Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра інформатики

РОЗРОБКА ВІРТУАЛЬНОГО СПІВРОЗМОВНИКА ДЛЯ ПРАКТИКИ РОЗМОВНИХ НАВИЧОК (АНГЛІЙСЬКА МОВА)

Текстова частина до курсової роботи за спеціальністю

122 «Комп’ютерні науки»

Керівник курсової роботи
Доцент кандидат наук Тригуб О. С.
_____ *(підпис)*
“____” _____ 2024 р.
Виконала студентка Полякова Любов Василівна
“____” _____ 2024 р.

Київ 2024

Зміст

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри інформатики
Гороховський С. С.

_____ (підпис)

„____”_____2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ на курсову роботу

студенту_____ факультету _____ _____курсу

ТЕМА _____

Зміст ТЧ до курсової роботи:

Анотація

Вступ

1 Розділ 1. Аналіз існуючих рішень

2 Розділ 2. Обґрунтування вибору технологій та інструментів

3 Розділ 3. Технічні можливості застосунку

4 Розділ 4. Опис бази даних

5 Розділ 5. Ресурси тем до відповідних рівнів володіння мовою

Висновки

Список літератури

Дата видачі „___” _____ 2024 р.

Керівник _____ (підпис)

Завдання отримав_____ (підпис)

Календарний план виконання роботи

№ п/п	Етап роботи	Термін виконання етапу
1.	Отримання завдання.	24.10.2023
2.	Обговорення основного функціоналу, дискусія з приводу можливих API для штучного інтелекту, їх доступності та вартості.	01.01.2024 - 14.01.2024
3.	Розробка прототипу в Figma.	14.01.2024 – 01.02.2024
3.	Розробка бекенду.	01.02.2024 – 01.04.2024
4.	Розробка фронтенду.	01.04.2024 – 01.05.2024
5.	Створення текстової частини.	01.05.2024 – 10.05.2024

Анотація

Розроблено віртуальний співрозмовник для практики розмовних навичок англійської мови за допомогою технологій React Native та Node.js, який дозволяє користувачам вести письмові діалоги на випадкові та обрані теми, з використанням Generative AI on Vertex AI. Основна мета проекту — поліпшення володіння розмовною англійською мовою через неперервну практику та інтерактивне спілкування. Основні функціональності включають перевірку граматики та лексики в процесі діалогу з можливістю їх вмикання та вимикання. Програма має легкий та інтуїтивно зрозумілий інтерфейс і може ефективно сприяти самостійному вивченню мови, надаючи користувачам цікаві та зручні інструменти для практики.

Вступ

У світі, де володіння англійською мовою стає все більш важливим для професійного та особистого розвитку, виникає потреба у доступному та ефективному інструменті для практики розмовних навичок.

Практика розмовних навичок є однією з найбільш ефективних методик удосконалення володіння іноземною мовою. Проте часто студенти стикаються з обмеженнями, такими як відсутність носіїв мови, високі витрати на репетиторів та незручні графіки занять. Використання віртуального співрозмовника може стати ефективним вирішенням цих проблем, надаючи користувачам можливість практикувати ‘живу’ англійську мову в будь-який момент та як завгодно довго.

Метою цієї курсової роботи є розробка віртуального співрозмовника, який дозволяє користувачам практикувати англійську мову через інтерактивний письмовий діалог. Для досягнення мети необхідно виконати такі завдання:

- Розробити архітектуру програми, яка підтримує діалоги з використанням штучного інтелекту.
- Реалізувати можливість перевірки граматики та лексики під час спілкування.
- Створити інтуїтивний інтерфейс для користувачів, де вони можуть легко взаємодіяти з віртуальним співрозмовником без зайвих питань та дискомфорту.
- Надати додаткові функції, такі як тестування граматики й лексики.

Об’єкт дослідження: Дослідження стосується застосунків, що використовують штучний інтелект для практики англійської мови.

Методи дослідження: Використовуються методи системного аналізу, розробки програмного забезпечення та тестування.

Проект пропонує новий підхід до самостійного вивчення англійської мови шляхом інтерактивного письмового діалогу з віртуальним співрозмовником. Інтеграція сучасних технологій, таких як React Native, Node.js та Generative AI on Vertex AI, створює умови для індивідуальної практики, яка ефективно допомагає користувачам покращити свої розмовні навички, граматику та словниковий запас.

Розділ 1. Аналіз існуючих рішень

1.1 SmallTalk2Me

Цей додаток пропонує функцію імітації інтерв'ю для підготовки до співбесід, автоматичні рівневі тести для оцінки навичок та навчальні історії для щоденної практики. SmallTalk2Me використовує штучний інтелект для персоналізації навчання, оцінки мовного рівня та надання зворотного зв'язку щодо граматики, вимови та словникового запасу. Це дозволяє учням визначити свої сильні та слабкі сторони і зосередитися на вдосконаленні специфічних навичок. Також SmallTalk2Me пропонує симулятори для IELTS Speaking та Writing та тести для оцінки загального рівня англійської мови. Тому цед додаток є корисним для тих, хто готується до цих іспитів.

Однак, симуляції в SmallTalk2Me більш спеціалізовані на співбесідах. Вони обмежені за кількістю сценаріїв і можуть бути менш персоналізовані, ніж вільний діалог зі штучним інтелектом. Користувачі не завжди можуть обрати конкретну тему або отримати негайний зворотний зв'язок щодо граматики.

1.2 Loora

Loora - популярний застосунок для практики англійської мови, пропонує користувачам достатньо широкий спектр тем. Користувачі можуть практикувати англійську з набором тем, які допомагають охопити бізнес, подорожі та повсякденні ситуації. Loora надає миттєвий відгук щодо граматики, допомагаючи користувачам вдосконалювати свої мовні навички під час діалогу. Також цей застосунок пропонує персоналізовані уроки та підлаштовується під рівень та цілі користувача, завдяки чому тренування ефективне для людей різного рівня.

Хоча миттєвий відгук є корисним, деяким користувачам не вистачає можливості вимкнути перевірку граматики та лексики для фокусування на змісті.

1.3 SpeakL

SpeakL створює атмосферу реальної розмови завдяки індивідуальним урокам із штучним інтелектом. Додаток допомагає практикувати діалоги та взаємодію з ботом, тренуючи впевненість та розширюючи словниковий запас. Також він дозволяє переклад на рідну мову та показує англійський варіянт, що корисно для вивчення нових виразів і самокорекції. Додаток підтримує кілька мов, допомагаючи учням різних рівнів та національностей ефективно вивчати англійську мову. Функція "Text Talk" дозволяє вводити текст рідною мовою, а штучний інтелект перекладає та допомагає зрозуміти, як краще сформулювати фразу англійською мовою.

Однак, список тем на які можна поговорити зі штучним інтелектом, а їх всього 40, досить обмежена.

1.4 Flow

Flow — це застосунок для практики розмовної англійської зі штучним інтелектом. Серед ключових функціональностей:

- Переклад повідомлень: наявна функція перекладу, яка дозволяє подолати мовні бар'єри.
- Підказки для відповідей: штучний інтелект надає корисні підказки для складання відповідей.
- Перефразування: штучний інтелект пропонує варіянти перефразування речень для покращення комунікації.

Однак, Flow має набір заздалегідь визначених тем, через що користувачі можуть відчувати брак різноманітності при підготовці до нестандартних діалогів.

1.5 Cory

Cory є інноваційним додатком для практики англійської, який дозволяє користувачам покращувати свої навички в реальних життєвих сценаріях завдяки AI-тренеру. Основні функції включають:

- Персоналізовані уроки: програма налаштовує уроки відповідно до рівня знань користувача.
- Симуляції реальних ситуацій: надає досвід розмов на основі сценаріїв з повсякденного життя, щоб підготувати до реальних розмов.
- Комплексне покращення навичок: програма допомагає розширити словниковий запас та граматику.

