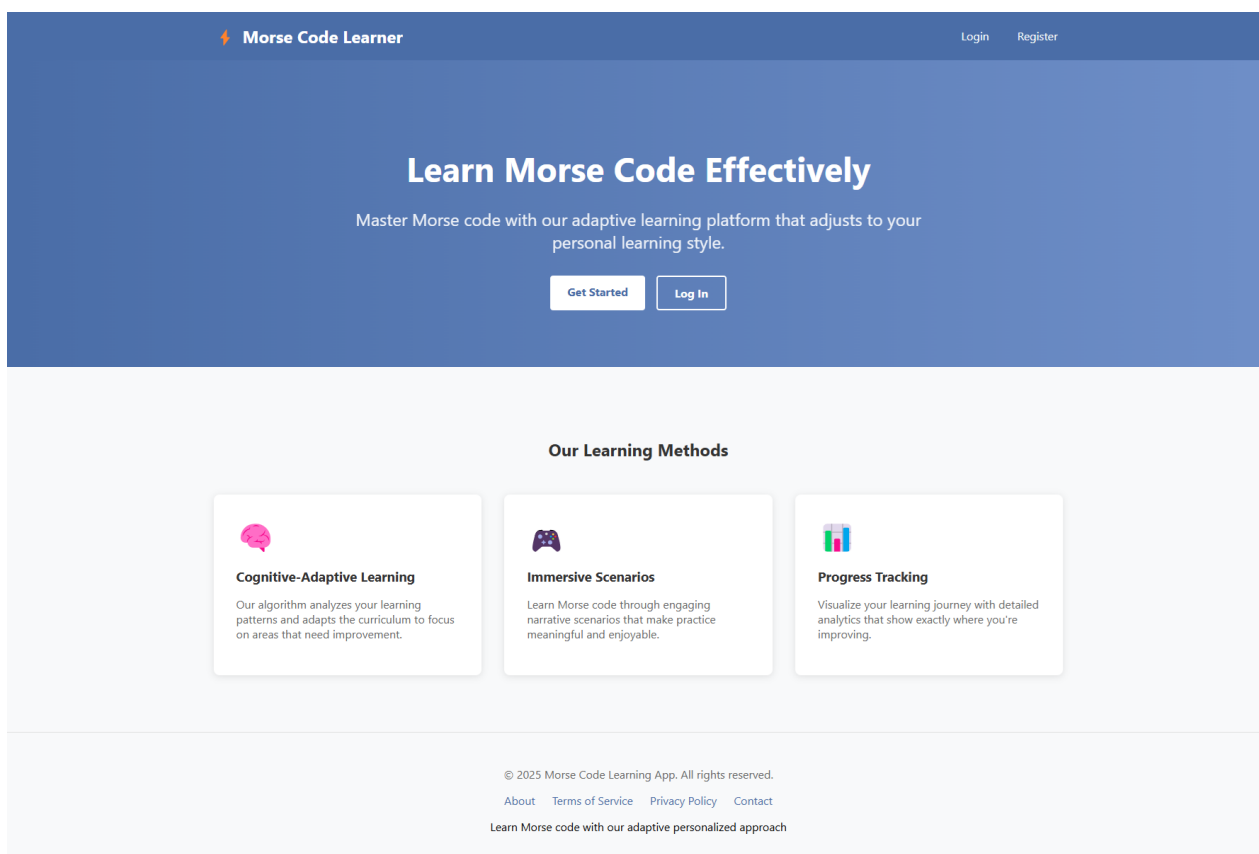


Рисунок 4.1 – Головна сторінка вебзастосунку

Щоб продовжити роботу з системою, натискаємо на кнопку «Get started»,

яка знаходиться в центрі головної сторінки. Після натискання відбувається перенаправлення на сторінку реєстрації, де треба заповнити персональні дані. На сторінці реєстрації міститься форма з декількома полями: ім'я, електронна пошта та пароль. Система вимагає ввести надійний пароль, щоб забезпечити безпеку облікового запису, а повідомлення про помилки виділяються червоним кольором, якщо дані введені неправильно. Сторінка реєстрації користувача продемонстрована на рисунку 4.2.

The screenshot displays the 'Create Your Account' registration form for the 'Morse Code Learner' application. The form is organized into several sections: 'Account Information', 'Personal Information', and 'Learning Preferences'. The 'Account Information' section includes fields for Username (filled with 'Alina'), Email (filled with 'alina.kremenichuk@gmail.com'), Password (masked with dots), and Confirm Password (also masked). The 'Personal Information' section includes fields for Full Name (filled with 'Alina Kremenichuk'), Age (filled with '21'), Previous Morse Code Experience (a dropdown menu set to 'None'), and Learning Goal (a dropdown menu set to 'Hobby / Personal Interest'). The 'Learning Preferences' section includes a 'Preferred Learning Method' section with three radio buttons: 'Visual (seeing)' (selected), 'Auditory (hearing)', and 'Kinesthetic (doing)'. At the bottom of the form is a blue button labeled 'Register & Continue to Diagnostic Test'. The form is set against a light gray background with a white border. The top of the page features a dark blue header with the 'Morse Code Learner' logo and 'Login' and 'Register' links. The bottom of the page has a light gray footer with copyright information and links to 'About', 'Terms of Service', 'Privacy Policy', and 'Contact'.

Morse Code Learner Login Register

Create Your Account

Account Information

Username
Alina

Email
alina.kremenichuk@gmail.com

Password
.....

Confirm Password
.....

Personal Information

Full Name
Alina Kremenichuk

Age
21

Previous Morse Code Experience
None

Learning Goal
Hobby / Personal Interest

Learning Preferences

Preferred Learning Method

☒ Visual (seeing) ☐ Auditory (hearing) ☐ Kinesthetic (doing)

Register & Continue to Diagnostic Test

© 2025 Morse Code Learning App. All rights reserved.
[About](#) [Terms of Service](#) [Privacy Policy](#) [Contact](#)

Рисунок 4.2 – Сторінка реєстрації користувача

Після успішної реєстрації система автоматично перенаправляє нас на сторінку початкового тестування, де відбувається оцінка рівня знань користувача про абетку Морзе. Це дуже важлива частина системи, бо саме тут збираються необхідні данні для створення індивідуальної програми навчання. Перший етап тестування фокусується на навчальних перевагах користувача. Тут містяться декілька питань з варіантами відповідей про те, який спосіб навчання більше підходить - візуальний, аудіальний чи комбінований. Також є питання про те, скільки часу користувач планує приділяти навчанню щоденно та яким чином він краще запам'ятовує інформацію. Сторінка початкового тестування рівня знань абетки Морзе користувача етапу «навчальні переваги» показана на рисунку 4.3.

Morse Code Learner Dashboard Practice Scenarios Testing Profile Logout

Learning Profile Assessment

1 Learning Preferences 2 Prior Experience 3 Learning Goals

Learning Preferences

How do you prefer to learn new information?

☐ By seeing (visual diagrams, charts, text)

☐ By hearing (audio explanations, discussions)

☐ By doing (hands-on practice, interactive exercises)

How often can you dedicate time to practice Morse code?

☐ Daily (10+ minutes)

☐ Several times a week

☐ Once a week

☐ Occasionally

Skip Assessment Next

© 2025 Morse Code Learning App. All rights reserved.
[About](#) [Terms of Service](#) [Privacy Policy](#) [Contact](#)
Learn Morse code with our adaptive personalized approach

Рисунок 4.3 – Сторінка початкового тестування рівня знань абетки Морзе користувача (навчальні переваги)

По завершенню першого етапу тестування і натисканню кнопки "Далі", система автоматично переходить до другого етапу. Цей етап схожий на попередній за структурою, але питання тут спрямовані на визначення досвіду використання абетки Морзе. Система запитує, чи має користувач попередній досвід роботи з абеткою Морзе, які символи уже знає, чи використовував він абетку Морзе на практиці. Сторінка другого етапу тестування показана на рисунку 4.4.

Morse Code Learner Dashboard Practice Scenarios Testing Profile Logout

Learning Profile Assessment

1 Learning Preferences 2 Prior Experience 3 Learning Goals

Prior Experience

Have you had any previous experience with Morse code?

☐ None

☐ A little (recognize a few symbols)

☐ Some (can send/receive basic messages)

☐ Experienced (proficient in Morse code)

Do you have any musical background or training?

☐ None

☐ A little (casual listener)

☐ Some (played an instrument)

☐ Extensive (formal musical training)

Back Next

© 2025 Morse Code Learning App. All rights reserved.
[About](#) [Terms of Service](#) [Privacy Policy](#) [Contact](#)
Learn Morse code with our adaptive personalized approach

Рисунок 4.4 – Сторінка початкового тестування рівня знань абетки Морзе користувача (попередній досвід)

Завершивши другий етап і натиснувши кнопку "Далі", переходимо до третього, останнього етапу тестування – визначення мети навчання. На цьому етапі система задає питання про те, навіщо користувач вивчає абетку Морзе: для професійного використання, для хоббі, для участі в змаганнях чи з інших причин. Також система запитує про те, які конкретно навички необхідно розвинути - швидкість розпізнавання символів, здатність передавати повідомлення, розуміти довгі тексти і т.д. Є також питання про те, який рівень майстерності необхідно досягти і за який час. Останній етап тестування представлений на рисунку 4.5.

The screenshot shows the 'Morse Code Learner' application interface. At the top is a blue navigation bar with the app name and links to Dashboard, Practice, Scenarios, Testing, Profile, and Logout. Below this is a section titled 'Learning Profile Assessment' with three numbered steps: 1. Learning Preferences, 2. Prior Experience, and 3. Learning Goals. The 'Learning Goals' step is active and contains two sections of radio button questions. The first section asks for the primary goal for learning Morse code, with options: Personal interest/hobby, Ham radio operation, Professional requirement, and Emergency preparedness. The second section asks for the level of proficiency aiming to achieve, with options: Basic (recognize common letters/numbers), Intermediate (send/receive simple messages), Advanced (fluent communication), and Expert (high-speed operation). At the bottom of the form are 'Back' and 'Complete Assessment' buttons. The footer contains copyright information for 2025 Morse Code Learning App, links to About, Terms of Service, Privacy Policy, and Contact, and a statement about the adaptive personalized approach.

Morse Code Learner Dashboard Practice Scenarios Testing Profile Logout

Learning Profile Assessment

1 Learning Preferences 2 Prior Experience 3 Learning Goals

Learning Goals

What is your primary goal for learning Morse code?

☐ Personal interest/hobby

☐ Ham radio operation

☐ Professional requirement

☐ Emergency preparedness

What level of proficiency are you aiming to achieve?

☐ Basic (recognize common letters/numbers)

☐ Intermediate (send/receive simple messages)

☐ Advanced (fluent communication)

☐ Expert (high-speed operation)

Back Complete Assessment

© 2025 Morse Code Learning App. All rights reserved.
[About](#) [Terms of Service](#) [Privacy Policy](#) [Contact](#)
Learn Morse code with our adaptive personalized approach

Рисунок 4.5 – Сторінка початкового тестування рівня знань абетки Морзе користувача (мета навчання)

Після того, як користувач відповів на всі питання тестування та натиснув кнопку "Завершити", система обробляє відповіді і перенаправляє на інформаційну панель. Ця сторінка містить детальну статистику про прогрес у вивченні абетки Морзе. Тут можна побачити загальний рівень знань, виражений у відсотках, а також діаграми, які показують, наскільки добре користувач володіє різними категоріями символів - літерами, цифрами та спеціальними знаками. На цій сторінці представлені графіки активності за останній тиждень, місяць та рік, що дозволяє відслідковувати динаміку навчання. Сторінка статистики вивчення абетки Морзе представлена на рисунку 4.6.

