

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістрський)

Розробка системи для аналізу даних платформи вивчення іноземних мов
(тема)

Виконав:
студент II курсу, групи СПРМ-22-1
Кошарний Є. Ю.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
Освітня програма Системне проектування
Керівник проф. каф. СТ Перова І.Г.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Гребеннік І. В.
(прізвище, ініціали)

2024 р.

Я як студент(ка) ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

10.06.2024



Кошарний Є.Ю.

(дата, підпис, прізвище студента/-ки)

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Керівник кваліфікаційної роботи



проф. каф. СТ Перова І.Г.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Керівник кваліфікаційної роботи



проф. каф. СТ Перова І.Г.

Попередній захист проведено 10 червня 2024 р.

Керівник кваліфікаційної роботи



проф. каф. СТ Перова І. Г.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Системотехніки

Рівень освіти другий (магістрський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«__» _____ 20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ


студентові Кошарному Євгенію Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка системи для аналізу даних платформи вивчення іноземних мов» затверджена наказом університету від «1» квітня 2024 р. № 259 Ст
2. Термін подання студентом роботи до екзаменаційної комісії «8» червня 2024 р.
3. Вихідні дані до роботи Об'єкт дослідження – процес надання освітніх послуг із вивчення іноземних мов, збір та аналіз даних платформи для вивчення іноземних мов, інформаційні технології та програмні методи створення системи аналізу даних, клієнтської і серверної частини інформаційної системи платформи вивчення іноземних мов. Сфера застосування – підтримка можливості прийняття рішень заснованих на аналізі даних платформи для вивчення іноземних мов. Технічне забезпечення: ІВМ-сумісний персональний комп'ютер. Програмний засіб з віконно-графічним інтерфейсом.
4. Перелік питань, що потрібно опрацювати в роботі Вступ. Аналіз предметної області. Аналіз предметної області, опис об'єкта та задачі проектування. Аналіз ринку та існуючих рішень. Визначення сфери застосування розроблюваної системи. Постановка задачі. Формування вимог до ІС. Визначення основних функціональних та нефункціональних вимог до вебдодатку та системи аналізу даних. Визначення функцій інтерфейсу клієнтської частини вебсайту. Створення дизайну вебдодатку. Прогнозування сценаріїв взаємодії користувачів із вебдодатком. Категоризація користувачів і визначення їхніх дозволів і привілеїв у системі. Проектування ІС. Проектування архітектури системи аналізу даних та вебдодатку. Дослідження та вибір найбільш прийнятних технологій для розробки ІС. Визначення сутностей і зв'язків у системі та проектування операційної бази даних. Визначення подій користувача для відправки в систему аналізу даних. Розробка та впровадження ІС. Програмна реалізація вебдодатку. Розгортання вебдодатку. Побудова та розгортання системи аналізу

даних. Покриття компонентів системи тестами. Висновки. Перелік джерел посилання.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій: Організаційна структура мовної школи (1аркуш формату А4). Use case діаграма MVP версії вебдодатку мовної школи (1аркуш формату А4). Дворівнева клієнт-серверна архітектура (1аркуш формату А4). Трирівнева клієнт-серверна архітектура (1аркуш формату А4). Загальна архітектура системи аналізу даних платформи для вивчення іноземних мов (1аркуш формату А4). Архітектура системи з обраними технологіями (1аркуш формату А4). ER-діаграма MVP версії операційної бази даних (1аркуш формату А4). Вебдодаток публічно доступний за доменним ім'ям (1аркуш формату А4). Звіт про відвідуваність вебсайту, частина 1 (1аркуш формату А4). Звіт про відвідуваність вебсайту, частина 2 (1аркуш формату А4).

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
<i>Розділи спеціальної частини</i>	<i>проф. каф. СТ Перова І.Г.</i>		07.06.2024

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	<i>Отримання завдання на виконання роботи</i>	01.04. 2024	<i>Виконано</i>
2	<i>Аналіз предметної області, опис об'єкта та задачі проектування</i>	10.04.2024	<i>Виконано</i>
3	<i>Аналіз ринку та існуючих рішень</i>	14.04.2024	<i>Виконано</i>
4	<i>Визначення сфери застосування розроблюваної системи</i>	20.04.2024	<i>Виконано</i>
5	<i>Постановка задачі</i>	25.04.2024	<i>Виконано</i>
6	<i>Формування вимог до ІС</i>	01.05.2024	<i>Виконано</i>
7	<i>Проектування ІС</i>	10.05.2024	<i>Виконано</i>
10	<i>Програмна реалізація вебдодатку</i>	18.05.2024	<i>Виконано</i>
11	<i>Побудова та розгортання системи аналізу даних</i>	25.05.2024	<i>Виконано</i>
15	<i>Представлення на рецензування</i>	07.06.2024	<i>Виконано</i>

Дата видачі завдання «01» квітня 2024 р.

Студент


(підпис)

Кошарний Є. Ю.
(прізвище, ініціали)

Керівник роботи


(підпис)

проф. каф. СТ Перова І.Г.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи магістра: 90 с., 61 рис., 0 табл., 6 джерел інформації.

СИСТЕМА АНАЛІЗУ ДАНИХ, БАЗА ДАНИХ, ІНОЗЕМНІ МОВИ, DATA PIPELINE, DATA WAREHOUSE, REACT, PYTHON.

Об'єктом дослідження в роботі є процес надання освітніх послуг із вивчення іноземних мов, збір та аналіз даних платформи для вивчення іноземних мов.

Предметом досліджень є інформаційні технології та програмні методи створення системи аналізу даних, клієнтської і серверної частини інформаційної системи платформи вивчення іноземних мов.

Мета роботи - розробка системи для аналізу даних з платформи для вивчення іноземних мов.

Методи дослідження – системний підхід, методи структурного аналізу, об'єктно-орієнтованого аналізу та проектування, моделювання даних.

Сфера застосування – підтримка можливості прийняття рішень заснованих на аналізі даних платформи для вивчення іноземних мов.

ABSTRACT

An explanatory note to the qualification work of a master: 90 pages, 61 figure, 0 tables, 6 sources.

DATA ANALYSIS SYSTEM, DATABASE, FOREIGN LANGUAGES, DATA PIPELINE, DATA WAREHOUSE, REACT, PYTHON.

The object of research is the process of providing educational services for learning foreign languages, collecting and analyzing data from a foreign language learning platform.

The subject of research is information technologies and software methods for creating a data analysis system, client and server parts of the information system of a foreign language learning platform.

The purpose of the qualification work is to develop a system for analyzing data from a foreign language learning platform.

Research methods: systematic approach, methods of structural analysis, object-oriented analysis and design, data modeling.

Scope of the research is support for decision-making based on data analysis of the foreign language learning platform.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1 Аналіз предметної області, опис об'єкта та задачі проектування.....	12
1.2 Аналіз ринку та існуючих рішень.....	19
1.3 Визначення сфери застосування розроблюваної системи.....	26
1.1 Постановка задачі.....	28
2 ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	30
2.1 Визначення основних функціональних та нефункціональних вимог до вебдодатку та системи аналізу даних.....	30
2.2 Визначення функцій інтерфейсу клієнтської частини вебсайту.....	32
2.3 Створення дизайну вебдодатку.....	33
2.4 Прогнозування сценаріїв взаємодії користувачів із вебдодатком.....	44
2.5 Категоризація користувачів і визначення їхніх дозволів і привілеїв у системі 47	
3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	49
3.1 Проектування архітектури системи аналізу даних та вебдодатку.....	49
3.2 Дослідження та вибір найбільш прийнятних технологій для розробки ІС...52	
3.3 Визначення сутностей і зв'язків у системі та проектування операційної бази даних.....	58
3.4 Визначення подій користувача для відправки в систему аналізу даних.....	60
4 РОЗРОБКА ТА ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	61
4.1 Програмна реалізація вебдодатку.....	61
4.2 Розгортання вебдодатку.....	69
4.3 Побудова та розгортання системи аналізу даних.....	74

4.4 Покриття компонентів системи тестами.....	83
ВИСНОВКИ.....	86
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	88
Додаток А.....	91

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ETL – Extract Transform Load;

БД – база даних;

ІС - інформаційна система;

MVP – minimum viable product (мінімально життєздатний продукт);

Python – мова програмування;

JavaScript – мова програмування;

Next.js – веб фреймворк;

IP – Internet Protocol;

SQL – Structured Query Language (мова структурованих запитів).