В той час як Cory зосереджений на симуляціях реальних ситуацій, він, на жаль, є обмеженим у варіативності тем і сценаріїв, які він пропонує.

1.6 ChitChatAI

Chitchat AI створений для покращення англійської розмовної практики. З його допомогою можна розвивати мовлення через рольові сценарії та бесіди зі штучним інтелектом. Додаток пропонує користувачам актуальні теми: від побутових ситуацій до розмов про сучасні тенденції. Він надає граматичні виправлення, пропонує варіянти виразів, підказки та оцінює навички, допомагаючи користувачам стати впевненішими в англійській мові. Також користувачі можуть переглядати власні розмовні статистики та зберігати ключові речення для подальшого повторення.

Попри широкий вибір тем, Chitchat AI все ж обмежений у своєму переліку тем.

1.7 Висновок

Враховуючи аналіз існуючих рішень, розробка даного застосунку має сенс, оскільки він може заповнити прогалини, які залишають поточні вже

існуючі застосунки. Пропонуючи можливість діалогів на випадкові та обрані теми з увімкненою чи вимкненою перевіркою граматики й лексики, застосунок може надати гнучкіший і персоналізованіший досвід для користувачів.

Розділ 2. Обґрунтування вибору технологій та інструментів

Обираючи мобільний застосунок для практики розмовної англійської, враховано кілька ключових факторів.

По-перше, зручність. Мобільні пристрої стали невід'ємною частиною повсякденного життя користувачів, і мобільний додаток дозволяє їм використовувати його будь-де і будь-коли, навіть у місцях, де немає доступу до комп'ютера. Це забезпечує мобільність і доступність для широкого кола користувачів, що активно працюють над вдосконаленням своєї англійської мови.

По-друге, інтерактивність. Мобільні додатки можуть забезпечити більш інтерактивний досвід завдяки використанню функцій сенсорного екрану, легкого голосового введення та інших можливостей мобільних пристроїв.

Крім того, мобільні застосунки зазвичай мають можливість реалізації сповіщень та нагадувань, що допомагає користувачам залишатися на зв'язку і регулярно використовувати додаток для покращення своїх навичок англійської мови.

Таким чином, обираючи мобільний додаток, було прагнення забезпечити максимальну доступність та зручність для користувачів, а також надати їм інтерактивний та захоплюючий досвід вивчення англійської мови.

2.1 Node.js

Node.js - це серверне середовище з відкритим кодом. Node.js безкоштовний та може працювати на різних платформах, таких як Windows, Linux, Unix, Mac OS X тощо. Особливість Node.js полягає в тому, що він використовує JavaScript на сервері. Node.js працює у режимі

однопотокового, неблокуючого, асинхронного програмування, що дуже ефективно використовує пам'ять. [1]

Вибір Node.js для мобільного додатку обгрунтовується кількома факторами:

- Швидкість розробки: Використання Node.js сприяє ефективній розробці за рахунок використання JavaScript як для серверної, так і для клієнтської частини додатку. Це дозволяє ефективно використовувати код на обох рівнях, що прискорює процес розробки.
- Масштабованість: Node.js забезпечує легку масштабованість додатку, що є важливим для передбачуваного зростання взаємодії користувачів з програмою.
- Асинхронність та багатопоточність: Модель асинхронного введення/виведення Node.js дозволяє ефективно обробляти багатопоточність, що особливо важливо для додатків з великою кількістю одночасних підключень, таких як мобільні застосунки.
- Екосистема npm: Широка екосистема пакетів npm дозволяє швидко знаходити та інтегрувати необхідні бібліотеки та рішення для додатку.

Отже, Node.js є відмінним вибором для створення мобільних додатків з штучним інтелектом з точки зору швидкості розробки, масштабованості та ефективності обробки багатопоточності.

2.2 React Native

React Native - це відкрита платформа для розробки мобільних додатків, яка базується на JavaScript та React. [2]

React Native було обрано для фронтенду мобільного додатку з декількох причин. По-перше, цей інструмент дозволяє розробляти мобільні застосунки для iOS та Android, використовуючи знайомий синтаксис

JavaScript. Це спрощує процес розробки, бо не потрібно вивчати окремі мови програмування для кожної платформи.

Крім того, React Native забезпечує високу швидкість розробки завдяки готовим компонентам і бібліотекам, що пропонуються спільнотою розробників. Це дозволяє прискорити процес створення функцій та взаємодії з користувачем.

Також важливо зазначити, що React Native дозволяє забезпечити однаковий користувацький досвід на обох платформах, зберігаючи при цьому високу продуктивність. Це дозволяє додатку привабити більше користувачів.

Враховуючи всі ці фактори, React Native стає логічним вибором для розробки фронтенду мобільного додатку.

2.3 MongoDB

MongoDB є відкритою, нереляційною системою управління базами даних, яка використовує гнучкі документи замість таблиць та рядків для обробки та зберігання різних форм даних. [3]

MongoDB було обрано для зберігання даних мобільного додатку з декількох причин. По-перше, MongoDB є NoSQL базою даних, яка дозволяє ефективно зберігати та обробляти саме неструктуровані дані. Дані в даному додатку можуть бути досить різноманітними, і MongoDB забезпечує гнучкість у їх зберіганні.

По-друге, MongoDB має відмінну масштабованість, що дозволяє додатку ефективно працювати при зростанні кількості користувачів та обсягу даних. Розширення бази даних за допомогою нових користувачів та їхніх даних можливе без значних перебоїв у роботі додатку.

Крім того, MongoDB надає широкий набір інструментів для адміністрування та моніторингу бази даних, що спрощує обслуговування та відлагодження.

2.4 Expo Go

Expo Go є платформою для розробки мобільних додатків, яка дозволяє розробникам запускати та тестувати React Native додатки на Android і iOS пристроях без необхідності компіляції нативного коду. Вона надає зручний спосіб перегляду змін у додатку в реальному часі, скануючи QR-код, який з'єднує додаток із розробницьким сервером.

Expo Go значно спрощує розробку та тестування мобільних додатків, використовуючи технології React Native, дозволяючи безпосередньо запускати проекти на мобільних пристроях без необхідності збирання нативного коду. Цей інструмент забезпечує швидкий попередній перегляд змін у коді в реальному часі. Використовуючи Expo Go, можна легко тестувати функціонал додатків, інтегрований з різноманітними бібліотеками та API, а також перевіряти взаємодію з користувацьким інтерфейсом та перехід між екранами.

Використання Expo Go дозволило заощадити значний час під час розробки додатка, оскільки можна було швидко ітерувати та випробовувати нові функції без затримок, пов'язаних із збіркою та розгортанням. Завдяки цьому додаток розвивався швидше.

2.5 Generative AI on Vertex AI

2.5.1 Аналіз альтернатив

Існує багато варіянтів API штучного інтелекту для інтеграції в даний додаток. Один з них — OpenAI GPT-3, який забезпечує розширені

можливості для генерації тексту та обробки природної мови. Плюси цього API включають високу точність і гнучкість у генерації діалогів на широкий спектр тем, що є ідеальним для використання в данному застосунку. Однак, використання GPT-3 може бути дорогим через цінову політику OpenAI. Ціни на API GPT-3 варіюються від 0.02 до 0.06 долара за тисячу токенів, залежно від обраного обсягу використання. Для оцінки витрат на використання API OpenAI GPT-3 під час експлуатації застосунку, визначімо приблизну кількість токенів, яка буде використовуватися кожним користувачем. Середній сценарій використання застосунку передбачає генерацію 500 токенів на сесію користувача. Якщо ціна за 1,000 токенів становить від 0.02 до 0.06 долара, то вартість однієї сесії буде коливатися від 0.01 до 0.03 долара.

Якщо взяти до уваги, що середній користувач використовує застосунок чотири рази на тиждень, місячна активність кожного користувача складе приблизно 16 сесій. Таким чином, місячна вартість використання GPT-3 на одного користувача складе від 0.16 до 0.48 долара.