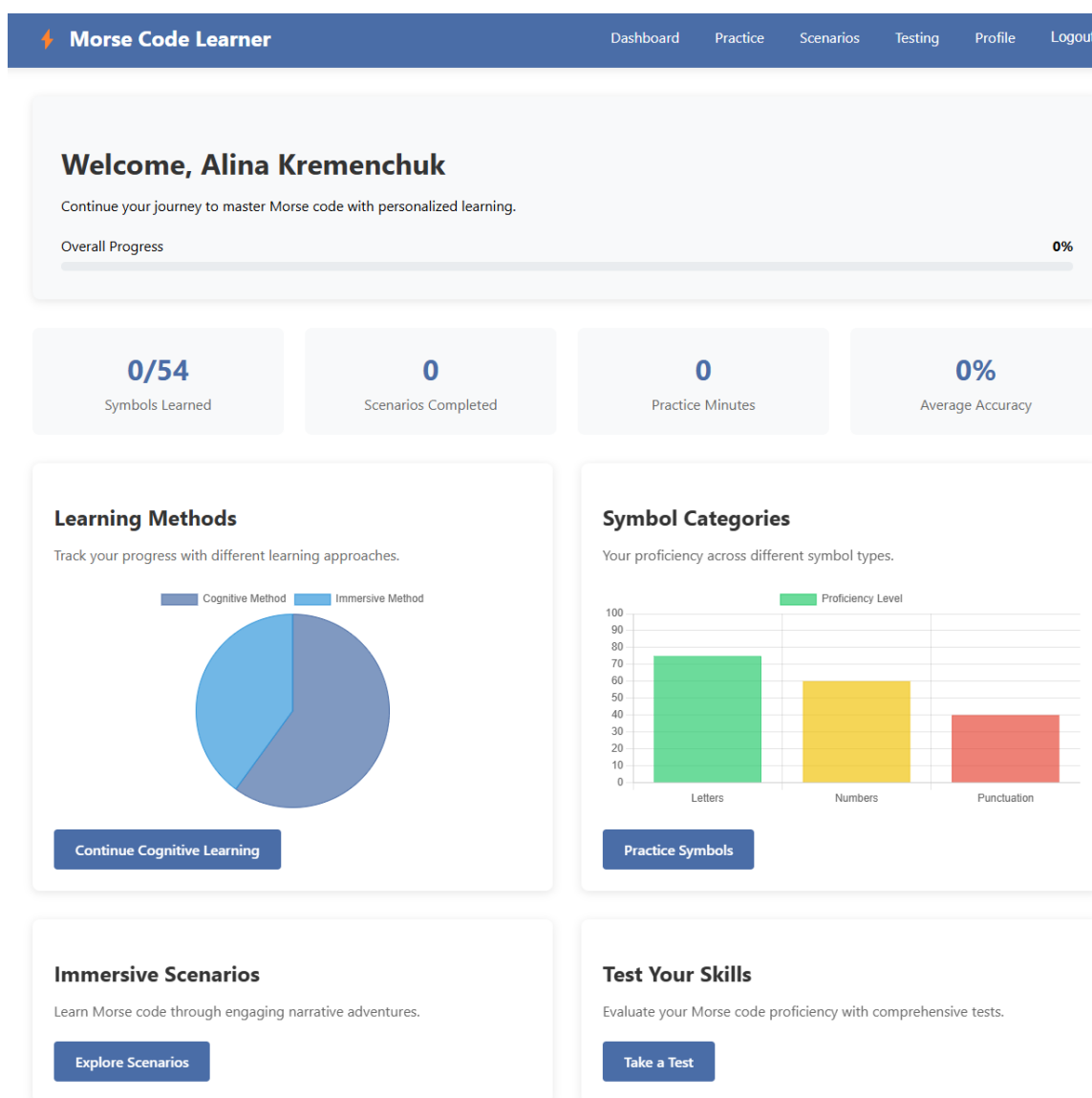


Рисунок 4.6 – Сторінка статистики вивчення абетки Морзе

З інформаційної панелі можна перейти на сторінку практики, де безпосередньо відбувається вивчення абетки Морзе. Ця сторінка пропонує декілька режимів для тренування різних навичок. Можна обрати режим розпізнавання окремих літер, де система відтворює звуковий сигнал, а користувач повинен вгадати відповідну літеру. Також є режим для вивчення слів, де потрібно розпізнати цілі слова, закодовані абеткою Морзе. Крім того, є режим самостійного введення кодів Морзе, де користувачу показують літеру чи слово, а він повинен ввести відповідний код. На сторінці є також можливість налаштувати складність завдань - від початкового рівня до експертного, з різною швидкістю відтворення сигналів та різним набором символів. Сторінка практики роботи з кодами Морзе показана на рисунку 4.7.

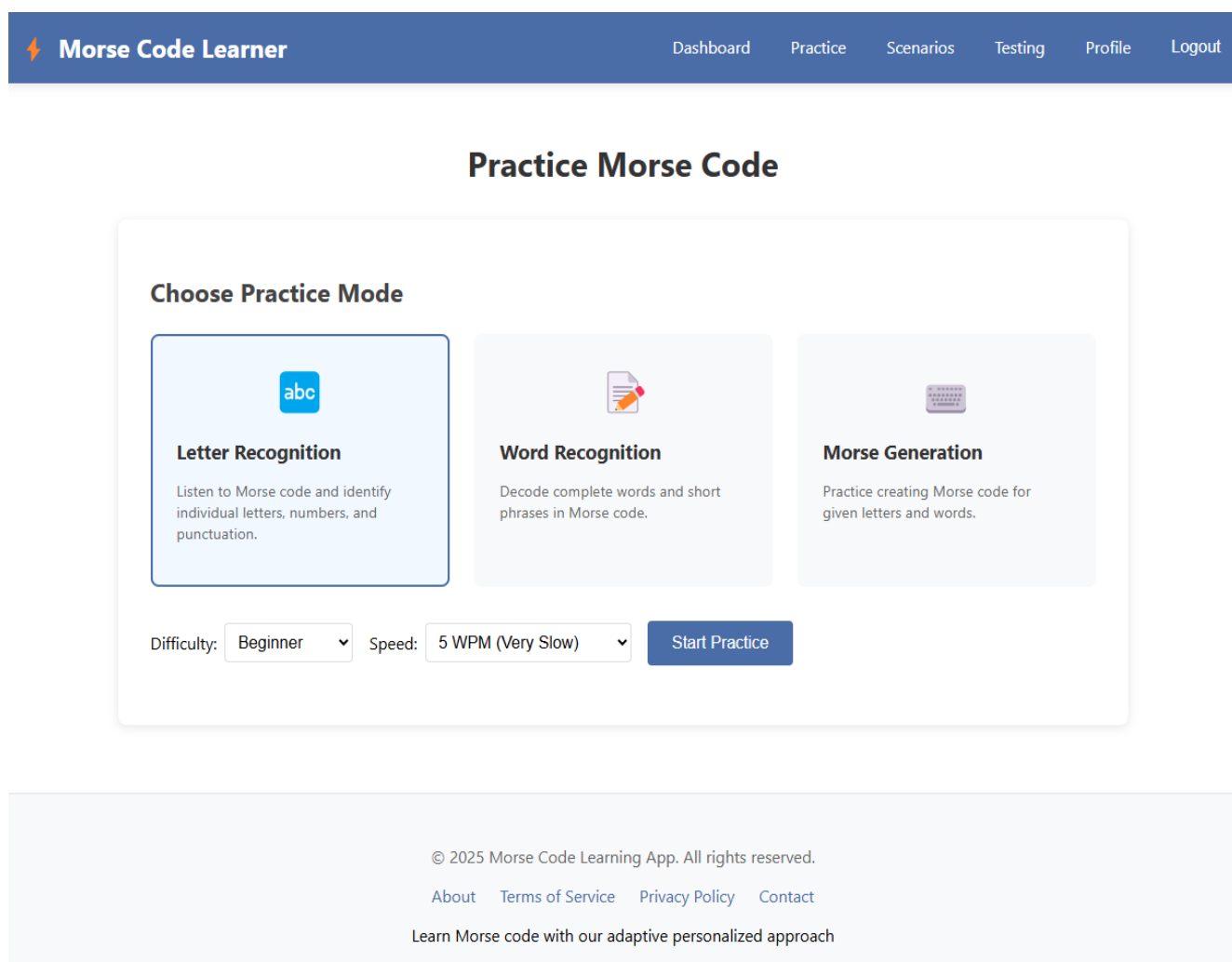


Рисунок 4.7 – Сторінка практики роботи з кодами Морзе

Коли користувач обирає один з режимів практики, наприклад, режим розпізнавання символів, система переносить користувача на відповідну сторінку тестування. Ця сторінка має дуже простий та зрозумілий інтерфейс, щоб користувач міг сконцентруватися на завданні. У центрі екрану є кнопка для прослуховування звукового сигналу Морзе, яку можна натиснути кілька разів, якщо потрібно повторно прослухати символ. Нижче розташоване поле для введення відповіді - літери чи символу, який був закодований. Поруч з полем є кнопка "Підтвердити", яка дозволяє відправити відповідь на перевірку. Також на цій сторінці є індикатор прогресу, який показує, скільки завдань вже виконано і скільки залишилося до кінця поточної сесії. Сторінка тестування знань символів абеткою Морзе показана на рисунку 4.8.

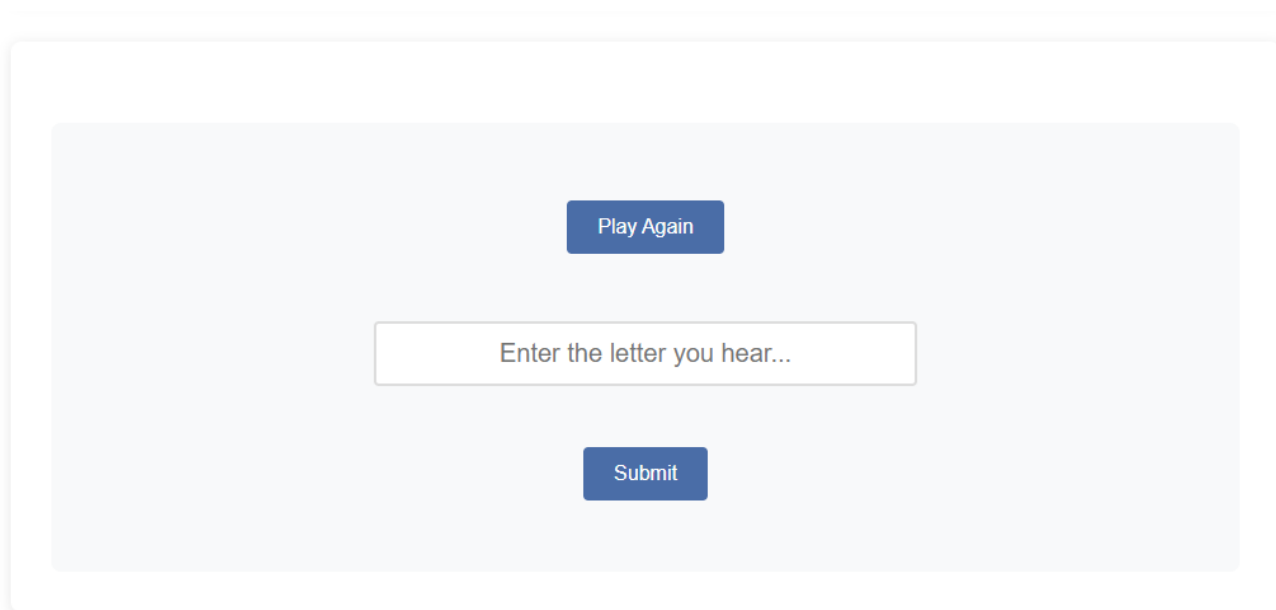


Рисунок 4.8 – Сторінка тестування знань символів абетки Морзе

Під час проходження тестування система одразу повідомляє, чи правильною була відповідь. Якщо користувач припустився помилки, система показує правильну відповідь і відтворює звуковий сигнал ще раз, щоб користувач міг запам'ятати правильний варіант. На рисунку 4.9 можна побачити, як виглядає така відповідь системи. Тут міститься повідомлення про те, що відповідь була неправильною, а також правильний варіант з візуальним представленням коду

Морзе. Після завершення всіх завдань в поточній сесії, система показує загальний результат тестування – кількість правильних відповідей, відсоток успішності та рекомендації щодо подальшого навчання. Також на цьому екрані є кнопка для переходу до наступної сесії або повернення на інформаційну панель. Відповідь застосунку при тестуванні та результат практики представлені на рисунку 4.9.