ВСТУП

У сучасному суспільстві Інтернет став доступний для великої кількості людей, і майже кожен має можливість підключитися до мережі. Цей зв'язок перетворився на фундаментальний стовп нашого повсякденного життя, пронизуючи різні аспекти нашого існування. Крім того, перехід бізнесу в онлайн-сферу став неминучою тенденцією, зумовленою усвідомленням його ефективності як інструменту просування та платформи для пропонування різноманітних послуг. Завдяки онлайн-платформам компанії можуть не лише залучати нових клієнтів, але й оптимізувати повторювані завдання, контролювати активи, полегшувати транзакції та навіть забезпечувати дистанційне навчання для учнів по всьому світу. Таким чином, Інтернет є багатограним інструментом, який не тільки покращує зв'язок, але й сприяє ефективності та інноваціям у різних сферах людської діяльності. Суттєва кількість людей повністю відмовилась від оффлайн навчання, аргументуючи це економією часу на переміщеннях або, як наприклад, у часи пандемії з метою зниження ризиків захворіти та зменшення контактів між людьми. Тому різні види бізнесів мають реагувати на такий запит і пропонувати нові можливості для онлайн навчання, надання інших послуг, купівлі товарів в інтернеті тощо. Оскільки популярність вебдодатків продовжує зростати, компанії все більше схильні заглиблюватися в розуміння поведінки своїх відвідувачів і клієнтів. Це передбачає отримання інформації про їхню активність на платформі, виявлення закономірностей та розуміння демографічних даних користувачів, щоб адаптувати персоналізований досвід та оптимізувати маркетингові стратегії.

У контексті цієї кваліфікаційної роботи основною метою є розробка надійного вебдодатків для платформи вивчення іноземних мов. Водночас, основна увага приділяється створенню комплексної системи аналізу даних, щоб надати мовній школі можливості прийняття рішень на основі даних. Використовуючи створену систему, школа іноземних мов зможе сприяти підвищенню рівня

залучення користувачів, покращенню навчальних методик та загальної продуктивності платформи.

Актуальність теми проекту є надзвичайно важливою, особливо в світлі зростаючої тенденції до дистанційного навчання. Зі збільшенням кількості людей, які обирають онлайн-навчання, попит на ефективні та персоналізовані платформи для вивчення мови зростає. Більше того, у нинішній ситуації цифрова присутність через вебсайт стала стандартною практикою майже для кожного малого бізнесу. Такі онлайн-платформи не тільки служать точкою взаємодії з клієнтами, але й вселяють довіру потенційним клієнтам.

Проте простого існування вебсайту вже недостатньо для стабільного розвитку. Щоб по-справжньому масштабувати бізнес і приймати обґрунтовані стратегічні рішення, вкрай важливо використовувати потужність даних. Збираючи, аналізуючи та інтерпретуючи величезні набори даних, що охоплюють усі аспекти бізнесу, організації можуть отримати дуже цінне розуміння про ефективність свого бізнесу. Ця інформація прокладає шлях до оптимізації операцій, цільових маркетингових кампаній і покращення взаємодії з користувачем. У сфері онлайн освіти розуміння поведінки користувачів є необхідним для вдосконалення методології навчання, адаптації контенту та сприяння взаємодії.

Таким чином розробка надійної вебплатформи для мовної школи у комбінації із створенням системи аналізу даних має значні перспективи. Інтегруючи ці компоненти, мета полягає в тому, щоб надати мовній школі інструмент, необхідний для прийняття стратегічних рішень, керуючись наявними даними, що зрештою сприятиме зростанню та зміцненню конкурентної переваги в мінливому середовищі дистанційної освіти та цифрових бізнес-операцій.

Об'єктом дослідження є бізнес-процес надання освітніх послуг із вивчення іноземних мов, збір, трансформація та аналіз даних платформи з платформи іноземних мов.

Предметом дослідження є інформаційні технології та програмні методи створення системи аналізу даних, клієнтської і серверної частини інформаційної системи платформи вивчення іноземних мов.

Проте, щоб впровадження такого програмного продукту пройшло гладко, необхідно враховувати численні зовнішні та внутрішні змінні. Отже, методології та технології моделювання, а також алгоритми, розроблені для аналізу атрибутів і поведінки складних об'єктів і систем, є обов'язковими. Ці ресурси спрямовані на всебічне вивчення об'єкта, його функціонування та обґрунтування доцільності створення такого програмного продукту.

Метою даної роботи є розробка системи аналізу даних платформи вивчення іноземних мов та створення Web-додатку з якого буде збиратися інформація.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області, опис об'єкта та задачі проектування

Предметна область є пов'язаною з обраною темою розробки системи для аналізу даних із платформи для вивчення іноземних мов, і обертається навколо операційної структури мовної онлайн-школи. Передбачена інформаційна система мовної школи розроблена таким чином, щоб задовольнити різноманітні функції, охоплюючи різні аспекти управління групами студентів, виконання навчальної програми та залучення користувачів.

За своєю суттю, кінцева версія інформаційної системи має на меті втілити такі важливі функції, як реєстрація та вхід у систему, що полегшує безперебійний доступ для студентів, викладачів та адміністративного персоналу. Крім базової аутентифікації, вона включає надійні можливості керування користувачами, що дозволяє створювати та керувати групами студентів, профілями співробітників, призначеннями викладачів і окремими організаційними відділами. Функція призначення ролей додатково вдосконалює привілеї доступу, гарантуючи, що взаємодія кожного користувача з платформою узгоджується з його визначеними обов'язками та дозволами. Головним у функціональності системи є її здатність керувати навчальним планом і складати розклад уроків. Завдяки інтуїтивно зрозумілим інтерфейсам адміністратори можуть планувати уроки, розподіляти ресурси та ефективно поширювати навчальні матеріали. Крім того, система полегшує розміщення додаткових матеріалів і освітніх блогів, збагачуючи досвід навчання та сприяючи безперервній взаємодії.

Рішення обрати дистанційне навчання впливає з безлічі вагомих факторів, кожен з яких сприяє зростанню популярності такого способу навчання. По-перше, зручність виділяється як основний мотиватор, оскільки дистанційне навчання усуває потребу у фізичній присутності, тим самим економлячи дорогоцінний час, який раніше витрачався на дорогу. Цей аспект став особливо актуальним на тлі викликів, пов'язаних із такими подіями, як нещодавна глобальна пандемія та

війна в Україні. Оскільки традиційні методи навчання порушені, дистанційне навчання пропонує стійку альтернативу, дозволяючи людям продовжувати отримувати нові знання, не виходячи з безпеки та комфорту власних домівок.

Компанії також визнають переваги дистанційної моделі. Переходячи на дистанційну роботу, організації можуть значно скоротити накладні витрати, пов'язані з орендою офісу, комунальними послугами та обслуговуванням. Цей зсув не тільки призводить до значної економії коштів, але й сприяє більшій гнучкості в управлінні персоналом. Організація віддаленої роботи дає можливість співробітникам працювати з будь-якого місця, долаючи географічні кордони та дозволяючи компаніям використовувати ширший пул талантів.