Для гіпотетичних 1000 користувачів витрати на місяць становитимуть:

Мінімальна вартість: 0.16 долара × 1000 користувачів = 160 доларів

Максимальна вартість: 0.48 долара × 1000 користувачів = 480 доларів.

Також для використання OpenAI GPT-3 API необхідно імплементувати зашифрування даних як у транзиті, так і в спокої, використання автентифікації і авторизації для доступу до API та регулярне аудитування і моніторинг системи на предмет можливих вразливостей.

Інший варіянт, який був розглянутий — це Google Cloud Natural Language API, який дозволяє аналізувати сентимент тексту, визначати сутності та проводити синтаксичний аналіз. Цей API корисний для

розробників, що потребують детального розуміння мовних особливостей, таких як морфологічна структура речень, їх частини мови, та залежності між компонентами речення. Це дозволяє не тільки розпізнавати ключові слова та фрази, але й визначати емоційний забарвлення тексту, що є корисним для аналізу зворотного зв’язку користувачів або аналізу настроїв у соціальних мережах. Google Cloud Natural Language API застосовується у розробці клієнтських служб підтримки для автоматизації відповідей та аналізу запитань користувачів, а також у маркетингових дослідженнях для збору та аналізу споживчих відгуків. Однак, Google Cloud Natural Language API в основному фокусується на аналітичних задачах, таких як розпізнавання елементів мови, аналіз сентименту та визначення сутностей, але не має такої ж здатності до створення вільних, спонтанних тек 8 стів, як GPT-3. Наприклад, під час генерації тексту на основі вхідних даних, GPT-3 може створювати відповіді, що здаються винахідливими та більш "людськими", складаючи цілісні історії або відповіді, які виходять за межі простого аналізу тексту. У контрасті, Google API вирішує конкретніші завдання, але без здатності до розгортання більш великих або творчих текстових блоків.

Ще один можливий варіянт — Microsoft Azure Text Analytics — цей API надає послуги, такі як виявлення ключових фраз, розпізнавання іменованих сутностей, аналіз мовлення і сентиментів.[4] Цей API використовується у багатьох корпоративних додатках для автоматизації обробки відгуків клієнтів, аналізу настроїв у соціальних медіа, та підтримки клієнтських служб через чат-ботів. Щодо вартості, Azure Text Analytics пропонує різні тарифні плани, які починаються від 0 доларів за перші 5,000 текстових записів на місяць до більш високих платежів за додаткове використання. У контексті застосунку для практики розмовної англійської, цей API може бути корисним для аналізу емоційного забарвлення та

визначення ключових понять у діалогах, однак цього не достатньо, оскільки основною функцією застосунку є генерація та підтримка більш складних діалогів на вибрані теми з перевіркою граматики та лексики. Azure Text Analytics не забезпечує генерації тексту.

Останній варіянт, який був розглянутий, але не був обраний для інтеграції в даний застосунок це Dialogflow від Google. Dialogflow спеціалізується на створенні інтерфейсів для розмовних застосунків, таких як чат-боти та голосові асистенти, використовуючи природну мову. Цей API включає підтримку багатьох мов, гнучкість у створенні користувацьких відповідей та здатність інтеграції з іншими сервісами Google. Основними перевагами Dialogflow є його здатність до розуміння складних запитів та контекстуальних діалогів, що робить його ідеальним для створення динамічних діалогових систем, які можуть адаптуватися до потреб користувача. Цей варіянт є досить логічним для інтегрування в даний застосунок, але був знайдений кращий.

Generative AI on Vertex AI API від Google надає широкий спектр можливостей для машинного навчання, зокрема для створення, розгортання та масштабування моделей прогнозування.[5] Цей API особливо корисний для чатботів та діалогових систем завдяки своїй здатності інтегрувати різноманітні моделі машинного навчання, що можуть покращити якість і точність відповідей на запити користувачів.

Крім того, завдяки своїй мультимодельності, Generative AI on Vertex AI зможе також допомогти у перевірці граматики та лексики.

Для оцінки витрат на використання Generative AI on Vertex AI API під час експлуатації застосунку також необхідно визначити приблизну кількість запитів, які будуть виконуватися кожним користувачем. Припустимо, що середній сценарій використання застосунку передбачає

40-50 запитів на прогнозування (виклик API) на сесію користувача. Відомо, що ціна на Generative AI on Vertex AI API становить приблизно $0.10 за 1,000 запитів.

Якщо середній користувач використовує застосунок чотири рази на тиждень, то місячна активність кожного користувача складе:

45 запитів за сесію × 16 сесій на місяць = 720 запитів на місяць

Згідно з ціновою політикою Google Vertex AI, де $0.10 на 1,000 запитів, вартість за 720 запитів становитиме:

720запитів÷1,000запитів×$0.10=$0.072

Таким чином, місячна вартість використання Vertex AI's Prediction API на одного користувача буде приблизно $0.072. Якщо розглядати 1000 користувачів, загальні витрати на місяць складатимуть:

1000користувачів×$0.072=$72

Ця оцінка показує, що використання Generative AI on Vertex AI є надзвичайно економічним для застосунків з великою кількістю користувачів, особливо порівняно з витратами на інші API, такі як GPT-3 від OpenAI. Це робить Vertex AI привабливим вибором розробки даного застосунку.

Розділ 3. Технічні можливості застосунку

3.1 Login, register

Наявність функцій входу та реєстрації у застосунку є критично важливою, особливо для платних додатків, оскільки це дозволяє забезпечити індивідуальний підхід до кожного користувача та контролювати доступ до ресурсів і сервісів. Реєстрація дозволяє користувачам створювати унікальний обліковий запис, при цьому система не дозволяє використовувати однакові імена користувачів чи адреси електронної пошти. Це сприяє безпеці та індивідуалізації використання застосунку.

Встановлені такі обмеження на реєстрацію: Пароль має містити більше ніж 8 символів, що забезпечує додатковий рівень захисту. Ім'я користувача повинне бути довшим за 4 символи, а адреса електронної пошти повинна слідувати певному шаблону для забезпечення її валідності.

Окрім цього, користувачі в процесі реєстрації обирають свій рівень володіння англійською мовою (див. Рис. 1), що дозволяє адаптувати застосунок до їхніх мовних здібностей.

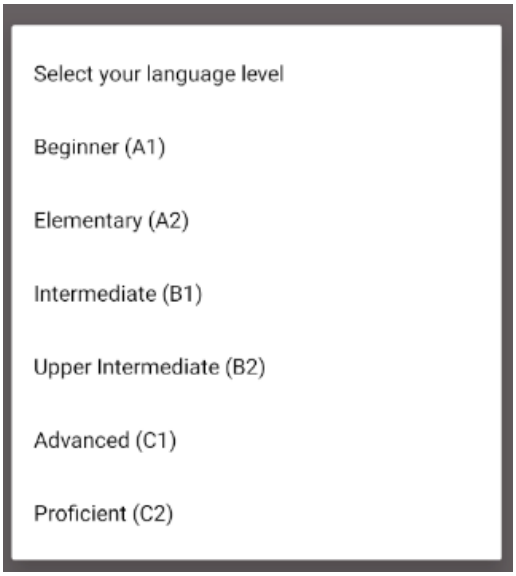

Рисунок 1. Вікно вибору рівня володіння англійською мовою

Для забезпечення безпеки, пароль користувача перетворюється в хеш перед зберіганням у базі даних MongoDB (див. Додаток 1), що гарантує, що навіть у випадку витоку даних, інформація про пароль залишиться захищеною.

```
userSchema.pre('save', async function (next) {
  if (!this.isModified('password')) return next();

  try {
    const salt = await bcrypt.genSalt(10);
    this.password = await bcrypt.hash(this.password, salt);
    return next();
  } catch (err) {
    return next(err);
  }
});
```