The screenshot displays a user interface for a Morse code practice application. At the top, there is a blue button labeled "Play Again". Below it is a white rectangular box containing the letter "N". Underneath the box is another blue button labeled "Submit". A pink horizontal bar contains the text "Incorrect. The correct answer is H." in red. Below this bar, the section "Practice Session Summary" is shown, with the following statistics: "Symbols practiced: 3" and "Correct answers: 0".

Рисунок 4.9 – Відповідь застосунку при тестуванні та результат практики

Окрім режиму практики, система також пропонує різноманітні тести для перевірки знань абетки Морзе. На сторінці тестів, яка показана на рисунку 4.10, розміщено список доступних тестів, розподілених за рівнем складності – від початкового до експертного. Для кожного тесту вказана кількість питань, очікувана тривалість проходження та короткий опис того, які навички перевіряються. Наприклад, є тести на розпізнавання літер, цифр, спеціальних символів, а також комплексні тести, які перевіряють здатність розпізнавати цілі слова та речення. Деякі тести мають обмеження за часом, що імітує реальні умови використання абетки Морзе. На цій сторінці також є фільтри, які дозволяють відсортувати тести за різними параметрами, наприклад, за тривалістю або за типом символів. Сторінка тестів для абетки Морзе представлена на рисунку 4.10.

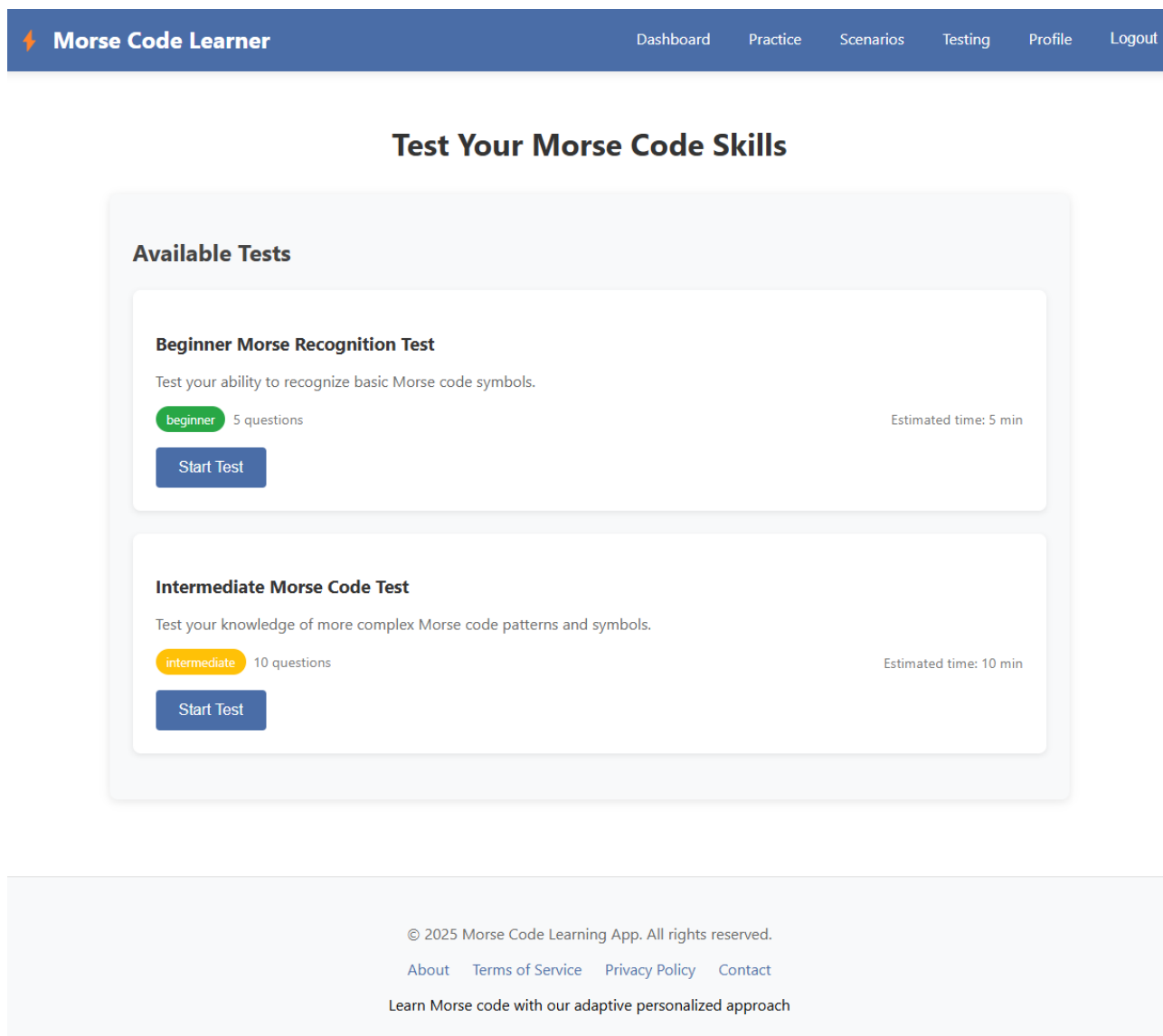


Рисунок 4.10 – Сторінка тестів для абетки Морзе

На сторінці імерсивних сценаріїв представлено три основні сценарії різного рівня складності. Кожен сценарій має власну візуальну ідентичність та контекстуальне оформлення, що створює атмосферу відповідної ситуації. Перший сценарій "Maritime Rescue Mission" призначений для початківців та моделює морську рятувальну операцію. Другий сценарій "Space Station Emergency" має середній рівень складності та відтворює екстрену ситуацію на космічній станції. Третій сценарій "Underground Mine Emergency" є найскладнішим та моделює аварійну ситуацію в підземній шахті.

Сторінка імерсивних сценаріїв продемонстрована на рисунку 4.11.

Morse Code Practice Scenarios

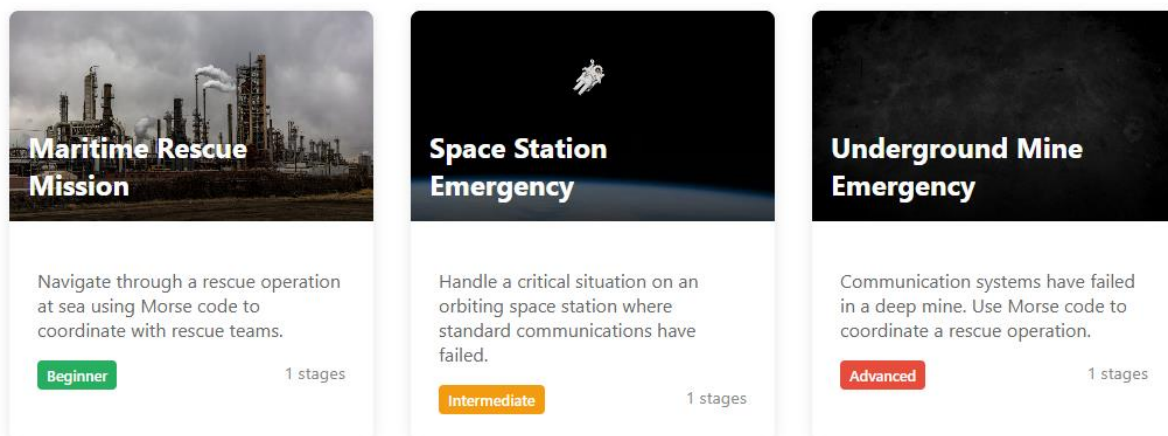


Рисунок 4.11 – Сторінка імєрсивних сценаріїв навчання

При виборі сценарію система переносить користувача на спеціальну сторінку з детальним описом місії. На рисунку 4.12 показано початок сценарію "Maritime Rescue Mission", де користувач отримує контекстуальну інформацію про свою роль та поточну ситуацію. Сторінка оформлена у відповідному морському стилі з фоновим зображенням промислового порту, що створює атмосферу реальної рятувальної операції. У верхній частині відображається назва місії та її опис, а нижче представлені цілі етапу навчання.

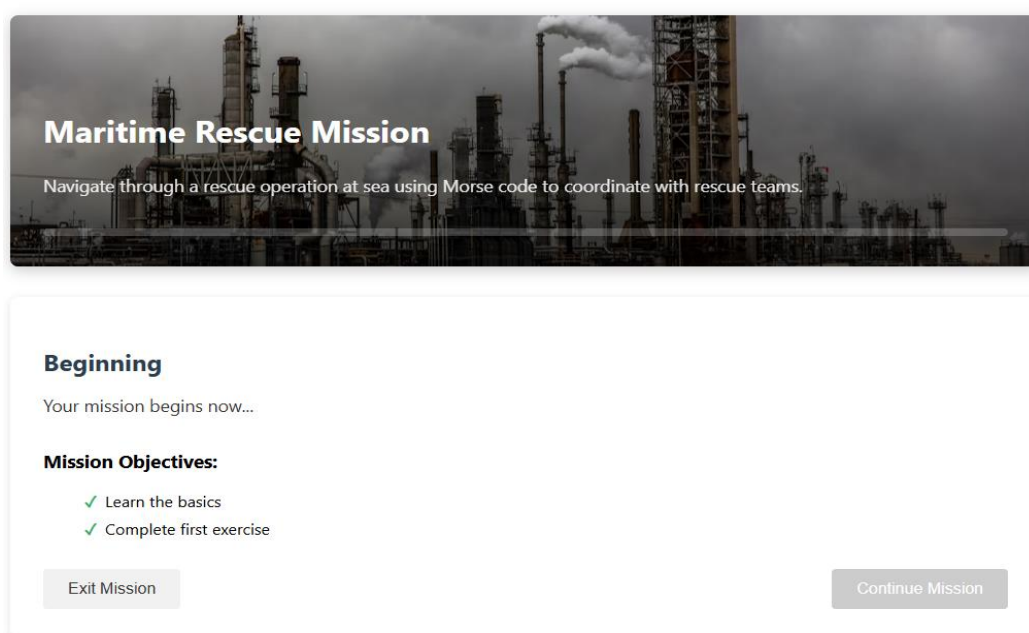


Рисунок 4.12 – Початкова сторінка сценарію морської рятувальної місії

Після ознайомлення з початковою інформацією користувач переходить до активної фази сценарію, показаної на рисунку 4.13. Тут представлено конкретне завдання - розшифрування сигналу лиха від човна в біді. Сторінка містить детальний наратив ситуації, список цілей місії та інтерактивну вправу з азбукою Морзе. Користувач може прослухати звуковий сигнал, налаштувати швидкість відтворення та ввести свою відповідь. Інтерфейс розроблено таким чином, щоб максимально занурити користувача в контекст рятувальної операції.

Maritime Rescue Mission

Navigate through a rescue operation at sea using Morse code to coordinate with rescue teams.

Distress Call

You are a radio operator on a coast guard vessel. A distress signal has been detected from a small boat in trouble.