Крім того, модель дистанційного навчання та роботи сприяє екологічній стійкості за рахунок зменшення викидів вуглецю, пов'язаних із щоденними поїздками та офісними приміщеннями. Це узгоджується зі зростаючим акцентом на корпоративній соціальній відповідальності та ініціативах сталого розвитку в різних галузях. Дистанційне навчання та дистанційна робота забезпечують невід'ємну гнучкість, дозволяючи людям адаптувати свій графік відповідно до особистих зобов'язань і вподобань. Ця гнучкість не тільки покращує баланс між роботою та особистим життям, але також сприяє продуктивності та задоволенню роботою.

По суті, численні переваги, пов'язані з дистанційним навчанням і віддаленою роботою, змінюють освітній ландшафт і переосмислюють традиційні уявлення про освіту. Оскільки як окремі особи, так і компанії продовжують використовувати ці моделі, тенденція до дистанційного навчання збережеться й розвиватиметься в найближчі роки.

Основний фокус у цій кваліфікаційній роботі приділено функції відстеження подій та взаємодії із вебсайтом, яка записує дії користувачів на платформі у пайплайн даних. Від відвідувань сторінок до відтворення відео та взаємодії з іншими функціями, ці події реєструються та передаються в конвеєр даних для подальшого аналізу. Цей детальний рівень відстеження дає безцінне розуміння поведінки користувачів, уподобань і моделей навчання, служачи наріжним

каменем для прийняття рішень на основі даних з метою покращення користувацького досвіду, оптимізації навчання та бізнес-процесів.

По суті, інформаційна система мовної школи слугує багатограним центром, організовуючи безперебійну роботу різноманітних освітніх процесів, одночасно сприяючи створенню та аналізу корисних даних. Система дає можливість стейкхолдерам приймати стратегічні рішення базуючись на наявних даних.

Таким чином загалом система є багатокomпонентною і складається із платформи для організації та проведення онлайн навчального процесу та пайплайну для обробки даних платформи з метою отримання інсайтів та їх візуалізації. Враховуючи складність побудови комплексної вебсистеми з такими функціями, як реєстрація/вхід, дозволи на основі ролей, керування контентом, профілі користувачів, блоги, планування уроків і керування групами, очевидно, що для такого проекту зазвичай потрібна спеціальна команда інженерів програмного забезпечення та розширені часові рамки від місяців до навіть років для повного впровадження цих вимог.

Тому, усвідомлюючи обмеження часу та ресурсів, притаманні кваліфікаційній роботі, стратегічна увага зосереджена на визначенні пріоритетів у розробці та розгортанні двох критичних компонентів: системи аналізу даних і мінімально життєздатного продукту (MVP) вебплатформи.

Основний акцент робиться на побудові та розгортанні системи аналізу даних, яка є основою для прийняття обґрунтованих рішень та оптимізації роботи мовної школи. Зосереджуючись на цьому аспекті, кваліфікаційна робота має на меті продемонструвати вміння розробляти та впроваджувати ефективний конвеєр даних, здатний отримувати, обробляти та аналізувати дані про діяльність користувача з платформи вивчення мови.

Паралельно увага спрямована на розробку MVP вебплатформи, хоч і в спрощеному вигляді. MVP має форму цільової сторінки, розробленої для демонстрації важливої інформації про мовну школу, надаючи потенційним користувачам можливість подати запит на пробний урок. Ця спрощена версія вебплатформи служить доказом концепції, пропонуючи зазирнути в

передбачуваний досвід користувача та функціональність, мінімізуючи зусилля на розробку та вкладення часу.

Застосовуючи цей підхід, кваліфікаційна робота встановлює баланс між здійсненністю та амбіціями, дозволяючи продемонструвати основні компетенції як у розробці системи аналізу даних, так і в розгортанні вебдодатків у межах визначених часових рамок і ресурсів. Ця стратегічна спрямованість забезпечує досягнення відчутних результатів, одночасно закладаючи основу для потенційного майбутнього розширення та інтеграції вебплатформи.

Мовна школа є багатогранною сутністю, що охоплює різноманітний набір ролей, важливих для її функціонування. У її структурі є адміністратори, відповідальні за організаційне управління, викладачі, які є основними фасилітаторами навчання, учні, які є одержувачами освіти, експерти з маркетингу, які займаються просуванням школи, бухгалтери, які керують фіскальними справами закладу, розробники програмного забезпечення, що забезпечують функціональність освітніх технологій та СММ-менеджерів, відповідальних за підтримку присутності закладу в соціальних мережах та залучення.

Педагоги відіграють ключову роль у навчальному закладі, взаємодіючи з учнями за допомогою різних методів, включаючи як групові заняття, так і персоналізовані індивідуальні заняття. Ці викладачі адаптують траєкторії навчання відповідно до індивідуальних потреб і вподобань кожного учня, забезпечуючи оптимізацію навчального досвіду для максимальної ефективності та залучення. Завдяки своїй відданості та досвіду викладачі є наріжним каменем місії навчального закладу щодо надання якісної мовної освіти своїм різноманітним клієнтам.

Як було зазначено раніше, наразі майже весь бізнес діджиталізується, переходить в онлайн та збирає, оброблює та аналізує дані для прийняття рішень. Тому треба відповідати цій тенденції створенням системи аналізу даних з вебдодатку платформи для вивчення іноземних мов.

Сформулюємо основну діяльність ІС та зробимо опис об'єктів проектування. Об'єктом проектування є вебдодаток, який є цільовою сторінкою

мовної школи. Його основна діяльність полягає в тому, щоб продемонструвати переваги школи, надати важливу інформацію, викликати інтерес, пропонувати безкоштовні навчальні матеріали та дозволити відвідувачам залишати заявки на пробні уроки. Вебдодаток має на меті ефективно донести ціннісну пропозицію школи, підкреслюючи її методологію викладання, досвід викладачів, доступні мови та гнучкість розкладу. Він містить коротку інформацію про школу, контактні дані, ціни та відгуки студентів. Також пропонуються безкоштовні навчальні матеріали, щоб дати відвідувачам відчуття стилю викладання та навчальну програму школи. Загалом цільова сторінка функціонує для залучення потенційних клієнтів, заохочення інтересу та полегшення процесу запиту на пробні уроки для тих, хто вивчає мову. У базі даних буде зберігатися інформація про залишені користувачами заявки на пробні заняття. Іншим об'єктом проектування є система аналізу даних, яка створена як ETL pipeline та інтегрована з вебдодатком.

Процес використання користувачем платформи може виглядати наступним чином.

1. Користувач переходить на вебсайт (цільову сторінку).
2. Вони читають детальний опис мовної школи, отримуючи уявлення про її пропозиції та підхід.
3. Користувач залучається до відеопрезентації, яка пропонує візуальний огляд школи, надаючи додатковий контекст і привабливість.
4. Вони вивчають розділ, присвячений корисним матеріалам, де вони можуть отримати доступ до таких ресурсів, як зразки уроків, мовні тести або матеріали для завантаження.
5. Нарешті, користувач заповнює форму, розміщену на вебсайті, і надсилає заявку на пробний урок, висловлюючи свою зацікавленість спробувати навчання в школі безпосередньо.

Всі дії користувача на вебсайті генерують події користувача (user events), ці події відправляються в систему для аналізу даних, де вони проходять подальшу обробку та перетворення. Цей конвеєр даних використовує різні методи для очищення, збагачення та агрегування необроблених даних подій, готуючи їх для

аналітичних запитів і генерації аналітичних даних. Використовуючи потужність аналізу даних, мовна школа може отримати глибше розуміння поведінки користувачів, уподобань і моделей взаємодії, таким чином інформуючи про прийняття стратегічних рішень, оптимізуючи взаємодію з користувачами та сприяючи безперервному вдосконаленню платформи.