Додаток 1. Хешування паролю перед збереженням у базі даних

Після завершення процесу реєстрації користувач перенаправляється на сторінку входу, де він має ввести свої облікові дані для аутентифікації. Після успішного входу в систему ім'я користувача, електронна адреса та рівень знання англійської мови поточного користувача отримуються з бази даних. Ця інформація зберігається в AsyncStorage, щоб її можна було використовувати згодом (див. Додаток 2).

```
await AsyncStorage.setItem('userToken', data.token);
await AsyncStorage.setItem('username', username);
await AsyncStorage.setItem('languageLevel', data.level);
await AsyncStorage.setItem('email', data.email);
```

Додаток 2. Зберігання облікових даних користувача в локальному сховищі

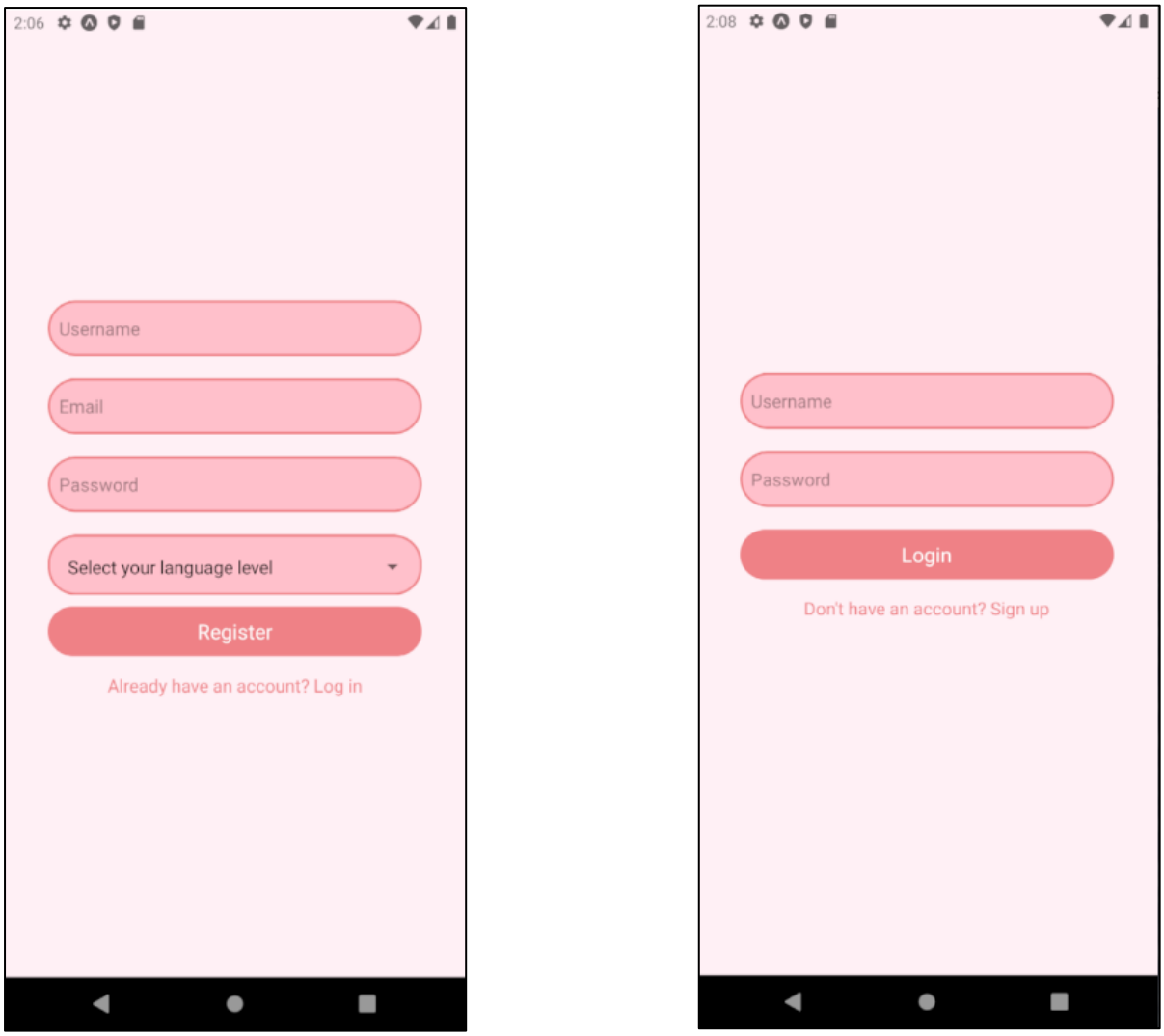

Рисунок 2. Сторінки реєстрації та логіну

3.2 Профіль користувача

Модуль профілю користувача, реалізований у додатку, надає інтерфейс для перегляду та управління особистою інформацією користувача. Користувачі можуть переглядати свої основні дані, такі як ім'я користувача, рівень знання мови, а також завантажити або змінити своє профільне зображення. Дані про користувача зберігаються та відновлюються з локального сховища AsyncStorage.

Функція getUserData виконує асинхронне завантаження даних з AsyncStorage, включаючи ім'я користувача, рівень знання мови та зображення профілю. Якщо користувач має зображення профілю, воно відображається; інакше використовується стандартне зображення no-user-image.

Профіль користувача також містить кнопку для редагування профілю, яка перенаправляє користувача на сторінку редагування профілю.

Компонент включає кнопку 'Log out', яка дозволяє користувачу вийти з системи, очищаючи всі локальні дані та перенаправляючи на сторінку входу.

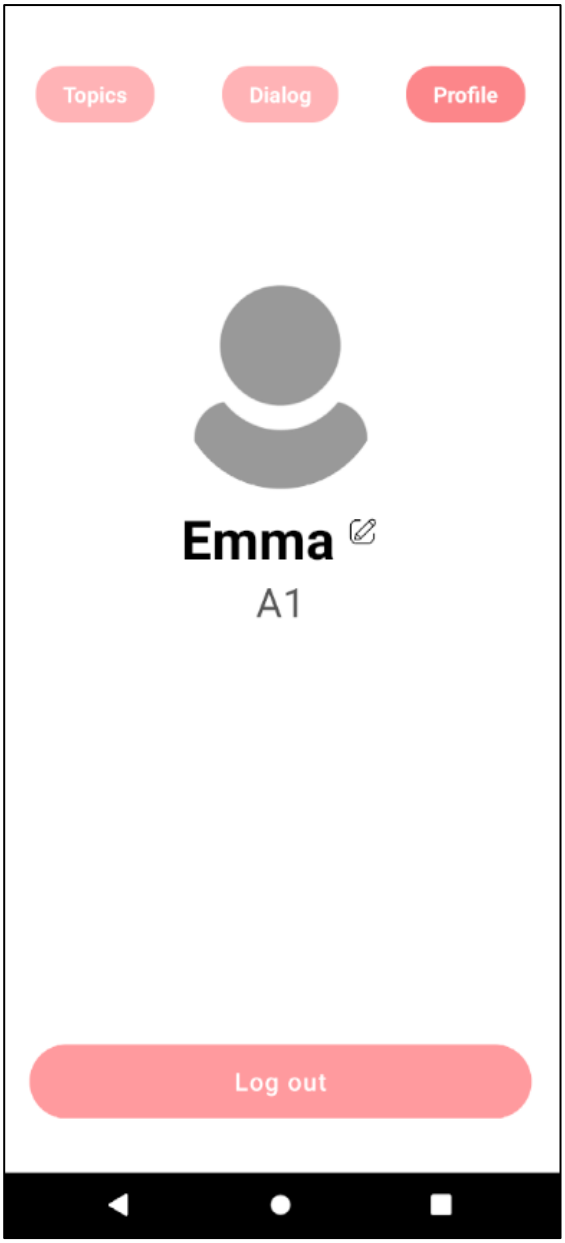

Рисунок 3. Сторінка профілю

3.3 Зміна інформації профілю користувача

Модуль зміни профілю користувача дозволяє користувачам оновлювати свої особисті дані, включаючи зображення профілю, ім'я користувача та електронну адресу.