Mission Objectives:

- ✓ Decode the incoming distress signal
- ✓ Determine the boat's coordinates
- ✓ Send a confirmation response

Distress Signal

Listen to the Morse code and decode the message:

Play Morse Code

Speed: 12 WPM

Enter your decoded message:

Рисунок 4.13 – Активна фаза сценарію з завданням декодування

При успішному виконанні завдання система відображає результат та нагороди, як показано на рисунку 4.14. Користувач отримує підтвердження правильності відповіді та інформацію про зароблені очки досвіду і бейджі. Система також показує блок "Mission Accomplished!" з деталями досягнень. Це створює відчуття прогресу та мотивує до продовження навчання. Кнопка "Complete Mission" дозволяє завершити поточний етап та перейти до наступного рівня сценарію.

Distress Signal

Listen to the Morse code and decode the message:

Play Morse Code

Speed: 12 WPM

Enter your decoded message:

SOS SOS BOAT IN DISTRESS

Submit

Correct! You decoded the message successfully.

Mission Accomplished!

You've earned:

+50 XP First Response

Continue to the next stage of your mission!

Exit Mission Complete Mission

Рисунок 4.14 – Результат виконання завдання та отримані нагороди

Імерсивні сценарії демонструють інноваційний підхід до навчання азбуки Морзе, поєднуючи практичні навички з контекстуальним навчанням. Такий метод

дозволяє користувачам не просто запам'ятовувати символи, а розуміти їх практичне застосування в реальних ситуаціях, що значно підвищує ефективність навчального процесу.

Таким чином, було проведено тестування системи для вивчення абетки Морзе. Продемонстровано її основний функціонал, типові приклади використання. Доведено працездатність системи. Розроблена система виконує поставлені на початку розробки завдання.

4.3 Висновки

У четвертому розділі було проведено аналіз методів тестування системи для вивчення абетки Морзе. Було проведено її тестування, продемонстровано основний функціонал, наведено приклади використання, доведено, що розроблена система виконує поставлені на початку розробки завдання.

ВИСНОВКИ

У результаті виконання бакалаврської кваліфікаційної роботи було розроблено вебзастосунок для вивчення абетки Морзе. Розроблений застосунок дозволяє вивчати символи та звуки з абетки Морзе, проходити тестування для перевірки вивчених символів, проходити практику у сценаріях, близьких до реального застосування, проходити повторення для відновлення забутих символів. Бакалаврську кваліфікаційну роботу реалізовано згідно методичних вказівок [22].

Вебзастосунок розроблено мовою JavaScript, середовищем розробки Visual Studio Code, фреймворку React.js та Node.js.

У першому розділі було проведено аналіз сучасного стану застосунків для вивчення абетки Морзе, доведено актуальність розробки такого застосунку. Проведено порівняльний аналіз аналогів з розроблюваним вебзастосунком, продемонстровано його переваги та таким чином доведено доцільність розробки. Проведено аналіз методів розв'язання задачі, виконано постановку задач дослідження з розробки власного вебзастосунку.

У другому розділі було проведено аналіз інформаційного забезпечення системи для вивчення абетки Морзе, описано дані, які у ній використовуються та формуються. Розроблено архітектуру системи за клієнт-серверним підходом, описано складові архітектури. Розроблено метод когнітивно-адаптивної прогресії з аналізом патернів помилок та метод навчання з імерсивними контекстними сценаріями, побудовано блок-схеми алгоритмів складових цих методів.

У третьому розділі було проведено варіантний аналіз та обґрунтування вибору програмних засобів, обрано мову JavaScript для розробки системи. Розроблено базу даних та модулі системи, реалізовано інтерфейс користувача та загальний алгоритм роботи системи.

У четвертому розділі було проведено аналіз методів тестування системи для вивчення абетки Морзе. Було проведено її тестування, продемонстровано основний функціонал, наведено приклади використання, доведено, що розроблена система виконує поставлені на початку розробки завдання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Morse codes in modern aviation [Електронний ресурс] – Режим доступу до ресурсу: aviation.stackexchange.com
2. А.І. Кременчук, Г.Б. Ракитянська. Вебзастосунок для вивчення абетки Морзе на основі адаптивного персоналізованого підходу / Матеріали LIV Всеукраїнської науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2025) : збірник доповідей [Електронний ресурс] – Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24124/19878>
3. Building a modern web application [Електронний ресурс] – Режим доступу до ресурсу: <https://www.zend.com/blog/modern-web-application>
4. What is adaptive learning? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.montclair.edu/itds/digital-pedagogy/pedagogical-strategies-and-practices/adaptive-learning/>
5. Exploring modern web app technologies [Електронний ресурс] – Режим доступу до ресурсу: <https://tech-stack.com/blog/modern-application-development/>
6. What are cloud services? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hpe.com/us/en/what-is/cloud-services.html>
7. Peng, H., Ma, S. & Spector, J.M. Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment. Smart Learn. Environ. 6, 9 (2019). <https://doi.org/10.1186/s40561-019-0089-y>
8. Cross browser audio basics [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/Media/Guides/Audio_and_video_delivery/Cross-browser_audio_basics
9. Integrating machine learning into web development [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@bartzalewski/integrating-machine-learning-into-web-development-f2cacd131b76>
10. Morse-it [Електронний ресурс] – Режим доступу до ресурсу:

<https://pacolabs.com/iOS/MorseIt/>

11. Morse Mania [Електронний ресурс] – Режим доступу до ресурсу: <https://play.google.com/store/apps/details?id=net.countrymania.morse&hl=uk>

12. Morse Code Flash Cards [Електронний ресурс] – Режим доступу до ресурсу: <https://apps.apple.com/ua/app/morse-code-flashcards/id6447498786?l=uk>

13. Casimir Saternos. Client-Server Web Apps with JavaScript and Java: Rich, Scalable, and RESTful. Farnham: O'Reilly Media, 2021. 260с.

14. Prabhu Eshwarla. Rust Servers, Services, and Apps. Нью-Йорк: Manning, 2023. 328с.

15. Maximilian Schwarzmüller. Progressive Web Apps (PWA): The Complete Guide. Birmingham: Packt Publishing, 2018. 320с.

16. Web development with essential libraries [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@kirushikanketheeswaran/web-development-with-essential-libraries-15d730fb735f>

17. Mark Richards. Fundamentals of Software Architecture. Farnham: O'Reilly Media, 2020. 422с.

18. Douglas Crockford. JavaScript: The Good Parts: The Good Parts. Farnham: O'Reilly Media, 2008. 172с.

19. Luciano Ramalho. Fluent Python: Clear, Concise, and Effective Programming. Farnham: O'Reilly Media, 2015. 790с.

20. Jon Duckett. PHP & MySQL: Server-side Web Development. Гобокен: Wiley, 2022. 671с.

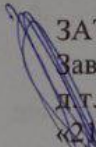
21. Edward Sciore. Database Design and Implementation. Гобокен: Wiley, 2008. 784с.

22. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 121 "Інженерія програмного забезпечення" / Уклад. О.Н. Романюк, Г.О. Черноволик – Вінниця: ВНТУ, 2022. – 52с.

ДОДАТКИ

Додаток А – Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

 ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., проф. О.Н. Романюк
«21» березня 2025 р.

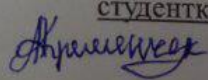
Технічне завдання
на бакалаврську кваліфікаційну роботу «Розробка вебзастосунку для
вивчення абетки Морзе на основі адаптивного персоналізованого підходу»
за спеціальністю
121 – Інженерія програмного забезпечення

Керівник бакалаврської кваліфікаційної роботи:

 к.т.н., доц. каф. ПЗ Ракитянська Г.Б.

«21» березня 2025 року.

Виконала:

 студентка гр. ІПІ-216 Кременчук А.І.

«21» березня 2025 року.

м. Вінниця – 2025 рік

1. Найменування та галузь застосування

Бакалаврська кваліфікаційна робота: «Розробка вебзастосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу».

Галузь застосування – вебтехнології.

2. Підстава для розробки.

Завдання на роботу, яке затверджене на засіданні кафедри програмного забезпечення – протокол №18 від 18 лютого 2025 р.

3. Мета та призначення розробки.

Метою роботи є покращення процесу вивчення абетки Морзе шляхом розробки спеціалізованого програмного забезпечення, що використовує адаптивний персоналізований підхід.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись БКР.

1. Building a modern web application [Електронний ресурс] — Режим доступу до ресурсу: <https://www.zend.com/blog/modern-web-application>
2. Peng, H., Ma, S. & Spector, J.M. Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment. Smart Learn. Environ. 6, 9 (2019). <https://doi.org/10.1186/s40561-019-0089-y>
3. Mark Richards. Fundamentals of Software Architecture. Farnham: O'Reilly Media, 2020. 422с.
4. Edward Sciore. Database Design and Implementation. Гобокен: Wiley, 2008. 784с.

5. Технічні вимоги

Вхідні дані – інформація про користувача, рівень знань абетки Морзе; вихідні дані – графіки зі статистикою, тести абетки Морзе; середовище розробки – Visual Studio Code; мова програмування – JavaScript; операційна система – кросплатформний додаток.

6. Конструктивні вимоги.

Інтерфейс програми повинен відповідати естетичним та ергономічним

вимогам, повинна бути зручною в обслуговуванні та керуванні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до бакалаврської кваліфікаційної роботи;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз стану розвитку застосунку для вивчення абетки Морзе	25.03.25 – 04.04.25
2	Розробка структури та алгоритмів програмного застосунку	05.04.25 – 18.04.25
3	Варіантний аналіз та вибір мови програмування розробки	19.04.25 – 22.04.25
4	Розробка застосунку для вивчення абетки Морзе	23.04.25 – 17.05.25
5	Тестування програмного забезпечення	18.05.25 – 25.05.25
6	Оформлення матеріалів до захисту БКР	26.05.25 – 30.05.25

10. Порядок контролю та прийняття.

Усі етапи бакалаврської роботи контролюються науковим керівником згідно плану по виконанню роботи. Прийняття бакалаврської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком захисту.

Додаток Б – Протокол перевірки на наявність текстових запозичень

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Розробка вебзастосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу»

Тип роботи: БКР

Підрозділ: кафедра програмного забезпечення, ФІТКІ

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 3.3 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

■ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

□ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

□ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

_____ (прізвище, ініціали, посада)

_____ (підпис)

_____ (прізвище, ініціали, посада)

_____ (підпис)

Особа, відповідальна за перевірку _____

Черноволик Г. О.

З висновком експертної комісії ознайомлений(-на)

Керівник _____

(підпис)

Ракитянська Г.Б. к.т.н., доцент каф. ПЗ

(прізвище, ініціали, посада)

Здобувач _____

(підпис)

Кременчук А.І.