Коли користувач замовив пробне заняття, то електронний лист з відповідною інформацією відправляється інформаційною системою на електронну пошту школи та визначених викладачів. Далі адміністратор або вчитель сконтактує з учнем та призначить пробне заняття, яке буде проведене на платформі Zoom або Google Meet.

Після успішного бронювання користувачем пробного уроку інформаційна система організовує безперебійний процес спілкування, щоб забезпечити плавний перехід від запиту до взаємодії. Ось як відбувається процес:

1. Створення сповіщень електронною поштою. Коли користувач надсилає форму заявки на пробний урок, інформаційна система запускає генерацію сповіщень електронною поштою. Ці сповіщення надсилаються визначеним одержувачам, зокрема на адміністративну електронну адресу школи та електронні адреси призначених учителів, відповідальних за проведення пробних уроків.

2. Вміст електронного листа. Електронний лист містить відповідну інформацію, пов'язану з замовленим пробним уроком. Це включає такі деталі, як ім'я користувача, контактна інформація, мета вивчення мови, а також будь-які конкретні переваги чи вимоги, зазначені користувачем під час подання заявки. Крім того, електронний лист містить інструкції щодо подальших дій, направляючи одержувача, як продовжити планування пробного уроку.

3. Дії одержувача. Після отримання сповіщення електронною поштою адміністратор або призначений викладач, відповідальний за координацію пробного уроку, вживає заходів. Вони переглядають інформацію користувача та контактні дані, надані в електронному листі, щоб розпочати спілкування з майбутнім студентом.

4. Планування пробного уроку. Адміністратор або вчитель зв'язується з користувачем електронною поштою, телефоном, в месенджері, щоб встановити контакт і обговорити розклад пробного уроку. Вони узгоджують взаємно зручний часовий проміжок на основі доступності користувача та розкладу вчителя.

5. Вибір платформи. Після запланованого пробного уроку адміністратор або вчитель підтверджує платформу для проведення уроку. Залежно від уподобань школи та технологічної інфраструктури, пробні уроки зазвичай проводяться на популярних платформах для відеоконференцій, таких як Zoom або Google Meet.

6. Проведення уроку. У запланований час пробний урок відбувається на призначеній платформі для відеоконференцій. Учитель проводить урок, знайомлячи користувача з середовищем вивчення мови, оцінюючи рівень його володіння та надаючи зразок навчального досвіду, адаптований до його потреб і вподобань.

Організуючи цей скоординований процес спілкування та використовуючи цифрові інструменти для проведення уроків, мовна школа забезпечує безперебійний та персоналізований досвід для користувачів, зацікавлених у дослідженні можливостей вивчення мови. Завдяки ефективній комунікації та взаємодії школа створює міцну основу для побудови тривалих стосунків з майбутніми студентами та виховання культури постійного навчання та зростання.

Система аналізу даних розроблена, щоб надати зацікавленим сторонам всебічне розуміння за допомогою візуалізацій, що стосуються різних ключових показників, пов'язаних із продуктивністю вебсайту мовної школи. Ці відомості включають: обсяг трафіку вебсайту, запити на пробні уроки, коефіцієнт конверсії, додаткові релевантні показники.

Представляючи зацікавленим сторонам цю візуалізовану інформацію, система аналізу даних дає їм змогу приймати обґрунтовані рішення, оптимізувати стратегії та підвищити загальну ефективність присутності мовної школи в Інтернеті.

Це являє собою ключовий аспект, який формує успішність компанії та створює прихильність клієнтів. Спеціальний відділ в організації збирає та

ілюструє статистичні дані. Організаційну структуру компанії та функції, які кожен відділ виконує в її діяльності буде розглянуто далі.

1.2 Аналіз ринку та існуючих рішень

На цьому етапі було проведено аналіз ринку та розглянуто приклади існуючих інформаційних систем платформ для вивчення іноземних мов. Оцінити системи аналізу даних конкурентів неможливо, бо це не загальнодоступні системи, тому фокус направлено саме на аналіз вебдодатків.

Було вивчено приклади існуючих інформаційних систем для платформ вивчення іноземних мов, щоб зрозуміти їхні особливості, дизайн, функціональні можливості та досвід користувача. Вивчаючи ці вебдодатки, ми отримали уявлення про галузеві тенденції, найкращі практики та інновації.

Аналіз мав на меті виявити сильні та слабкі сторони поточних пропозицій, розпізнати переваги і проблемні точки, а також виявити можливості для вдосконалення на ринку. Цей процес дав цінну інформацію для розробки та вдосконалення власного вебдодатку та інформаційної системи мовної школи. На рисунках 1.1 та 1.2 показано результат пошуку в Google за запитом «вчити англійську онлайн».

Ми бачимо топ сайтів, які пропонують вивчення англійської онлайн. Перші результати це “Preply”, “Englishdom” та “Duolingo”. Ці три варіанти взято для подальшого аналізу.

Під час аналізу буде надано короткий опис особливостей бізнесу, дизайну вебсторінки, швидкості завантаження та оцінку користувацького досвіду.

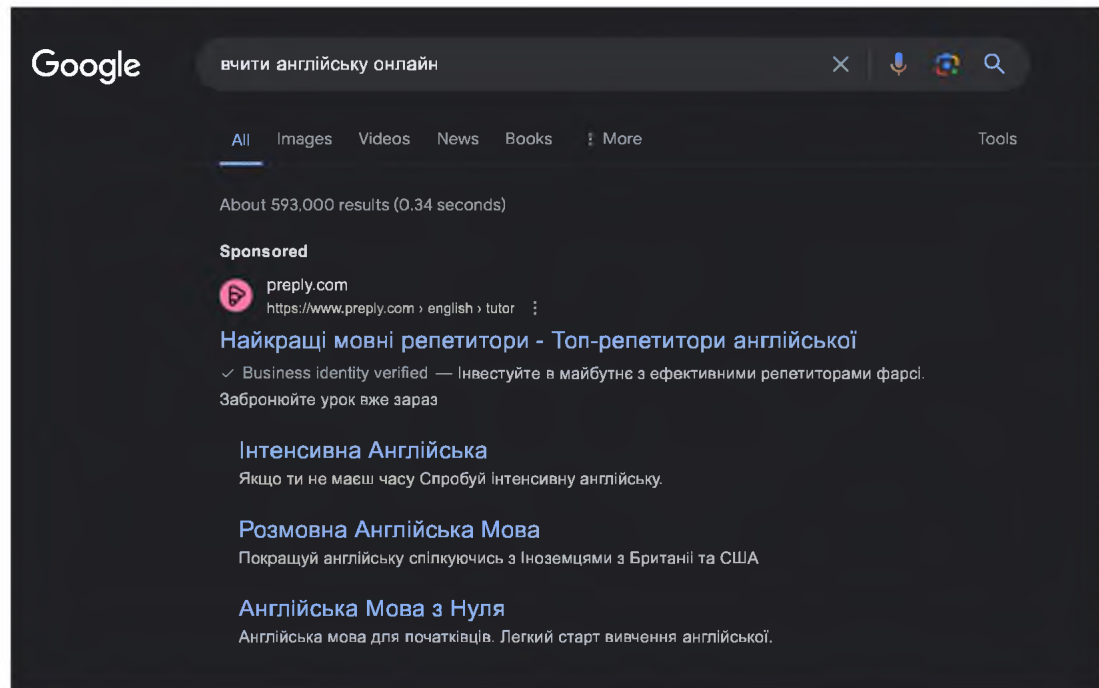


Рисунок 1.1 – Результат пошуку в Google за запитом
«вчити англійську онлайн»