При зміні зображення профілю використовується бібліотека expo-image-picker, що дозволяє користувачам вибирати нове зображення з галереї свого пристрою. Користувачеві необхідно надати дозвіл для доступу до медіафайлів.

Для імені користувача та електронної пошти встановлені правила валідації, які перевіряються перед відправкою даних на сервер:

- Ім'я користувача має бути унікальним і не співпадати з існуючими в базі. Якщо введене нове ім'я користувача вже існує, користувач отримує повідомлення про те, що обране ім'я вже зайняте.
- Електронна адреса також перевіряється на відповідність формату електронних адрес. Неправильно введена адреса спричиняє виведення помилки. Електронна адреса користувача має бути унікальною і не співпадати з існуючими в базі.

Запит на зміну профілю відправляється на сервер, де дані профілю користувача оновлюються в базі даних. У разі успішного оновлення користувач отримує повідомлення про успіх та перенаправляється на сторінку профілю. Всі зміни зберігаються в AsyncStorage, що дозволяє зберегти актуальність даних при перезапуску додатку.

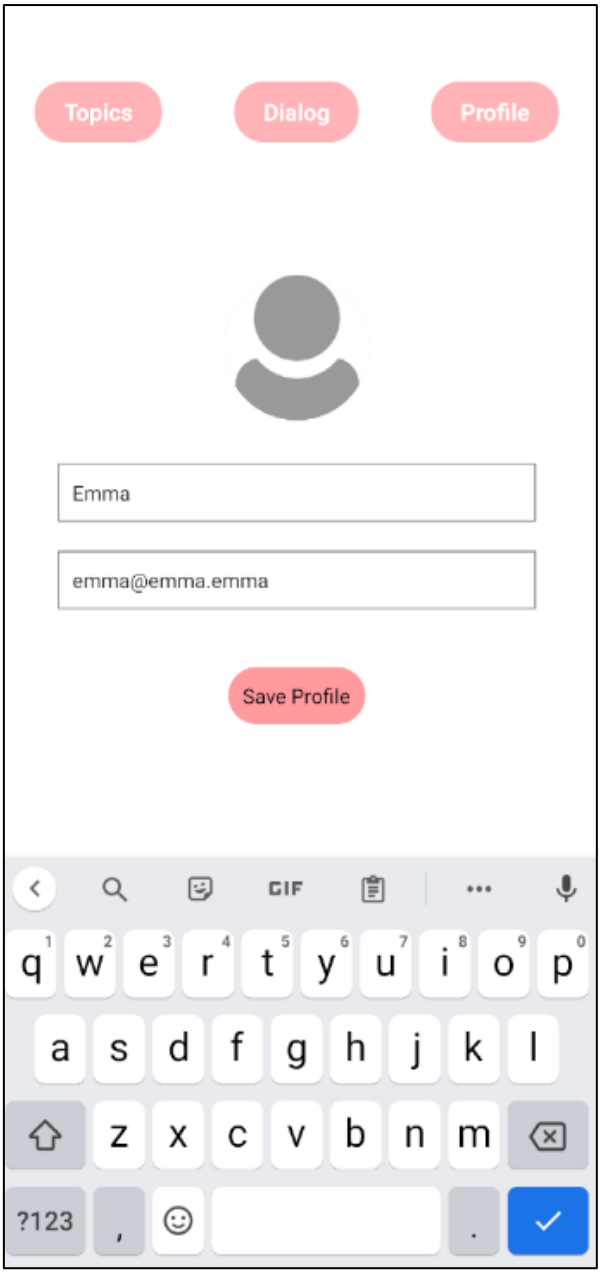

Рисунок 4. Сторінка зміни профілю

3.4 Діалог, який ініціалізується користувачем

Розроблено діалогову систему зі штучним інтелектом, яка ініціюється користувачем і може обговорювати будь-яку тему за вибором користувача. Система включає функціонал перевірки граматики та лексики, який можна вмикати або вимикати залежно від потреб користувача.

В межах розробленої діалогової системи використовується Generative AI on Vertex AI API а саме PaLM 2 (Pathway Language Model 2).

PaLM 2 - це велика мовна модель. [6] Це одна з найпотужніших мовних моделей у світі, здатна виконувати широкий спектр завдань, пов'язаних з обробкою природної мови, включаючи:

- Генерація тексту: PaLM 2 може генерувати реалістичний та зв'язний текст у різних форматах, таких як вірші, код, сценарії, музичні твори, електронні листи тощо.
- Переклад мов: PaLM 2 може точно та природно перекладати текст з однієї мови на іншу.
- Відповіді на запитання: PaLM 2 може давати вичерпні та інформативні відповіді на запитання, навіть якщо вони відкриті, складні або дивні.
- Узагальнення тексту: PaLM 2 може надавати короткі та інформативні резюме довгих текстових документів.
- Написання різних видів творчого контенту: PaLM 2 може писати різні види творчого контенту, такі як вірші, код, сценарії, музичні твори, електронні листи, листи тощо.

"Text-bison" - це модель з сімейства PaLM 2, яка спеціально налаштована для діалогових систем. Це означає, що вона має кращі можливості для ведення розмов, ніж інші моделі PaLM 2. "Text-bison" може

розуміти контекст розмови, генерувати відповідні та цікаві відповіді, а також вести розмову природним та захоплюючим чином.

Як працює модель "text-bison"?

"Text-bison" - це нейронна мережа, навчена на величезному наборі даних тексту та коду. Цей набір даних включає текст з книг, статей, веб-сайтів та коду з GitHub. Нейронна мережа навчається виявляти закономірності в цих даних, а потім використовує ці закономірності для створення нового тексту або коду.

Крім "text-bison", сімейство PaLM 2 пропонує декілька інших моделей:

- Unicorn: Найбільша модель сімейства, ідеальна для складних завдань, таких як програмування та ланцюгового міркування (CoT). Вона має потужні можливості обробки інформації та генерування складних відповідей.
- Bison: "Золота середина" сімейства PaLM. Вона добре справляється з широким спектром мовних завдань, таких як класифікація та підсумовування. Її оптимізовано для забезпечення точності та швидкості роботи за розумною ціною. Bison пропонує інтерфейси для роботи з текстом, чатами, кодом.
- Gecko: Найменша та найдешевша модель, призначена для простих завдань. Вона підходить для випадків, коли не потрібна висока потужність, наприклад, для первинної обробки даних або простих чат-ботів.

Як працює діалогова система?

Користувач вводить своє повідомлення в тектовому полі, яке після обробки передається на сервер за допомогою асинхронного запиту. Цей

запит включає не лише саме питання, але й додаткові параметри, такі як поточний контекст розмови та налаштування перевірки граматики та лексики.

У діалоговій системі існують два перемикачі (switches), які відповідають за перевірку граматики та лексики. Ці налаштування можуть бути активовані або деактивовані користувачем за допомогою графічного інтерфейсу. Стан цих перемикачів передається на бекенд разом із запитом, що дозволяє серверу налаштовувати обробку відповідно до потреб користувача.

Інформація про повідомлення та поточний контекст розмови, а також стан перемикачів перевірки граматики і лексики, передаються на бекенд за допомогою асинхронного запиту. Після отримання відповіді від сервера, нова інформація, така як результати перевірки граматики, лексичні рекомендації, та відповіді на питання, інтегрується в поточну розмову та відображається у користувацькому інтерфейсі.

Поточна розмова зберігається у локальному сховищі (AsyncStorage). Кожен раз, коли користувач вводить нове питання або коли система генерує відповідь, весь контекст розмови оновлюється та перезаписується в AsyncStorage, забезпечуючи збереження історії діалогу.