(прізвище, ініціали)

Додаток В – Лістинг програми

```
import React, { useEffect } from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import { Provider } from 'react-redux';
import store from './redux/store';
import { checkAuth } from './redux/actions/authActions';

// Components
import Header from './components/common/Header';
import Footer from './components/common/Footer';
import LandingPage from './components/common/LandingPage';
import Register from './components/auth/Register';
import Login from './components/common/Login';
import Dashboard from './components/dashboard/Dashboard';
import SymbolLearning from './components/learning/SymbolsLearning';
import CognitiveProgress from './components/learning/CognitiveProgress';
import ImmersiveScenario from './components/scenarios/ImmersiveScenario';
import PracticeModule from './components/training/PracticeModule';
import TestingModule from './components/training/TestingModule';
import ProfileSettings from './components/user/ProfileSettings';
import PrivateRoute from './components/common/PrivateRoute';
import Diagnostic from './components/diagnostic/Diagnostic';
import ScenariosPage from './components/scenarios/ScenariosPage';

import './App.css';

function App() {
  useEffect(() => {
    store.dispatch(checkAuth());
```

```
}, []);
```

```
return (
```

```
  <Provider store={store}>
```

```
    <Router>
```

```
      <div className="app-container">
```

```
        <Header />
```

```
        <main className="main-content">
```

```
          <Routes>
```

```
            <Route path="/" element={<LandingPage />} />
```

```
            <Route path="/register" element={<Register />} />
```

```
            <Route path="/login" element={<Login />} />
```

```
            <Route path="/dashboard" element={
```

```
              <PrivateRoute>
```

```
                <Dashboard />
```

```
              </PrivateRoute>
```

```
            } />
```

```
            <Route path="/symbols" element={
```

```
              <PrivateRoute>
```

```
                <SymbolLearning />
```

```
              </PrivateRoute>
```

```
            } />
```

```
            <Route path="/cognitive-progress" element={
```

```
              <PrivateRoute>
```

```
                <CognitiveProgress />
```

```
              </PrivateRoute>
```

```
            } />
```

```
            <Route path="/scenarios/" element={
```

```
              <PrivateRoute>
```

```
                <ScenariosPage />
```

```

    </PrivateRoute>
  } />
  <Route path="/scenarios/:id" element={
    <PrivateRoute>
      <ImmersiveScenario />
    </PrivateRoute>
  } />
  <Route path="/practice" element={
    <PrivateRoute>
      <PracticeModule />
    </PrivateRoute>
  } />
  <Route path="/testing" element={
    <PrivateRoute>
      <TestingModule />
    </PrivateRoute>
  } />
  <Route path="/profile" element={
    <PrivateRoute>
      <ProfileSettings />
    </PrivateRoute>
  } />
  <Route path="/diagnostic" element={
    <PrivateRoute>
      <Diagnostic />
    </PrivateRoute>
  } />
</Routes>
</main>
<Footer />

```

```

    </div>
  </Router>
</Provider>
);
}

```

```
export default App;
```

```

// src/services/analyticsService.js
import firebase from 'firebase/app';
import 'firebase/analytics';

```

```

// Firebase configuration
const firebaseConfig = {
  apiKey: process.env.REACT_APP_FIREBASE_API_KEY,
  authDomain: process.env.REACT_APP_FIREBASE_AUTH_DOMAIN,
  projectId: process.env.REACT_APP_FIREBASE_PROJECT_ID,
  storageBucket: process.env.REACT_APP_FIREBASE_STORAGE_BUCKET,
  messagingSenderId:
process.env.REACT_APP_FIREBASE_MESSAGING_SENDER_ID,
  appId: process.env.REACT_APP_FIREBASE_APP_ID,
  measurementId: process.env.REACT_APP_FIREBASE_MEASUREMENT_ID
};

```

```

// Initialize Firebase
firebase.initializeApp(firebaseConfig);
const analytics = firebase.analytics();

```

```

class AnalyticsService {
  // Track user registration

```

```
trackRegistration(method = 'email') {
  analytics.logEvent('sign_up', {
    method
  });
}
```

// Track login

```
trackLogin(method = 'email') {
  analytics.logEvent('login', {
    method
  });
}
```

// Track learning session

```
trackLearningSession(duration, method, symbolsLearned) {
  analytics.logEvent('learning_session', {
    duration, // in seconds
    method, // 'cognitive' or 'immersive'
    symbols_count: symbolsLearned.length,
    symbols: symbolsLearned.join(',')
  });
}
```

// Track scenario completion

```
trackScenarioCompletion(scenarioId, scenarioTitle, durationSeconds) {
  analytics.logEvent('scenario_completion', {
    scenario_id: scenarioId,
    scenario_title: scenarioTitle,
    duration: durationSeconds
  });
}
```

```

}

// Track test completion
trackTestCompletion(testId, testTitle, score, durationSeconds) {
  analytics.logEvent('test_completion', {
    test_id: testId,
    test_title: testTitle,
    score,
    duration: durationSeconds
  });
}

// Track feature usage
trackFeatureUsage(featureName, parameters = {}) {
  analytics.logEvent('feature_use', {
    feature_name: featureName,
    ...parameters
  });
}

// Set user properties
setUserProperties(properties) {
  Object.entries(properties).forEach(([key, value]) => {
    analytics.setUserProperty(key, value);
  });
}
}

export default new AnalyticsService();

```

```
// src/services/errorAnalysisService.js
class ErrorAnalysisService {
  constructor() {
    this.errorMatrix = {};
    this.symbolsAccuracy = {};
  }

  loadUserData(userData) {
    if (userData) {
      this.errorMatrix = userData.errorMatrix || {};
      this.symbolsAccuracy = userData.symbolsAccuracy || {};
    }
    return this;
  }

  recordAttempt(expectedSymbol, actualSymbol, responseTime) {
    // Initialize symbol data if it doesn't exist
    if (!this.symbolsAccuracy[expectedSymbol]) {
      this.symbolsAccuracy[expectedSymbol] = {
        correct: 0,
        attempts: 0,
        avgResponseTime: 0,
        lastPracticed: new Date()
      };
    }

    const isCorrect = expectedSymbol === actualSymbol;
    const symbolData = this.symbolsAccuracy[expectedSymbol];

    // Update accuracy statistics
  }
}
```

```

symbolData.attempts += 1;
symbolData.lastPracticed = new Date();

if (isCorrect) {
    symbolData.correct += 1;
} else {
    // Record error in matrix
    if (!this.errorMatrix[expectedSymbol]) {
        this.errorMatrix[expectedSymbol] = {};
    }
    if (!this.errorMatrix[expectedSymbol][actualSymbol]) {
        this.errorMatrix[expectedSymbol][actualSymbol] = 0;
    }
    this.errorMatrix[expectedSymbol][actualSymbol] += 1;
}

// Update average response time
symbolData.avgResponseTime = (symbolData.avgResponseTime *
(symbolData.attempts - 1) + responseTime) / symbolData.attempts;

return this;
}

getAccuracy(symbol) {
    if (!this.symbolsAccuracy[symbol] || this.symbolsAccuracy[symbol].attempts
=== 0) {
        return 0;
    }

    return (this.symbolsAccuracy[symbol].correct /

```

```

this.symbolsAccuracy[symbol].attempts) * 100;
    }

    getOverallAccuracy() {
        let totalCorrect = 0;
        let totalAttempts = 0;

        Object.values(this.symbolsAccuracy).forEach(data => {
            totalCorrect += data.correct;
            totalAttempts += data.attempts;
        });

        return totalAttempts > 0 ? (totalCorrect / totalAttempts) * 100 : 0;
    }

    getProblematicSymbols(thresholdAccuracy = 90) {
        const problematic = [];

        for (const symbol in this.symbolsAccuracy) {
            const accuracy = this.getAccuracy(symbol);
            if (accuracy < thresholdAccuracy &&
this.symbolsAccuracy[symbol].attempts >= 5) {
                problematic.push({
                    symbol,
                    accuracy,
                    avgResponseTime: this.symbolsAccuracy[symbol].avgResponseTime,
                    attempts: this.symbolsAccuracy[symbol].attempts
                });
            }
        }
    }
}

```

```

    return problematic.sort((a, b) => a.accuracy - b.accuracy);
  }

  getConfusedPairs() {
    const pairs = [];

    for (const expected in this.errorMatrix) {
      for (const actual in this.errorMatrix[expected]) {
        if (this.errorMatrix[expected][actual] >= 3) {
          pairs.push({
            expected,
            confused: actual,
            count: this.errorMatrix[expected][actual],
            accuracy: this.getAccuracy(expected)
          });
        }
      }
    }

    return pairs.sort((a, b) => b.count - a.count);
  }

  generateCustomExercises() {
    const exercises = [];

    // Exercises for problematic symbols
    const problematicSymbols = this.getProblematicSymbols();
    if (problematicSymbols.length > 0) {
      exercises.push({

```

```

    type: 'recognition',
    title: 'Problematic Symbols Recognition',
    description: 'Practice with symbols that need improvement',
    symbols: problematicSymbols.map(item => item.symbol),
    repetitions: Math.min(5, problematicSymbols.length * 2),
    difficulty: 'adaptive'
  });
}

// Exercises for confused pairs
const confusedPairs = this.getConfusedPairs();
if (confusedPairs.length > 0) {
  const pairSymbols = new Set();
  confusedPairs.slice(0, 3).forEach(pair => {
    pairSymbols.add(pair.expected);
    pairSymbols.add(pair.confused);
  });

  exercises.push({
    type: 'differentiation',
    title: 'Similar Symbols Differentiation',
    description: 'Learn to distinguish between easily confused symbols',
    symbols: Array.from(pairSymbols),
    pairs: confusedPairs.slice(0, 3),
    repetitions: 3,
    difficulty: 'challenging'
  });
}

return exercises;

```

```

}

getRecommendedSymbolsToLearn(knownSymbols) {
  // Define symbol groups by difficulty
  const symbolGroups = {
    beginners: ['E', 'T', 'A', 'N', 'I', 'M', 'S', 'U', 'R', 'W', 'D', 'K', 'G', 'O'],
    intermediate: ['H', 'V', 'F', 'L', 'P', 'J', 'B', 'X', 'C', 'Y', 'Z', 'Q'],
    numbers: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '0'],
    punctuation: ['.', ',', '?', '"', '!', '/', '(', ')', '&', ':', ';', '=', '+', '-', '_', '"', '$', '@']
  };

  // Create a set of known symbols for efficient lookup
  const knownSet = new Set(knownSymbols);

  // Function to get recommended symbols from a group
  const getFromGroup = (group, count) => {
    return group.filter(symbol => !knownSet.has(symbol)).slice(0, count);
  };

  // Get recommendations based on known symbol count
  if (knownSymbols.length < 5) {
    return getFromGroup(symbolGroups.beginners, 3);
  } else if (knownSymbols.length < 15) {
    return getFromGroup(symbolGroups.beginners, 2);
  } else if (knownSymbols.length < 26) {
    return [...getFromGroup(symbolGroups.beginners, 1),
...getFromGroup(symbolGroups.intermediate, 1)];
  } else if (knownSymbols.length < 36) {
    return [...getFromGroup(symbolGroups.intermediate, 1),
...getFromGroup(symbolGroups.numbers, 1)];
  }
}

```

```

    } else {
        return getFromGroup(symbolGroups.punctuation, 2);
    }
}

getUserData() {
    return {
        errorMatrix: this.errorMatrix,
        symbolsAccuracy: this.symbolsAccuracy,
        overallAccuracy: this.getOverallAccuracy(),
        problematicSymbols: this.getProblematicSymbols(),
        confusedPairs: this.getConfusedPairs()
    };
}

}

export default new ErrorAnalysisService();

// src/services/morseService.js
class MorseService {
    constructor() {
        this.audioContext = null;
        this.oscillator = null;
        this.gainNode = null;
        this.frequency = 700; // Hz
        this.dotDuration = 60 / (50 * 12); // 12 WPM standard
        this.initialized = false;
    }

    initialize() {

```

```

    if (this.initialized) return;

    this.audioContext = new (window.AudioContext ||
window.webkitAudioContext)();
    this.gainNode = this.audioContext.createGain();
    this.gainNode.gain.value = 0;
    this.gainNode.connect(this.audioContext.destination);

    this.oscillator = this.audioContext.createOscillator();
    this.oscillator.type = 'sine';
    this.oscillator.frequency.value = this.frequency;
    this.oscillator.connect(this.gainNode);
    this.oscillator.start();

    this.initialized = true;
}

setSpeed(wpm) {
    this.dotDuration = 60 / (50 * wpm);
    return this;
}

playMorseSymbol(symbol) {
    if (!this.initialized) this.initialize();

    const morseCode = this.symbolToMorse(symbol);
    return this.playMorseSequence(morseCode);
}

playMorseSequence(sequence) {