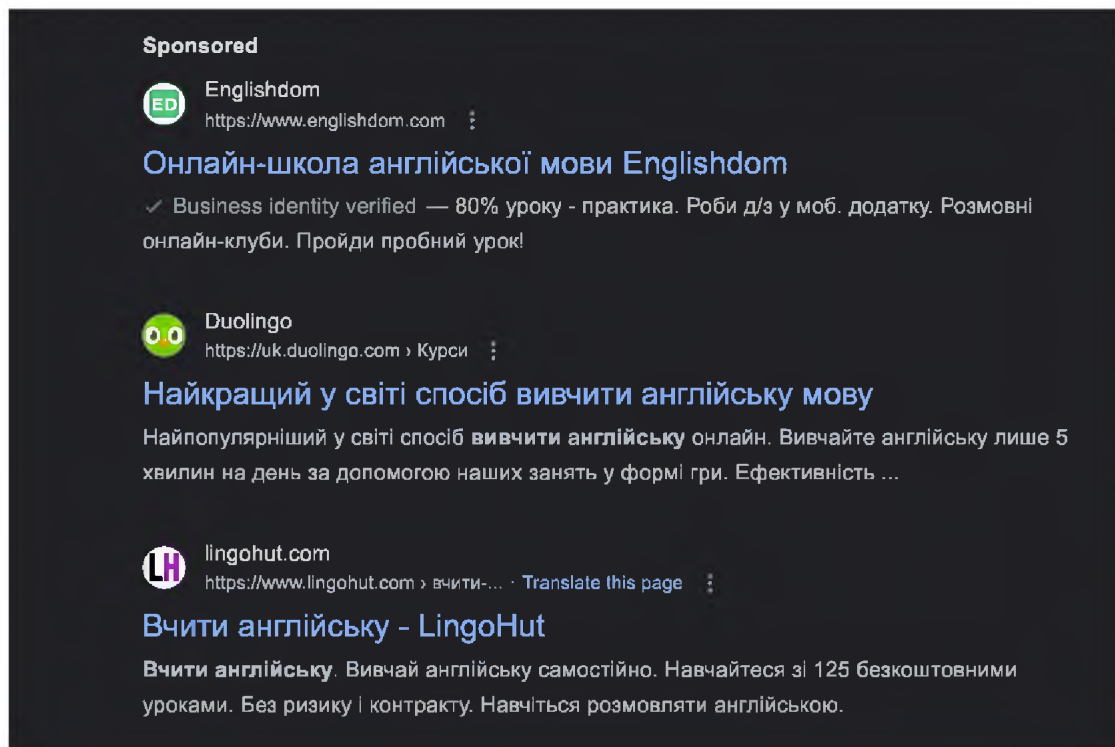


Рисунок 1.2 – Результат пошуку в Google. Продовження.

Першим буде проаналізовано «Preply». Preply працює як онлайн-магазин вивчення мов, що організовує зв'язки між репетиторами та учнями по всьому

світу. Завдяки їх платформі учні мають можливість досліджувати профілі викладачів, вибирати на основі своїх уподобань, відгуків і організувати персоналізовані уроки. Незважаючи на те, що Preply відрізняється від традиційних мовних шкіл, комісійний підхід Preply заохочує до високоякісного навчання. Це велика платформа, яка охоплюючи більш ніж 180 країн і надає учням доступ до широкого кола репетиторів, сприяючи гнучкості та персоналізації процесу вивчення мови. Однак, Preply не є прямим конкурентом, бо вони не являють собою мовну школу. Тим не менш, їх вебдодаток швидко завантажується, має гарний сучасний дизайн та створює приємний користувацький досвід. На рисунку 1.3 продемонстровано головну сторінку вебсайту.

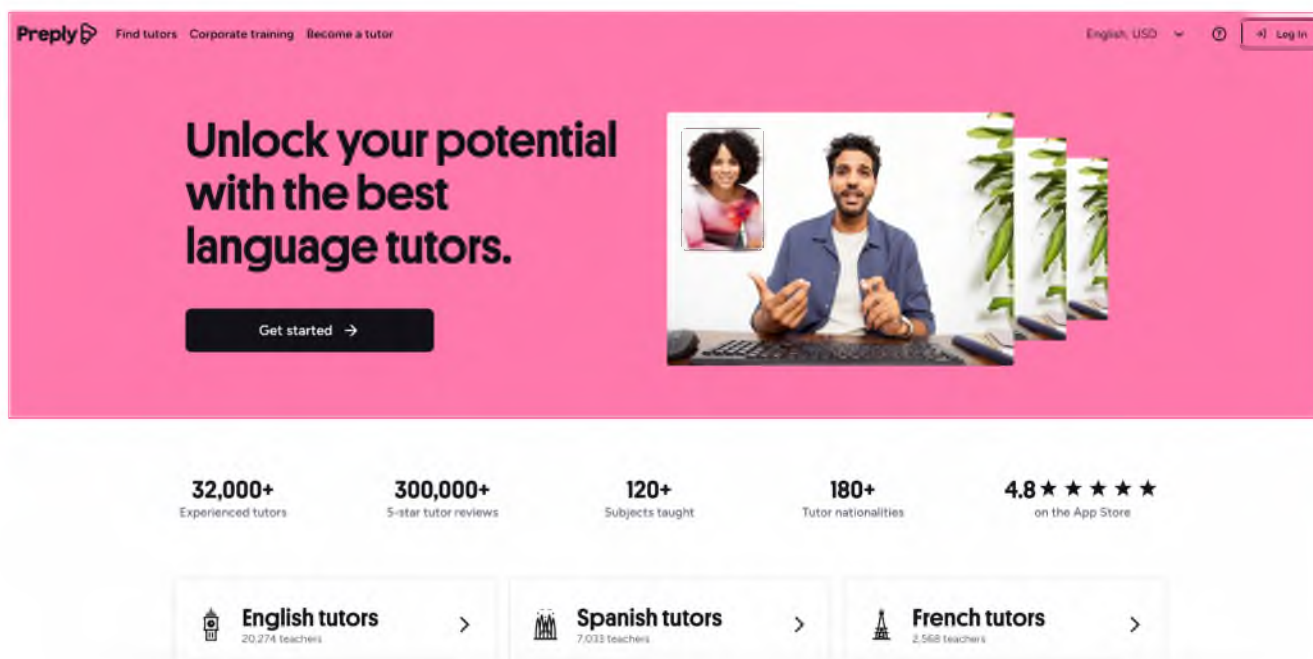


Рисунок 1.3 – Головна сторінка Preply

На головній сторінці можна побачити банер із закликом доєднатись, основні показники їх платформи та секцію з репетиторами розподіленими за мовними категоріями. На рисунку 1.4 зображено секцію з категоріями репетиторів.

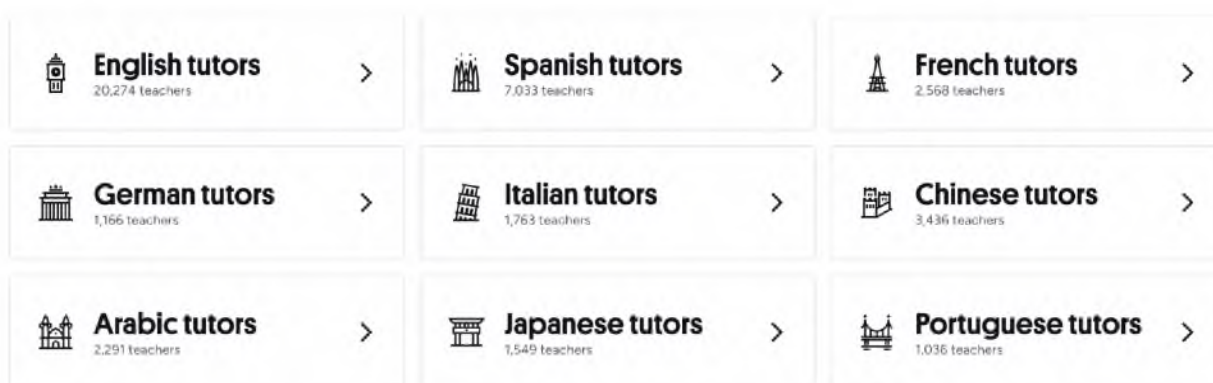


Рисунок 1.4 – Секція з категоріями репетиторів на сайті Preply

Наступним буде розглянуто “Englishdom”. Це велика онлайн-школа англійської мови, одна із найпопулярніших в Україні. Цю компанію можна вважати прямим конкурентом, бо вона є мовною школою. Вони пропонують індивідуальні та групові заняття з вчителем, проходження завдань у додатку та на сайті. Також пропонується пройти безкоштовне онлайн тестування з англійської мови для визначення рівня володіння мовою, звісно за допомогою цього тестування можливо оцінити тільки письмову частину, але це вже дає якесь розуміння для учня і певні ввідні дані. Цю ідеї пропонувати безкоштовне онлайн тестування можна запозичити і запланувати її втілення у своєму вебдодатку. На рисунку 1.5 зображено головну сторінку сайту Englishdom.