Функція generateTextResponse на бекенді генерує відповіді у діалоговій системі. Ця функція обробляє запити користувачів, використовуючи контекст попередніх повідомлень у розмові та вказаний мовний рівень користувача. Запит формулюється таким чином, щоб модель могла врахувати як тему розмови, так і мовний рівень, забезпечуючи при цьому природність та релевантність тексту відповіді.

Під час формування запиту до API, generateTextResponse використовує наступний підхід:

З текстових вхідних даних та історії діалогу формується контекст (formattedContext), який включає останн 20 повідомлень (див. Додаток 3). Контекст сформований так, що кожне повідомлення позначається роллю відправника (користувач або AI).

```
const formattedContext = contextWindow.map((msg) =>
`${msg.type === 'question' ? 'User' : 'AI'}:
"${msg.text}"`).join('\n');
```

Додаток 3. Формування контексту розмови

Для формулювання запитів до API використовується спеціалізований текстовий запит (prompt), який формулюється таким чином: користувач вводить своє запитання, яке передається моделі у вигляді:

```
  const requestBody = {
    instances: [
      {
        prompt: `Here's what's been discussed recently: "${formattedContext}".
Now, I'm saying in a casual way, like between friends: "${textQuery}". Reply
in a friendly and short manner.
        Use vocabulary and grammar suitable for a ${languageLevel} level
English learner.
        Never repeat what I wrote. Never repeat yourself from the previous
responses: [${formattedContext}].
        Give only one variant of answer. Always ask a subsequent question at
the end of your reply. Do not include brackets in you responce`,
      },
    ],
  };
```

Додаток 4. Формування тіла запиту для AI для отримання відповіді в діалозі

Цей запит передається моделі, яка аналізує надану інформацію та генерує відповідь, що відображає розуміння контексту дискусії(formattedContext) та мовного рівня користувача(languageLevel).

Результатом роботи функції є JSON-об'єкт із прогнозами (predictions), який передається на фронтенд. Цей об'єкт містить:

- Текстову відповідь AI на запит користувача;
- Результати перевірки граматики, якщо ця опція була активована;
- Пропозиції щодо покращення лексики, якщо це було запитано користувачем.

Для перевірки граматики тексту використовується функція checkGrammar, яка приймає текст від користувача як аргумент. Запит формулюється так, що модель перевіряє текст на наявність граматичних помилок і повертає виправлений варіянт. Якщо граматичних помилок немає, повертається символ "&".

Функція checkVocabulary використовується для підвищення лексичного рівня тексту. Вона аналізує подані речення і пропонує альтернативні варіянти слів або фраз, краще підходящі для вказаного рівня володіння мовою.

```
  const prompt = `Please check and correct the grammar of
the following sentence: "${text}". If there are any
errors, give me only the corrected sentence. If there are
no gramatical errors, write only symbol &`;
```

Додаток 5. Формування тіла запиту для AI для перевірки граматики

```
const prompt = `Language Level: ${languageLevel}. If
exists, suggest a better vocabulary for the following
sentence: "${text}". Write only the suggestion, and only
one`;
```

Додаток 6. Формування тіла запиту для AI для пропозиції покращення лексики

Отже, у розробленій системі діалогу виконуються функції, які дозволяють користувачеві ініціювати розмову на будь-яку тему, з можливістю включення перевірки граматики та лексики (див. Рис. 2 та Рис. 3). Система використовує сучасні технології AI на основі Generative AI on Vertex AI API, зокрема модель PaLM 2.

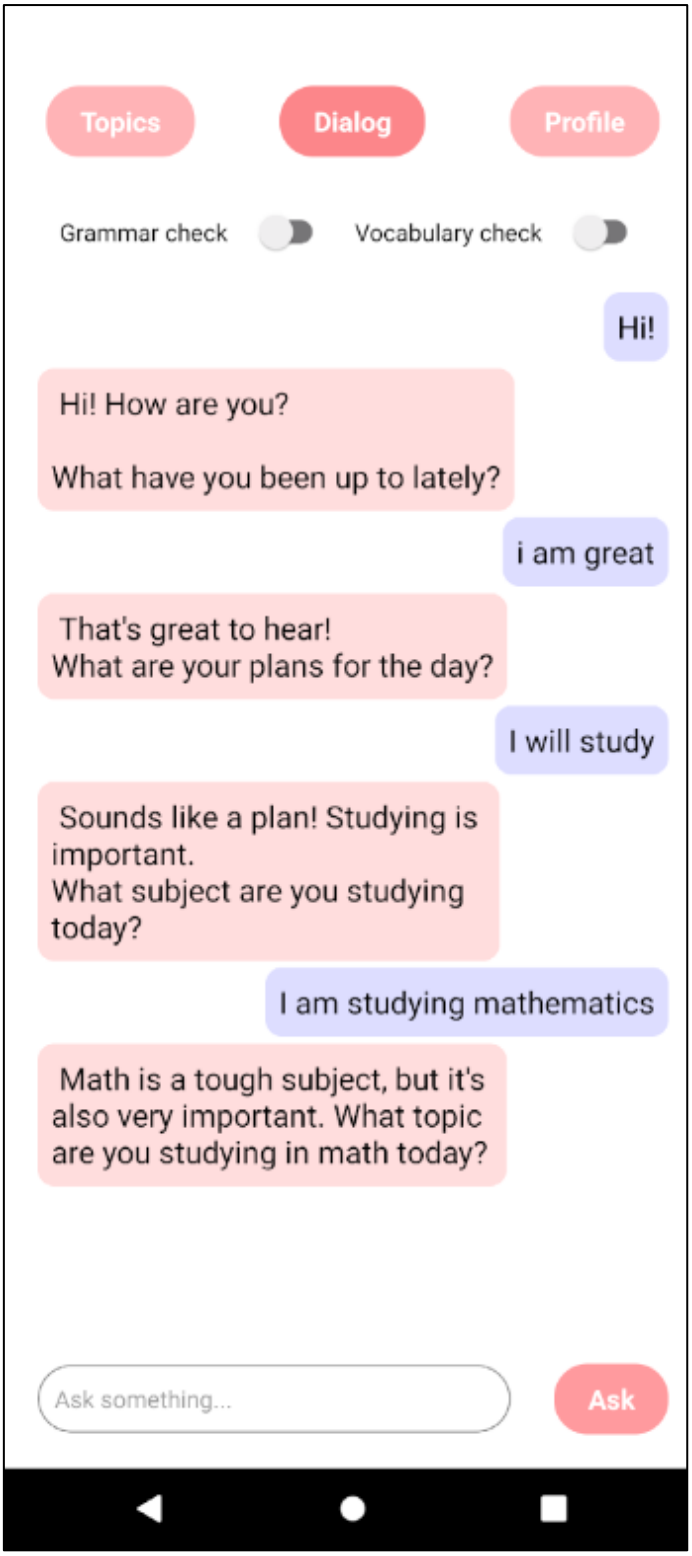

Рисунок 5. Діалог користувача рівня A1 з вимкнутими режимами перевірки граматики та лексики