```

```
if (!this.initialized) this.initialize();
```

```
return new Promise((resolve) => {
```

```
  let currentTime = this.audioContext.currentTime;
```

```
  const pauseDuration = this.dotDuration;
```

```
  let totalDuration = 0;
```

```
  for (let i = 0; i < sequence.length; i++) {
```

```
    switch (sequence[i]) {
```

```
      case '!':
```

```
        this.gainNode.gain.setValueAtTime(1, currentTime);
```

```
        currentTime += this.dotDuration;
```

```
        this.gainNode.gain.setValueAtTime(0, currentTime);
```

```
        totalDuration += this.dotDuration;
```

```
        break;
```

```
      case '-':
```

```
        this.gainNode.gain.setValueAtTime(1, currentTime);
```

```
        currentTime += this.dotDuration * 3;
```

```
        this.gainNode.gain.setValueAtTime(0, currentTime);
```

```
        totalDuration += this.dotDuration * 3;
```

```
        break;
```

```
      case ' ':
```

```
        currentTime += this.dotDuration * 7;
```

```
        totalDuration += this.dotDuration * 7;
```

```
        break;
```

```
    }
```

```
    // Add pause between elements (except after spaces or at the end)
```

```
    if (i < sequence.length - 1 && sequence[i] !== ' ' && sequence[i+1] !== ' ')
```

```
{
```

```

        currentTime += pauseDuration;
        totalDuration += pauseDuration;
    }
}

// Resolve promise after the sequence finishes playing
setTimeout(() => resolve(), totalDuration * 1000);
});
}

```

```

symbolToMorse(symbol) {
    const morseCodeMap = {
        'A': '-.-', 'B': '-...', 'C': '-.-.', 'D': '-..', 'E': '.',
        'F': '..-.', 'G': '--.', 'H': '....', 'I': '..', 'J': '---',
        'K': '-.-', 'L': '-..', 'M': '--', 'N': '-.', 'O': '---',
        'P': '..-.', 'Q': '--.-', 'R': '-.-', 'S': '...', 'T': '-',
        'U': '..-', 'V': '...-', 'W': '-.-', 'X': '-.-.-', 'Y': '-.-.-',
        'Z': '-...', '0': '-----', '1': '-----', '2': '..---',
        '3': '...--', '4': '....-', '5': '.....', '6': '-....',
        '7': '--...', '8': '---..', '9': '----.', ':': ':-.-.-',
        ',': '-.-.-.-', '?': '..-.-.', '"': '----.', '!': '-.-.-.-',
        '/': '-.-.-', '(': '-.-.-', ')': '-.-.-.-', '&': '-.-.-',
        ':': '---...', ';': '-.-.-.-', '=': '-.-.-', '+': '-.-.-',
        '-': '-....-', '_': '..-.-.-', '"': '-.-.-.-', '$': '...-.-.-',
        '@': '-.-.-.-'
    };

    return morseCodeMap[symbol.toUpperCase()] || "";
}

```

```

morseToSymbol(morse) {
  const symbolMap = {
    '.-': 'A', '-...': 'B', '-.-.': 'C', '-..': 'D', '!.': 'E',
    '..-': 'F', '--.': 'G', '....': 'H', '...': 'I', '---': 'J',
    '-.-': 'K', '-..': 'L', '--': 'M', '-.': 'N', '---': 'O',
    '---': 'P', '--.-': 'Q', '-.-': 'R', '...': 'S', '-': 'T',
    '-.': 'U', '...': 'V', '--': 'W', '-..': 'X', '-.-': 'Y',
    '-..': 'Z', '-----': '0', '----': '1', '---': '2',
    '---': '3', '---': '4', '----': '5', '----': '6',
    '---': '7', '---': '8', '----': '9', '---': '!',
    '---': ',', '---': '?', '----': '"', '---': "'",
    '---': '/', '---': '(', '---': ')', '---': '&',
    '---': ':', '---': ';', '---': '=', '---': '+',
    '---': '-', '---': '_', '---': '"', '---': '$',
    '---': '@'
  };

  return symbolMap[morse] || "";
}

```

```

textToMorse(text) {
  return text.toUpperCase().split("").map(char => {
    if (char === ' ') return ' ';
    return this.symbolToMorse(char);
  }).join(' ');
}

```

```

morseToText(morse) {
  return morse.split(' ').map(word =>
    word.split(' ').map(symbol => this.morseToSymbol(symbol)).join("")
  );
}

```

```

    ).join(' ');
  }

  cleanup() {
    if (this.oscillator) {
      this.oscillator.stop();
      this.oscillator.disconnect();
    }

    if (this.gainNode) {
      this.gainNode.disconnect();
    }

    if (this.audioContext) {
      this.audioContext.close();
    }

    this.initialized = false;
  }
}

export default new MorseService();

import axios from 'axios';
import setAuthToken from '../utils/setAuthToken';

// Load User
export const checkAuth = () => async dispatch => {
  if (localStorage.token) {
    setAuthToken(localStorage.token);
  }
}

```

```

    }

    try {
      const res = await axios.get('/api/auth/user');

      dispatch({
        type: 'USER_LOADED',
        payload: res.data
      });
    } catch (err) {
      dispatch({
        type: 'AUTH_ERROR'
      });
    }
  };

// Register User
export const registerUser = (formData, callback) => async dispatch => {
  try {
    const config = {
      headers: {
        'Content-Type': 'application/json'
      }
    };

    const res = await axios.post('/api/auth/register', formData, config);

    dispatch({
      type: 'REGISTER_SUCCESS',
      payload: res.data
    });
  }
};

```

```

    });

    dispatch(checkAuth());

    if (callback) callback();
  } catch (err) {
    const errors = err.response.data.errors;

    if (errors) {
      // Handle errors if needed
    }

    dispatch({
      type: 'REGISTER_FAIL',
      payload: err.response.data.error || 'Registration failed'
    });
  }
};

// Login User
export const loginUser = (email, password) => async dispatch => {
  try {
    const config = {
      headers: {
        'Content-Type': 'application/json'
      }
    };

    const res = await axios.post('/api/auth/login', { email, password }, config);
  }
};

```

```

    dispatch({
      type: 'LOGIN_SUCCESS',
      payload: res.data
    });

    dispatch(checkAuth());
  } catch (err) {
    dispatch({
      type: 'LOGIN_FAIL',
      payload: err.response.data.error || 'Invalid credentials'
    });
  }
};

// Logout
export const logout = () => dispatch => {
  dispatch({ type: 'CLEAR_PROFILE' });
  dispatch({ type: 'LOGOUT' });
};

// Clear Errors
export const clearErrors = () => dispatch => {
  dispatch({ type: 'CLEAR_ERRORS' });
};

```

Додаток Г – Ілюстративна частина

ІЛЮСТРАТИВНА ЧАСТИНА

**РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АБЕТКИ МОРЗЕ НА ОСНОВІ
АДАПТИВНОГО ПЕРСОНАЛІЗОВАНОГО ПІДХОДУ**

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська кваліфікаційна робота на тему:
«Розробка вебзастосунку для вивчення абетки Морзе на основі
адаптивного персоналізованого підходу»

Автор: студентка групи 1ПІ-216 Кременчук А.І.
Науковий керівник: к.т.н., доц. каф. ПЗ Ракитянська Г.Б.

Вінниця - 2025

Рисунок Г.1 — Титульний слайд

Актуальність теми

Абетка Морзе, хоч і вважається застарілою для масового використання, досі відіграє важливу роль у низці специфічних сфер. Серед радіоаматорів (операторів любительського радіо) Морзе залишається популярним способом обміну повідомленнями. Багато радіоаматорів продовжують передавати повідомлення абеткою Морзе на швидкості 15–20 слів за хвилину, навіть попри те, що знання Морзе більше не є обов'язковим для отримання ліцензії. Характерний ритм кодів дозволяє таким сигналам пробиватися крізь перешкоди і шум ефіру краще, ніж голосові передачі.

Завдяки цьому з допомогою простого передавача малої потужності (на кілька ват) і дротяної антени оператори Морзе можуть встановлювати зв'язок на відстанях у тисячі кілометрів.

У сучасній авіації цифрові канали зв'язку та голосові радіосистеми майже повністю витіснили абетку Морзе з повсякденного використання. Втім, Морзе все ще використовується для ідентифікації навігаційних радіомаяків: наземні VOR/NDB маяки передають свої індикатори у вигляді звукових сигналів Морзе, що пілоти прослуховують для підтвердження правильності налаштування.

Навіть якщо більшість пілотів не знають Морзе досконало, навчальні програми все ж ознайомлюють їх з базовими сигналами. Деякі льотні школи досі включають абетку Морзе до навчального курсу як засіб останньої надії: у випадку відмови радіосистем пілот зможе подати сигнал SOS блиманням бортових вогнів чи ручним радіомаяком. У ряді країн (наприклад, Бразилія, Японія) навіть існують екзаменаційні вимоги на вміння розпізнавати основні сигнали Морзе для отримання авіаційних сертифікатів. Це свідчить, що авіаційна спільнота розглядає знання Морзе як корисний навик у непередбачених ситуаціях.

Морзе розглядається військовими як резервний канал зв'язку на випадок, якщо високотехнологічні системи будуть недоступні. Шифровані цифрові канали можуть зазнати збоїв чи глушіння, тоді як передача повідомлень абеткою Морзе потребує мінімального обладнання і може працювати в умовах, де інші засоби неефективні. У спеціальних операціях навички Морзе можуть застосовуватися для прихованого зв'язку, адже вузькосмуговий сигнал Морзе важче виявити чи заглушити противнику. Таким чином, навіть у XXI столітті абетка Морзе залишається в арсеналі військових як запасний, але надійний спосіб комунікації.

Рисунок Г.2 — Актуальність теми

Мета, об'єкт та предмет дослідження

Метою роботи є покращення процесу вивчення абетки Морзе шляхом розробки спеціалізованого програмного забезпечення, що використовує адаптивний персоналізований підхід.

Об'єкт дослідження – процес розробки застосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу.

Предмет дослідження – методи та засоби розробки застосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу.

Рисунок Г.3 — Мета, об'єкт та предмет дослідження

Завдання дослідження

Основними задачами дослідження є:

- розробити архітектуру застосунку для вивчення абетки Морзе на основі адаптивного персоналізованого підходу;
- розробити інтерфейс користувача застосунку;
- розробити метод персоналізованого підходу;
- розробити алгоритми роботи застосунку;
- розробити функціонал застосунку;
- провести тестування розробленого застосунку.

Рисунок Г.4 — Завдання дослідження

Наукова новизна

1. Подальшого розвитку отримав метод когнітивно-адаптивної прогресії з аналізом патернів помилок, який, на відміну від відомих, використовує "матриці помилок" для детального відстеження специфічних помилок користувача, автоматично коригує частоту появи проблемних символів у навчальних вправах, адаптує швидкість передачі для різних груп символів на основі індивідуальних труднощів кожного користувача, та застосовує прогресивне розширення набору символів лише після досягнення 90% точності розпізнавання поточних символів.

2. Подальшого розвитку отримав метод навчання з імерсивними контекстними сценаріями, який, на відміну від відомих, занурює користувача у персоналізовані наративні контексти (морські подорожі, космічні місії, історії порятунку), де використання абетки Морзе має практичне сюжетне значення, застосовує ритмічно-музичну синхронізацію для перетворення абстрактних кодів Морзе.

Рисунок Г.5 — Наукова новизна

Практична цінність отриманих результатів

Практична цінність одержаних результатів полягає в можливості використання розробленої системи для вивчення абетки Морзе та регулярної перевірки рівня знань.