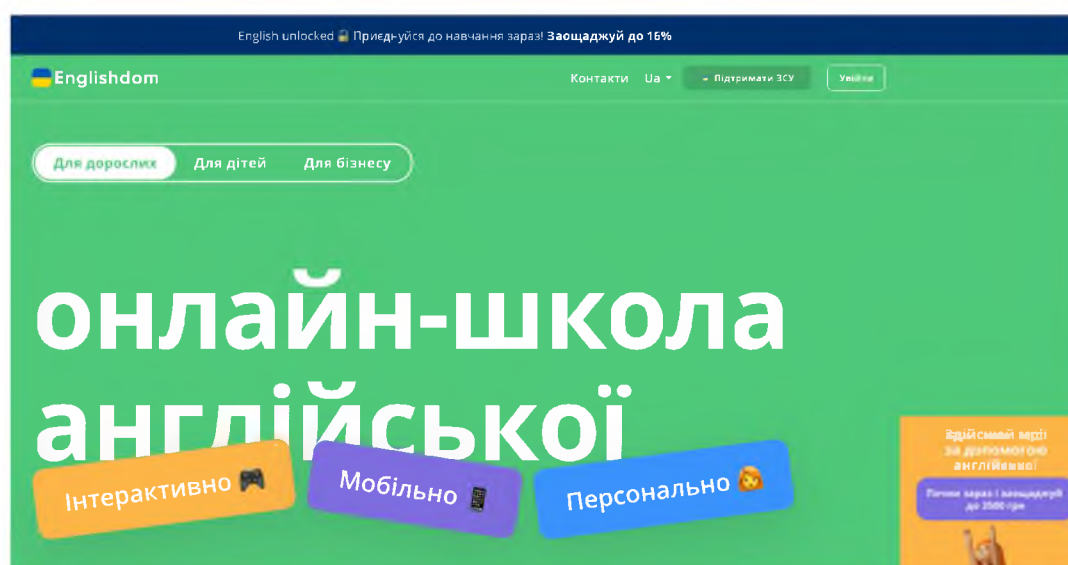


Рисунок 1.5 – Головна сторінку сайту Englishdom

Сторінка має приємний сучасний дизайн, одразу пропонується знижка 16%, звісно це скоріше маркетинговий хід, але може заохотити потенційних клієнтів скористатись саме їх послугами. Від моменту початку завантаження сторінки до того, як контент можна вважати завантаженим і користувач бачить сторінку проходить приблизно три секунди, що є непоганим результатом, але його можна покращувати.

Також нижче на головній сторінці користувач може побачити різні пакети послуг та їх ціни, що є зручним та наглядним досвідом для користувача. Можна відмітити, що вона пропонує доволі широкий вибір пакетів, що дозволяє залучити більшу кількість клієнтів. На рисунку 1.6 можна побачити секцію з цінами.

Вартість навчання

Що більше занять, то нижча ціна

- Урок триває 50 хвилин
- Ціна – від 11 usd/урок
- Є оплата частинами.

Окрім занять з викладачем, у вартість також входять

- Доступ до інтерактивної онлайн-платформи
- Розмовні клуби для практики говоріння
- Мобільні додатки, щоб займатися де завгодно
- Менеджер підтримки, який вирішує усі питання.

[Спробувати безкоштовно](#)

Пакет	Кількість занять	Локальний (USD)	Носій мови (USD)
Мінімальний	5	13	28
Економний	10	12	26
Популярний	20	12	25
Вигідний	50	11	23

Рисунок 1.6 – Секція з цінами на сайті Englishdom

Останнім буде розглянуто вебсторінку Duolingo. Duolingo — це американська освітня технологічна компанія, яка виробляє навчальні програми та

забезпечує мовну сертифікацію. Вони пропонують мобільній застосунок та вебплатформу, в яких користувачі можуть проходити уроки та вивчати різні мови. Duolingo не є мовною школою, а саме додатком для вивчення мов у інтерактивному форматі. Ця компанія не може вважатись прямим конкурентом, але технологічно пропонує якісні рішення, якими користуються мільйони користувачів. На рисунку 1.7 зображено головну сторінку вебплатформи, вона має сучасний мінімалістичний дизайн та високу швидкість завантаження.