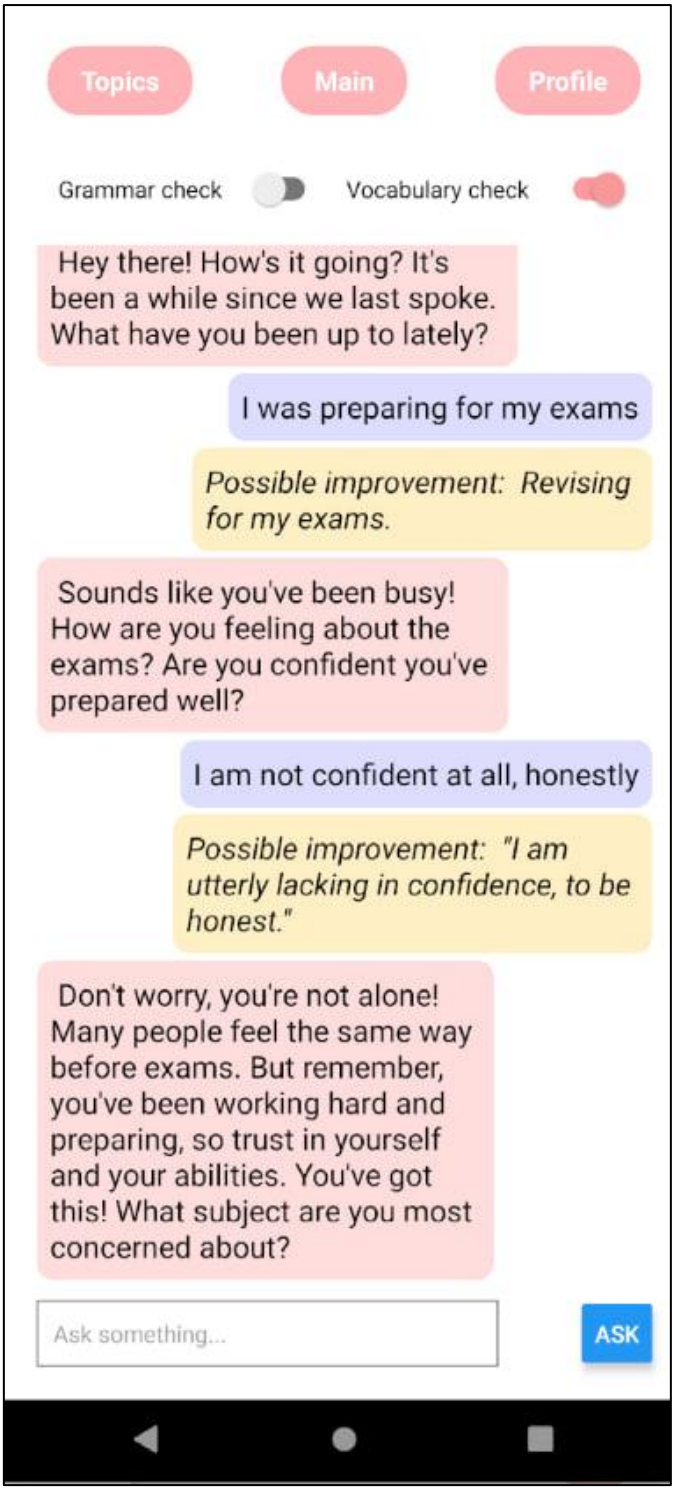

Рисунок 6. Діалог користувача рівня C1 з вимкнутим режимом перевірки граматики та з вімкнутим режимом перевірки лексики

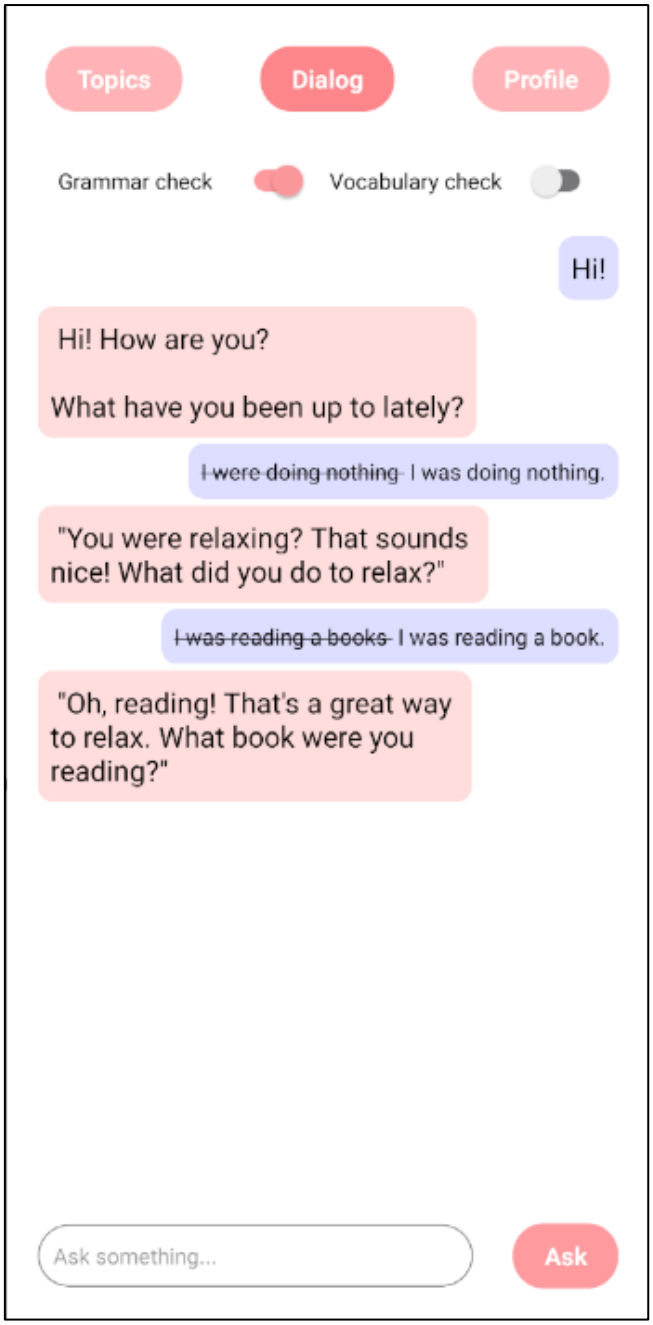

Рисунок 7. Діалог користувача рівня A2 з вімкнутим режимом перевірки граматики та з вимкнутим режимом перевірки лексики

3.5 Діалоги на вибрану тему

Імплементовано функціонал ведення тематичних діалогів, що дозволяє користувачам обговорювати специфічні питання або теми за їх вибором. Початковий вступ до теми генерується системою автоматично на основі вибраної користувачем теми.

Теми для діалогів у застосунку витягуються з бази даних, де вони зберігаються як окремі записи. Це реалізовано за допомогою моделі, яка в MongoDB визначається схемою з полями, такими як рівень складності мови та сама тема обговорення.

```
import mongoose from 'mongoose';

const topicsSchema = new mongoose.Schema({
  level: String,
  topic: String,
});

const Topics = mongoose.model('topic', topicSchema);
export default Topic;
```

Додаток 7. Схема бази даних для тем діалогів

Всі теми рівня поточного авторизованого користувача відображаються на сторінці Topics. До кожної теми існує кнопка, яка ініціалізує діалог на відповідну тему.

Система автоматично генерує вступне питання, пов'язане з вибраною темою, щоб почати бесіду. Після цього користувач може продовжувати діалог.

Тематичні діалоги також включають можливості перевірки граматики та лексики, які є доступними у діалогах на випадкові теми. Ці функції можуть бути активовані або деактивовані користувачем в будь-який час.