Рисунок Г.6 — Практична цінність отриманих результатів

Аналоги

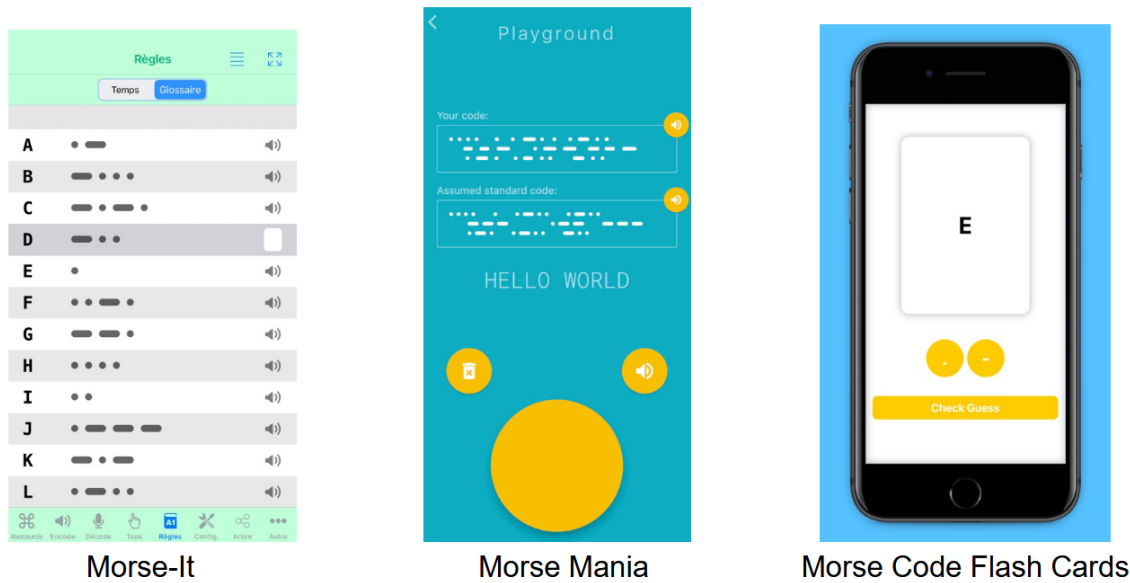


Рисунок Г.7 — Аналоги

Порівняльний аналіз аналогів

Характеристика	Morse-It	Morse Code	Morse Code Flash Cards	Власна розробка
Аудіо тренування	+	+	+	+
Візуальні підказки	+	+	+	+
Режими тренування	+	+	+	+
Відслідковування прогресу	+	+	-	+
Інструкції та підказки	+	+	+	+
Адаптація навчання під темп користувача	-	-	-	+
Загальний бал	5	5	4	6

Рисунок Г.8 — Порівняльний аналіз аналогів

Використані технології

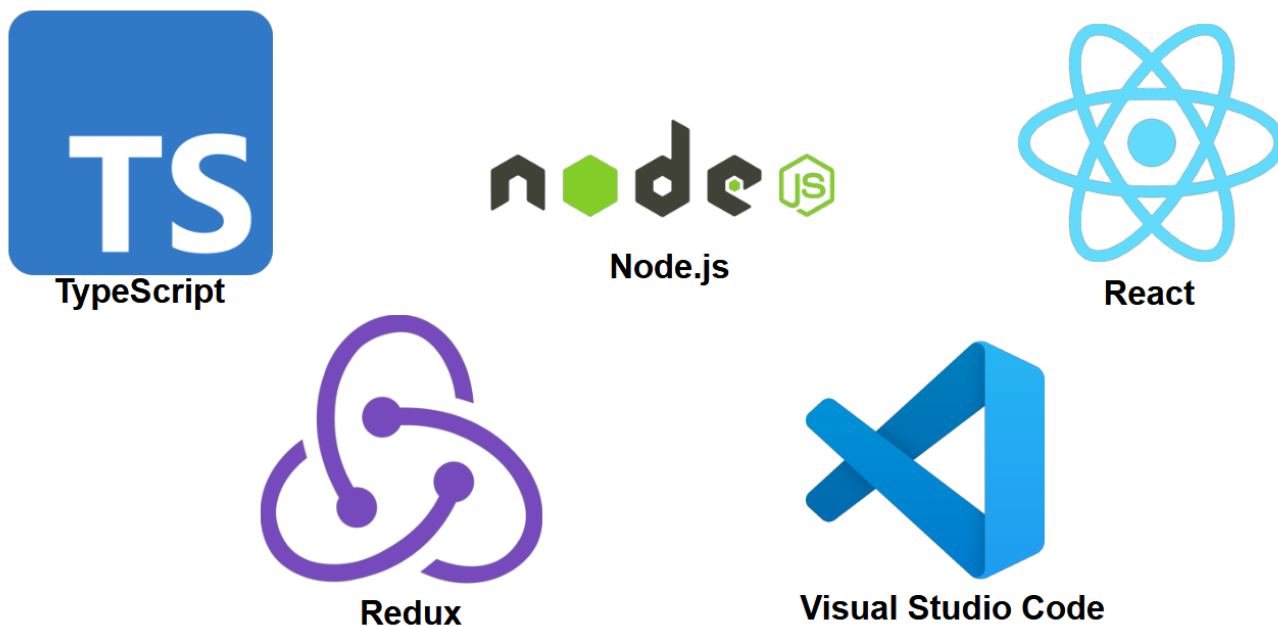


Рисунок Г.9 — Використані технології

Розробка методу когнітивно-адаптивної прогресії з аналізом патернів помилок

1. Початкова оцінка користувача. Сюди входить: проведення короткого тесту для визначення початкового рівня сприйняття аудіо та швидкості реакції, збір даних про попередній досвід з Морзе, створення початкового профілю користувача.
2. Представлення символів через аудіо та візуальний супровід, починаючи з повільної швидкості передачі символів з поступовим збільшення швидкості на основі показників засвоєння.
3. Відстеження специфічних пар символів, які користувач плутає, визначення типових помилок, на основі яких створюється «матриця помилок» для кожного користувача.
4. Поступове збільшення частоти появи проблемних символів та автоматичне коригування швидкості WPM для різних груп символів.
5. Після досягнення 90% точності у визначення проблемних символів, система розпізнає це як успішне їх опанування, та додає нові символи.
6. На основі успішності навчання, формуються звіти про прогрес з візуалізацією даних, які представлені у вигляді графіків та діаграм. На основі отриманого прогресу у навчанні, формуються автоматичні рекомендації щодо оптимальної тривалості та частоти тренувань.

Рисунок Г.10 — Розробка методу когнітивно-адаптивної прогресії з аналізом патернів помилок

Розробка методу когнітивно-адаптивної прогресії з аналізом патернів помилок

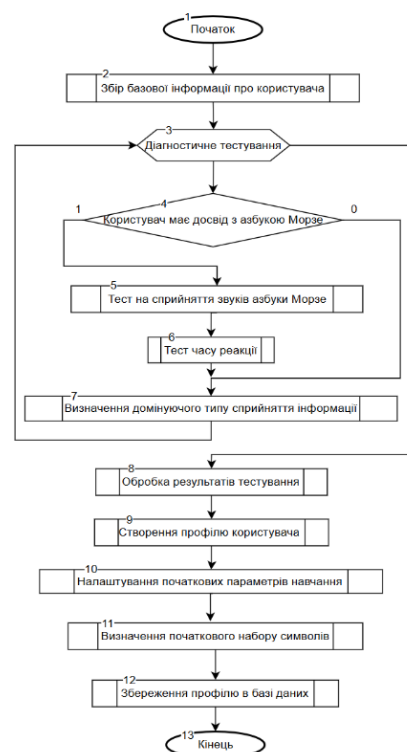


Рисунок Г.11 — Блок-схема методу когнітивно-адаптивної прогресії з аналізом патернів помилок

Розробка методу навчання з імерсивними контекстними сценаріями

1. Аналіз типу сприйняття інформації користувачем.
2. Створення динамічної моделі навчання шляхом формування персоналізованої карти складності для кожного символу Морзе.
3. Створення тематичних сценаріїв з Адаптацією сценаріїв на основі інтересів користувача.
4. Генерування ритмічних патернів, що відповідають кодам Морзе, налаштування темпу та тону сигналів відповідно до індивідуальних параметрів сприйняття.
5. Динамічне коригування складності на основі опанування користувачем абетки Морзе.
6. Автоматичне виявлення символів, що "забуваються" з часом та проактивне повторення матеріалу на основі передбачення можливого забування