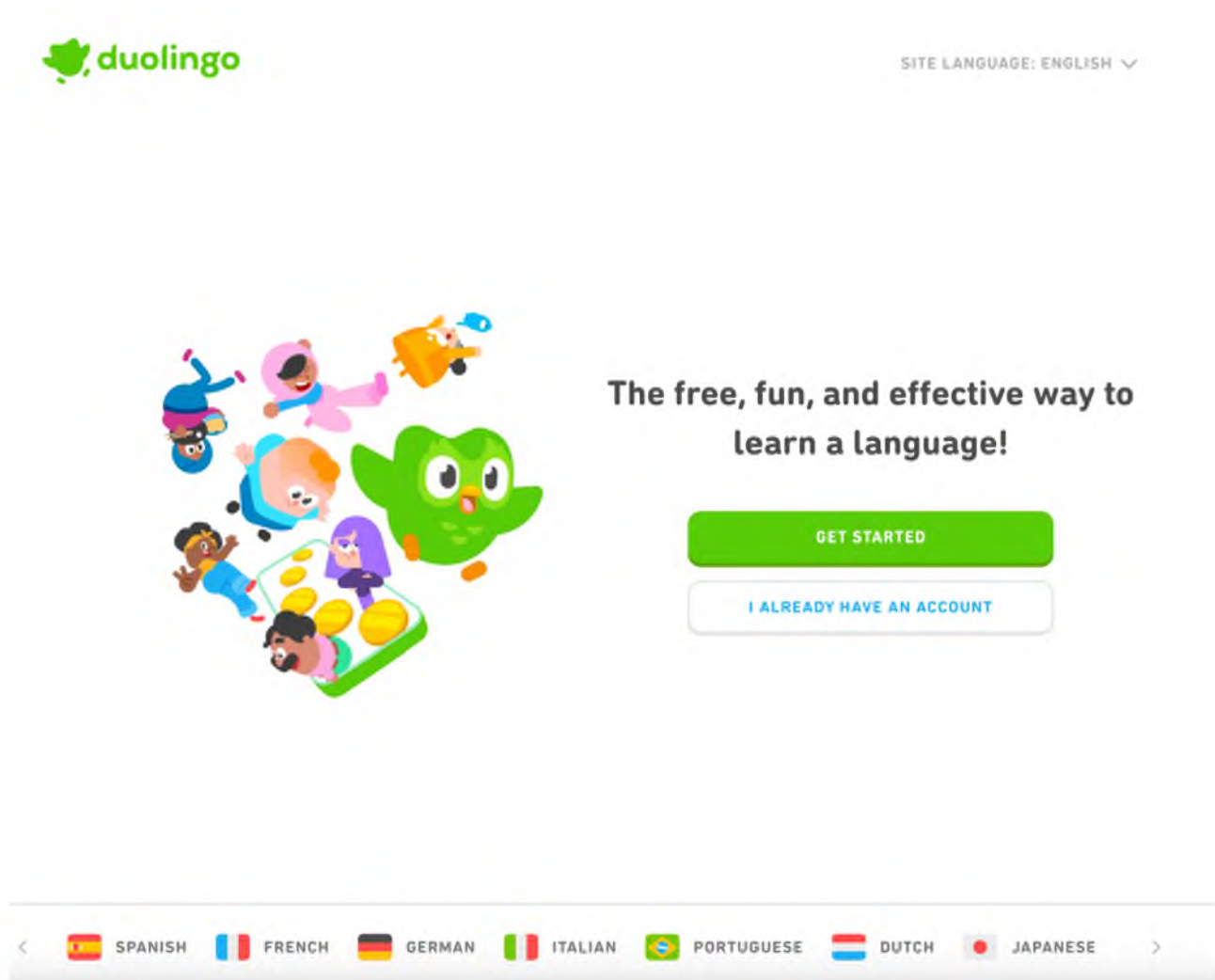


Рисунок 1.7 – Головна сторінка сайту Duolingo

На рисунку 1.8 зображено ігровий інтерфейс на вебсайті Duolingo для вивчення німецької мови.

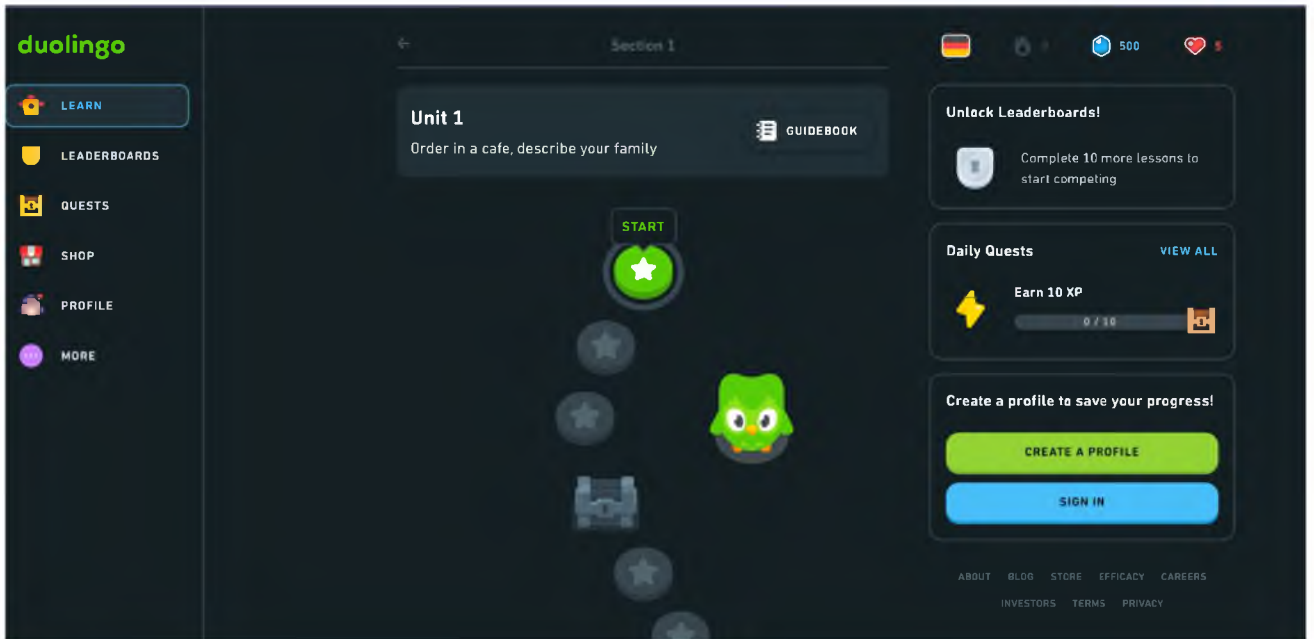


Рисунок 1.8 – Ігровий інтерфейс на вебсайті Duolingo

Після аналізу трьох успішних конкурентів у сфері платформ для вивчення іноземних мов стає очевидним, що диференціація та унікальність мають вирішальне значення для досягнення успіху. Усвідомлюючи це, наша мовна школа та навчальна платформа можуть використовувати це розуміння, віддаючи пріоритет створенню справді особливого досвіду та середовища для учнів.

Одним із підходів до досягнення цієї диференціації є створення школи з унікальною тематикою, такою як американське ранчо з декором у ковбойському стилі. Занурюючи учнів у це тематичне середовище, ми прагнемо запропонувати унікальний досвід навчання, який захоплює та мотивує учнів, імітуючи яскраве мовне середовище.

Цей інноваційний підхід не тільки відрізняє нас від конкурентів, але й розвиває відчуття автентичності та занурення, що покращує засвоєння мови. Поряд із наданням високоякісних навчальних матеріалів та персоналізованих шляхів навчання, ця особлива атмосфера забезпечить нам конкурентну перевагу на ринку, приваблюючи студентів, які прагнуть захоплюючого та незабутнього досвіду вивчення мови.

1.3 Визначення сфери застосування розроблюваної системи

Для того, щоб визначити сферу застосування системи аналізу даних для платформи вивчення іноземних мов спочатку необхідно описати яку структуру може мати компанія такого типу. На рисунку 1.9 можна побачити структуру підприємства, яке є мовною школою та в якому може бути використана система аналізу даних для платформи вивчення іноземних мов.

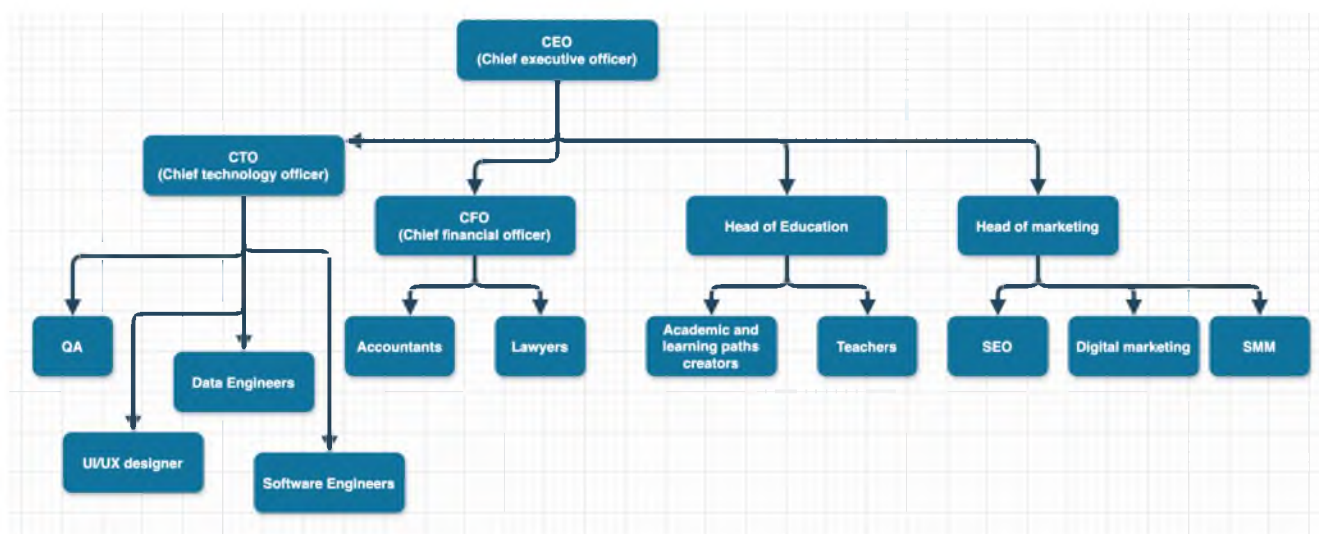


Рисунок 1.9 – Організаційна структура мовної школи

В організаційній структурі школи з вивчення іноземних мов кожна посада несе окремі обов'язки, які мають вирішальне значення для успіху школи [1].