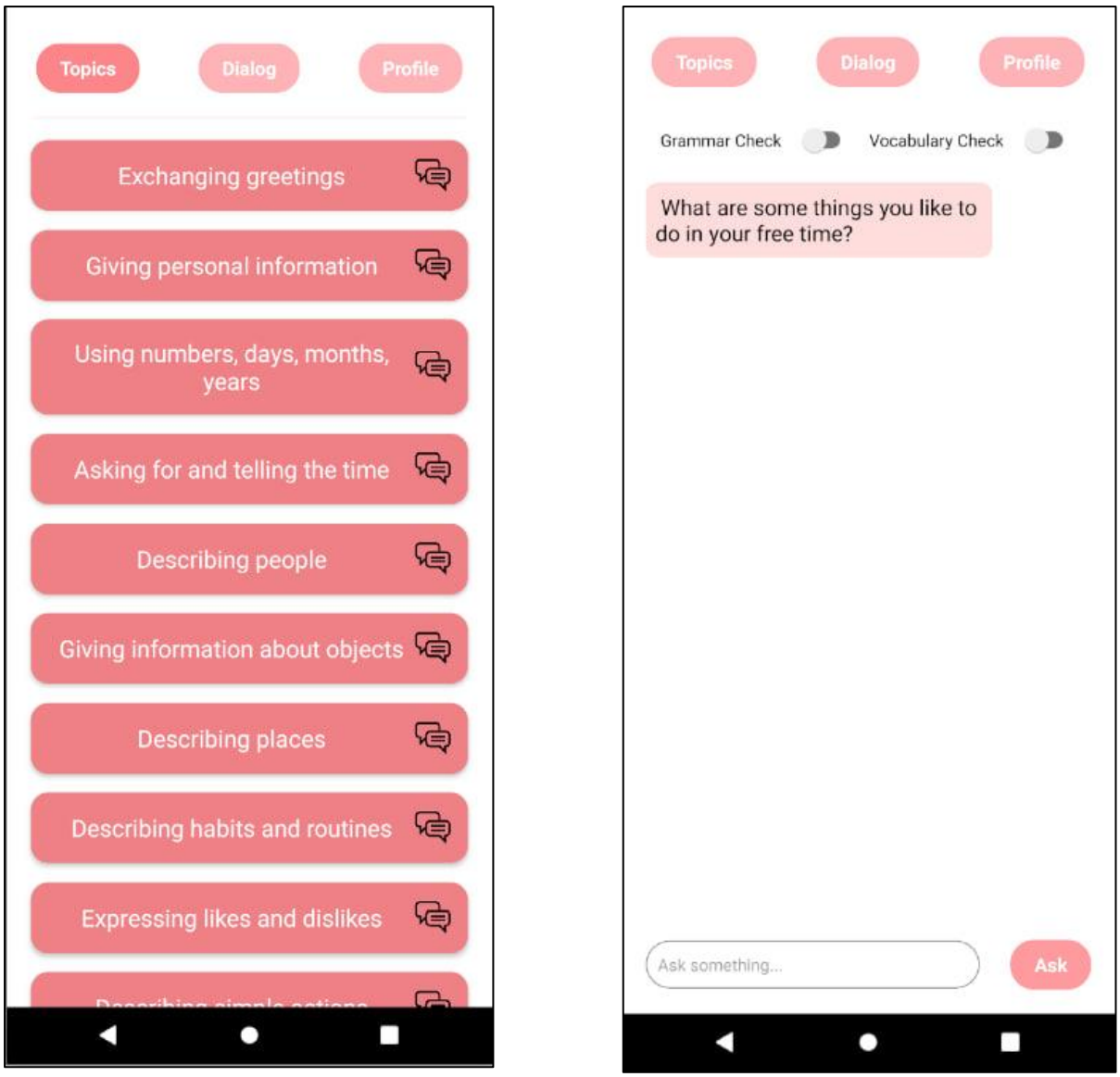

Рисунок 8. Функціонал діалогу а вибрану тему

Розділ 4. Опис бази даних

Для зберігання даних у даному застосунку обрано MongoDB, нереляційну базу даних, яка підтримує високу гнучкість та зручність у роботі з великими об'ємами неструктурованих або слабоструктурованих даних. Однією з головних переваг MongoDB є її "схема без схеми", що дозволяє з легкістю модифікувати дані без перебудови цілої бази даних. Це робить MongoDB зручним якщо знадобитися швидка адаптація до змін у структурі даних.

Наразі в базі зберігаються профілі користувачів та теми для діалогів. Профілі включають інформацію про ім'я користувача, електронну пошту, пароль, рівень знання мови та зображення профілю. Щоб забезпечити безпеку, паролі користувачів шифруються перед збереженням. MongoDB підтримує унікальність імені та електронної пошти, що запобігає дублюванню облікових записів.

Вибір MongoDB замість реляційної бази даних зумовлений потребою в горизонтальному масштабуванні та високій продуктивності при обробці великих даних. Реляційні бази даних, які масштабуються переважно вертикально, можуть виявитися менш ефективними при зберіганні та обробці великих обсягів інформації. MongoDB, з іншого боку, дозволяє розподіляти дані через шардінг, забезпечуючи кращу продуктивність та доступність[7].

У майбутньому планується розширити функціональність бази даних за рахунок зберігання відгуків на діалоги та самі діалоги, що дозволить аналізувати взаємодії користувачів та покращувати систему. Це сприятиме розвитку аналітичних інструментів всередині застосунку і покращенню самих алгоритмів штучного інтелекту.

Розділ 5. Ресурси тем до відповідних рівнів володіння мовою

Під час розробки діалогової системи для мобільного застосунку, важливо забезпечити адаптацію тем діалогів до різних рівнів володіння англійською мовою користувачів. Теми для діалогів, розділені за рівнями, були взяті з ресурсу English Radar, який є надійним джерелом для вивчення англійської мови.

English Radar використовує Спільну європейську мовну рамку (CEFR) для класифікації рівнів знання мови. CEFR є міжнародно визнаним стандартом для опису здобуття мовних навичок, що поділяє знання мови на три основні рівні: початковий (A), середній (B) та високий (C), кожен з яких має два підрівні: 1 і 2. Цей фреймворк дозволяє стандартизувати вимірювання мовних здібностей на міжнародному рівні, забезпечуючи точне і зрозуміле представлення володіння мовою.

Використання ресурсу, адаптованого до CEFR, гарантує, що кожна тема діалогу відповідає конкретному рівню мовної компетенції, що допомагає користувачам застосунку ефективно практикувати та вдосконалювати свої мовні навички відповідно до їхнього поточного рівня.

Висновки

У результаті розробки цієї курсової роботи було створено мобільний додаток, що сприяє практиці розмовної англійської мови через діалоги з використанням штучного інтелекту. Використання таких технологій, як React Native, Node.js та Generative AI on Vertex AI, забезпечило гнучкість та масштабованість проекту, роблячи його доступним для широкої аудиторії користувачів. Це стало можливим завдяки вибору розробки саме мобільного додатку, який можна використовувати у будь-якому місці та в будь-який час, тим самим значно розширюючи можливості для навчання. Також було інтегровано перевірку лексики і граматики з використанням штучного інтелекту.

Ми живемо в епоху штучного інтелекту, яка невпинно трансформує всі сфери нашого життя. Розвиток технологій та їхня інтеграція в освітній процес відкривають нові можливості для підвищення ефективності самоосвіти. Розробка віртуального співрозмовника для практики англійської мови, є вагомим прикладом використання штучного інтелекту для забезпечення гнучкості та доступності освітнього процесу.

Проект демонструє, як розуміння та прийняття штучного інтелекту можуть відігравати важливу роль у сучасному освітньому середовищі. Ми повинні не лише звикати до присутності ШІ у нашому житті, але й активно вивчати можливості його використання для покращення особистісного та професійного розвитку. Враховуючи динамічний розвиток технологій, ключовим аспектом стає постійне оновлення знань та вмінь, щоб ефективно адаптуватися та використовувати новітні досягнення у своїх цілях і завданнях.

Джерела

1. W3Schools Introduction to Node.js. URL: Node.js Introduction (w3schools.com)
2. Офіційний веб-сайт React Native. URL: React Native · Learn once, write anywhere
3. Стаття на тему MongoDB на сайті IBM. URL: What Is MongoDB? | IBM
4. Використання аналітики тексту з MMLSpark у Azure Synapse Analytics: Покроковий навчальний посібник. URL: Tutorial: Text Analytics with Azure AI services - Azure Synapse Analytics | Microsoft Learn
5. Отримання прогнозів за допомогою Vertex AI: Покроковий навчальний посібник. URL: Get predictions from a custom trained model | Vertex AI | Google Cloud
6. Застосування генеративного штучного інтелекту з моделями PaLM 2 та LangChain на Vertex AI. URL: Generative AI applications with Vertex AI PaLM 2 Models and LangChain | Google Cloud Blog
7. Документація MongoDB: Шардінг. URL: Sharding - MongoDB Manual v7.0