Рисунок Г.12 — Розробка методу навчання з імерсивними контекстними сценаріями

Розробка методу навчання з імерсивними контекстними сценаріями

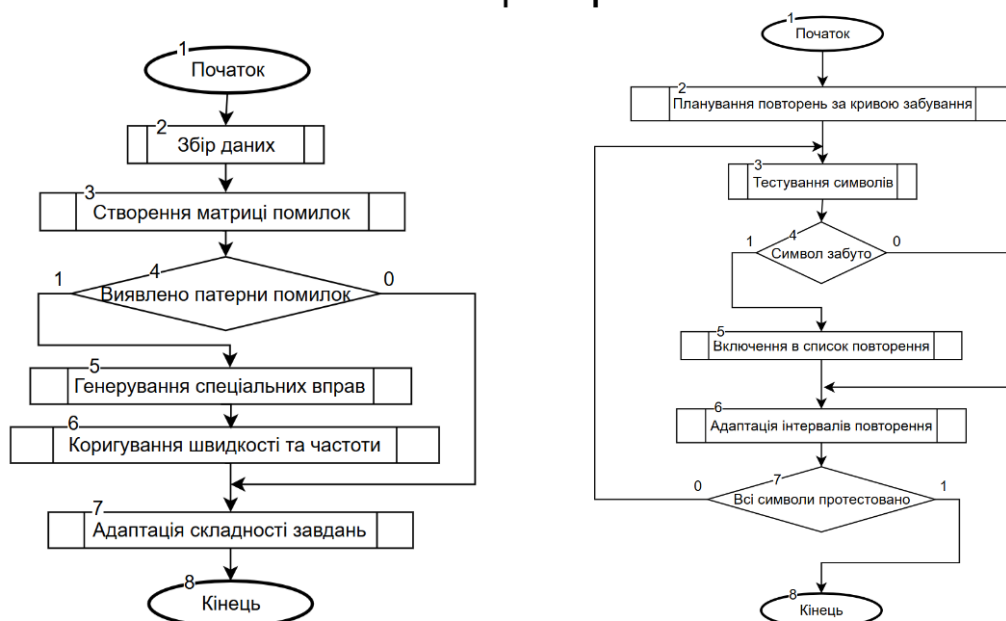


Рисунок Г.13 — Розробка методу навчання з імерсивними контекстними сценаріями

Тестування застосунку

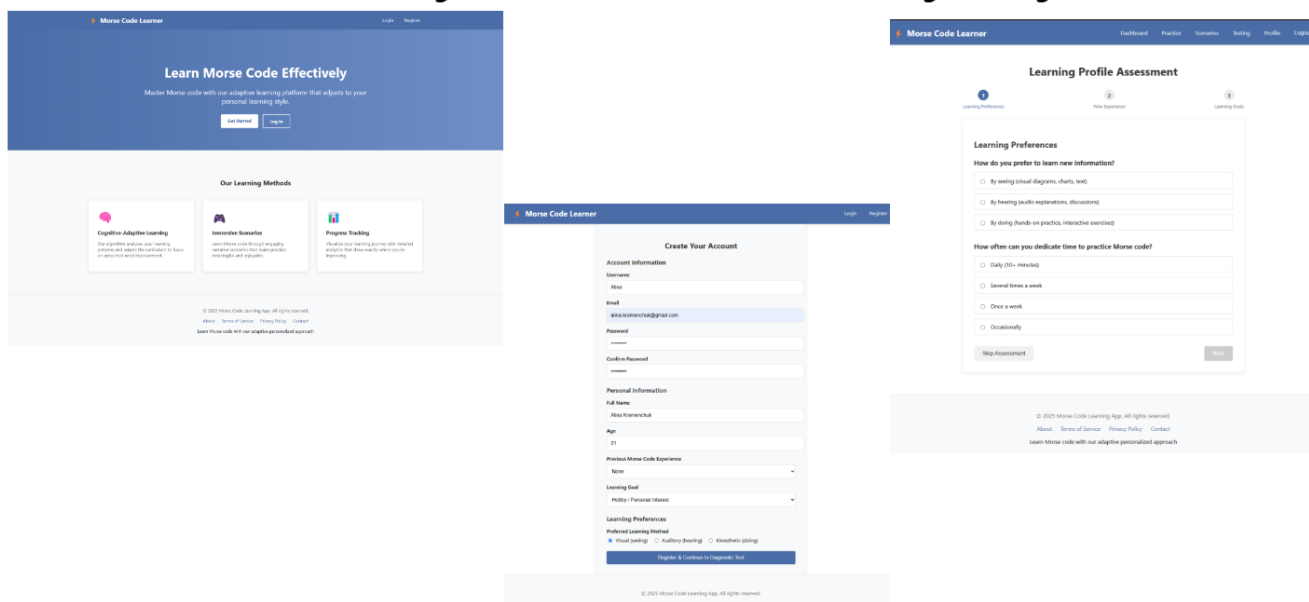


Рисунок Г.14 — Тестування реєстрації

Тестування застосунку

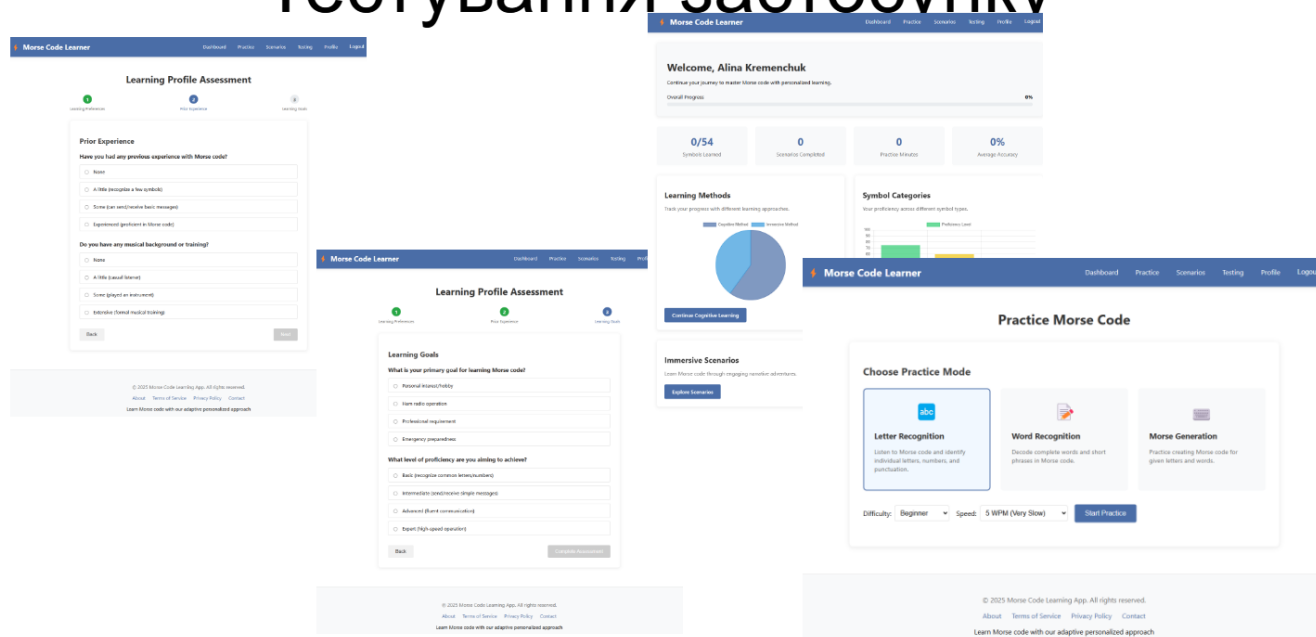


Рисунок Г.15 — Тестування головного модуля застосунку

Тестування застосунку

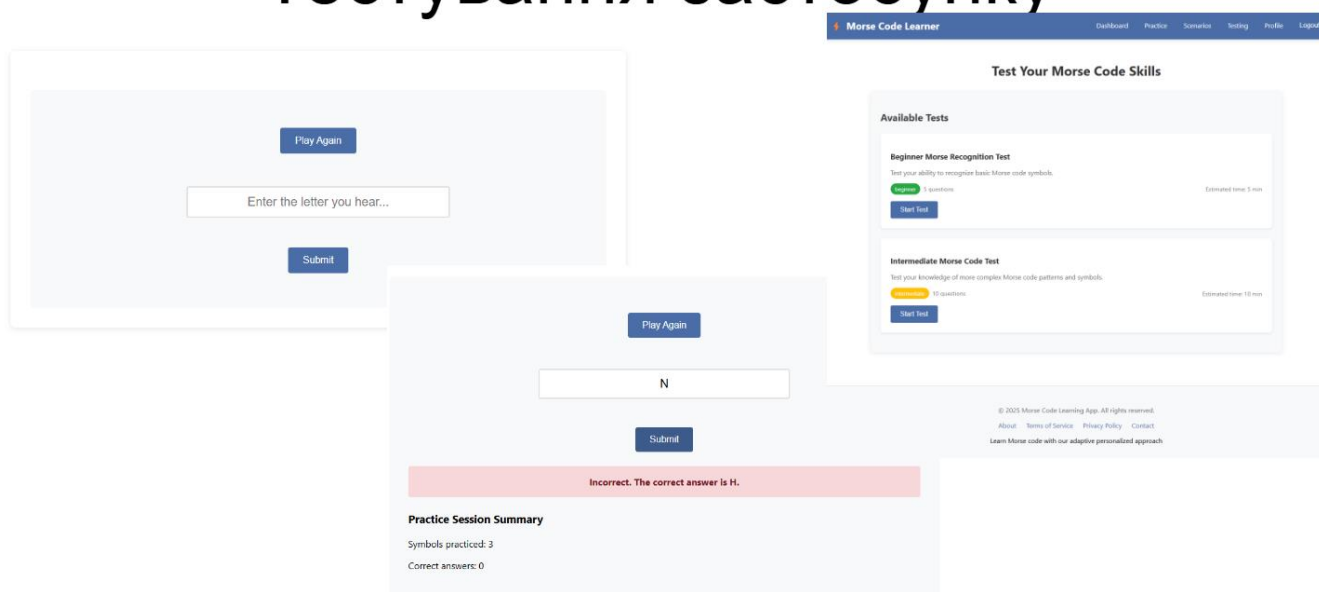


Рисунок Г.16 — Тестування процесу навчання

Тестування застосунку

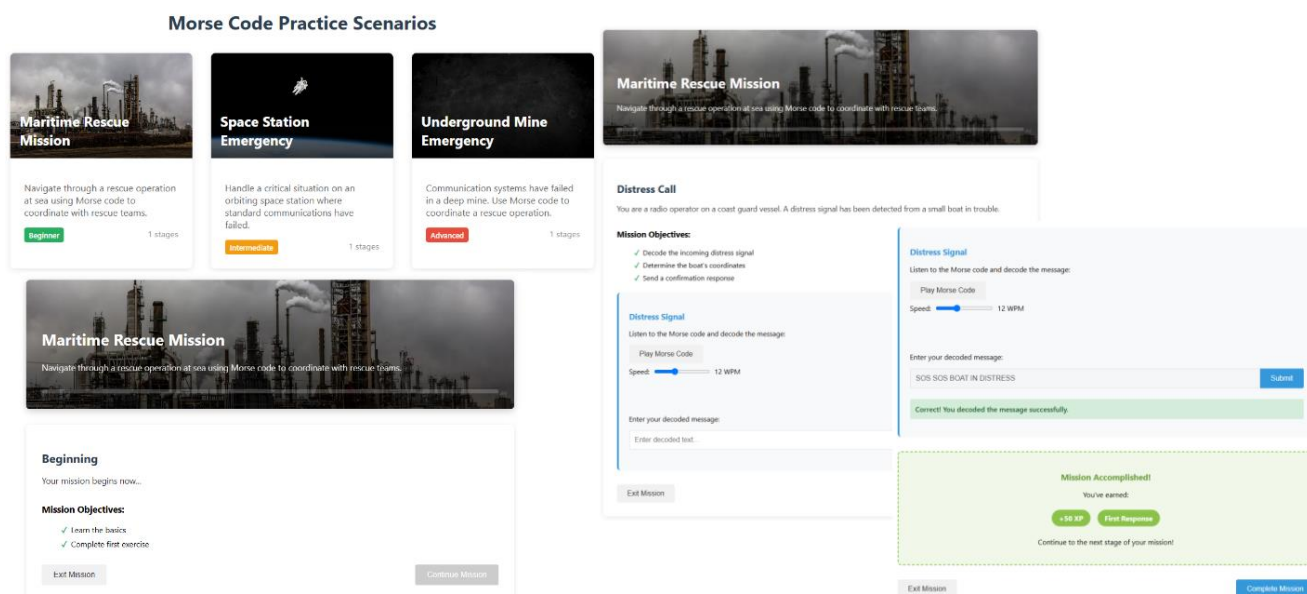


Рисунок Г.17 — Тестування імерсивних сценаріїв

Висновки

У результаті виконання бакалаврської кваліфікаційної роботи було розроблено вебзастосунок для вивчення абетки Морзе. Розроблений застосунок дозволяє вивчати символи та звуки з абетки Морзе, проходити тестування для перевірки вивчених символів, проходити практику у сценаріях, близьких до реального застосування, проходити повторення для відновлення забутих символів. Бакалаврську кваліфікаційну роботу реалізовано згідно методичних вказівок.

Вебзастосунок розроблено мовою JavaScript, середовищем розробки Visual Studio Code, фреймворку React.js та Node.js.

У першому розділі було проведено аналіз сучасного стану застосунків для вивчення абетки Морзе, доведено актуальність розробки такого застосунку. Проведено порівняльний аналіз аналогів з розроблюваним вебзастосунком, продемонстровано його переваги та таким чином доведено доцільність розробки. Проведено аналіз методів розв'язання задачі, обрано гібридний метод з використанням бібліотек для роботи із звуками. Проведено постановку задач дослідження з розробки вебзастосунку з вивчення абетки Морзе на основі адаптивного персоналізованого підходу.

Рисунок Г.18 — Висновки (частина 1)

Висновки

У другому розділі було проведено аналіз інформаційного забезпечення системи для вивчення абетки Морзе, описано дані, які у ній використовуються та формуються. Розроблено архітектуру системи за клієнт-серверним підходом, описано складові архітектури. Розроблено метод когнітивно-адаптивної прогресії з аналізом патернів помилок та метод навчання з імерсивними контекстними сценаріями, побудовано блок-схеми алгоритмів складових цих методів.

У третьому розділі було проведено варіантний аналіз мов програмування, обрано мову JavaScript для розробки системи для вивчення абетки Морзе, розроблено базу даних, що забезпечує збереження даних про користувача, тести та іншу супутню інформацію, розроблено модулі системи, реалізовано інтерфейс користувача та загальний алгоритм роботи системи.

У четвертому розділі було проведено аналіз методів тестування системи для вивчення абетки Морзе. Було проведено її тестування, продемонстровано основний функціонал, наведено приклади використання, доведено, що розроблена система виконує поставлені на початку розробки завдання.

Рисунок Г.19 — Висновки (частина 2)

Апробація результатів роботи і публікації

Апробація матеріалів. Описані у бакалаврській кваліфікаційній роботі положення доповідались на конференції «LIV Всеукраїнська науково-технічна конференція підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2025)».

Публікації. А.І. Кременчук, Г.Б. Ракитянська. Вебзастосунок для вивчення абетки Морзе на основі адаптивного персоналізованого підходу / Матеріали LIV Всеукраїнської науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2025) : збірник доповідей [Електронний ресурс] — Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24124/19878>

Рисунок Г.20 — Апробація результатів роботи і публікації