СЕО (головний виконавчий директор) відіграє ключову роль у визначенні загального бачення та стратегічного напрямку школи. Він забезпечує керівництво виконавчою командою та керівниками відділів, забезпечуючи узгодження з організаційними цілями. Крім того, генеральний директор представляє школу перед зовнішніми зацікавленими сторонами, контролюючи ефективність і сприяючи підзвітності, укладає угоди та інші важливі документи для сприяння діяльності підприємства, виступає представником компанії, підтримує та розвиває корпоративну культуру.

СТО (Chief Technology Officer) очолює технологічний відділ, наглядає за розробкою програмного забезпечення, гарантією якості, розробкою користувацького інтерфейсу, користувацького досвіду та моделюванням даних. Він (вона) встановлює технічні стандарти, керує інженерами програмного забезпечення, дизайнерами, тестувальниками та дата інженерами, затверджує архітектуру системи та забезпечує своєчасне виконання програмних продуктів і послуг.

Фінансовий директор (Chief Financial Officer) відповідає за фінансові операції школи, включаючи складання бюджету, фінансове планування та звітність. Він (вона) керує бухгалтерською діяльністю, грошовими потоками та фінансовими ризиками, надаючи стратегічні фінансові поради для підтримки прийняття рішень.

Голова відділу зосереджується на розробці навчального плану, керівництві вчителями та відділом розробки методик викладання та матеріалів. Він (вона) відповідає за професійний зріст, високий рівень знання мов та постійний розвиток вчителів та методистів, впроваджує найкращі практики навчання.

Керівник відділу маркетингу очолює відділ маркетингу, контролюючи стратегії залучення студентів і просування школи. Вони керують SEO, цифровим маркетингом і діяльністю в соціальних мережах, розробляють рекламні кампанії, аналізують тенденції ринку та оцінюють ефективність маркетингу.

Підводячи підсумок, кожна посада в організаційній структурі школи з вивчення мов сприяє унікальному досвіду та лідерству для просування місії та цілей школи.

У мовній школі генеральний директор, фінансовий директор і відділ маркетингу є прямими зацікавленими сторонами системи аналізу даних. Вони значною мірою покладаються на інформацію, надану цією системою, щоб приймати обґрунтовані рішення на основі даних у відповідних сферах відповідальності.

Як вищий виконавчий директор, генеральний директор використовує аналіз даних, щоб отримати повне розуміння показників ефективності школи,

включаючи кількість студентів, отримання прибутку та ефективність роботи. Використовуючи ці знання, генеральний директор може приймати стратегічні рішення щодо розподілу ресурсів, можливостей розширення та довгострокових стратегій зростання. Виявляючи тенденції та використовуючи можливості, генеральний директор гарантує, що школа збереже свою конкурентоспроможність на ринку вивчення мов.

Тим часом фінансовий директор залежить від аналізу даних для оцінки фінансового стану школи, аналізу тенденцій надходжень, моделей витрат і фінансових ризиків. Ці знання дозволяють фінансовому директору оптимізувати розподіл бюджету, визначити можливості економії коштів і забезпечити фінансову відповідальність. Обґрунтовані рішення щодо інвестиційних пріоритетів, фінансового планування та стратегій управління ризиками стають можливими завдяки аналізу даних, що підтримує фінансову стабільність і цілі зростання школи.

Подібним чином відділ маркетингу покладається на аналіз даних для вимірювання ефективності маркетингових кампаній, відстеження витрат на залучення клієнтів і оцінки рентабельності інвестицій. Ці знання дають змогу приймати рішення щодо цільових демографічних груп, рекламних каналів і стратегій обміну повідомленнями, щоб зацікавити майбутніх студентів. Удосконалюючи маркетингові стратегії та ефективно розподіляючи ресурси, відділ маркетингу забезпечує безперервне вдосконалення потенційних клієнтів і коефіцієнтів конверсії.

По суті, генеральний директор, фінансовий директор і відділ маркетингу відіграють невід'ємну роль як зацікавлені сторони системи аналізу даних у школі вивчення мов. Використовуючи знання, отримані від цієї системи, вони приймають обґрунтовані рішення, які впливають на продуктивність організації, фінансову стабільність і ефективність маркетингу, що зрештою сприяє успіху школи в конкурентному середовищі.

1.1 Постановка задачі

Постановку задачі кваліфікаційної роботи можна сформулювати наступним чином: розробити систему для аналізу даних платформи вивчення іноземних мов. Це включає в себе розробку MVP вебдодатку платформи вивчення іноземних мов та окремо створення системи для аналізу даних.

Розробка інформаційної системи включає кілька етапів, спрямованих на полегшення технічних операцій, таких як розробка системи та мінімізація можливих помилок. Ці етапи охоплюють:

- аналіз предметної області, ситуації на ринку та конкурентів;
- формування вимог до інформаційної системи;
- визначення основних функціональних та нефункціональних вимог до вебдодатку та системи аналізу даних;
- визначення функцій інтерфейсу клієнтської частини вебсайту;
- створення дизайну вебдодатку;
- прогнозування сценаріїв взаємодії користувачів із вебдодатком;
- категоризація користувачів і визначення їхніх дозволів і привілеїв у системі;
- проектування архітектури системи аналізу даних та вебдодатку;
- дослідження та вибір найбільш прийнятних технологій для розробки інформаційної системи;
- визначення сутностей і зв'язків у системі та проектування операційної бази даних;
- визначення подій користувача для відправки в систему аналізу даних;
- програмна реалізація вебдодатку;
- розгортання вебдодатку;
- побудова та розгортання системи аналізу даних;
- покриття компонентів системи тестами.

Виконання всіх цих етапів є необхідним для успішного розвитку інформаційної системи.

2 ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Визначення основних функціональних та нефункціональних вимог до вебдодатку та системи аналізу даних

Основні функції вебдодатку і системи аналізу даних зосереджені навколо створення MVP (Minimum Viable Product), який служить привабливою платформою для залучення нових клієнтів до мовної школи. Версія MVP вебдодатку розроблена для ефективною демонстрації особливостей школи, її цінностей та її унікальної теми американського ранчо та ковбоїв.

Відвідувачі вебсайту побачать візуально привабливий інтерфейс, який занурює їх в атмосферу американського ранчо з відповідними елементами дизайну. Ця тематична презентація має на меті захопити відвідувачів і виділити школу серед конкурентів, пропонуючи незабутні та захоплюючі враження.

Крім того, вебдодаток надаватиме вичерпну інформацію про пропозиції школи, включаючи подробиці про мовні курси, методології навчання та досвід вчителів. Цей вміст ретельно підбрано, щоб передати ціннісну пропозицію школи та підкреслити її прагнення до якісної освіти.

Окрім інформації про школу, вебдодаток запропонує доступ до корисних матеріалів, призначених для підтримки вивчення мови. Ці матеріали можуть містити зразки уроків, мовні тести та ресурси для завантаження, що дає відвідувачам змогу ознайомитися з навчальним вмістом, доступним у школі.

Основна мета вебдодатку – заохотити відвідувачів записатися на пробний урок. Щоб полегшити це, у вебсайт інтегровано форму для запису, яка дозволить відвідувачам легко надіслати свою контактну інформацію та подати запит на пробний урок. Цей спрощений процес призначений для максимізації конверсії відвідувачів у студентів та заохочення взаємодії з пропозиціями школи.

Загалом MVP версія вебдодатку служить потужним інструментом для залучення нових клієнтів до мовної школи, ефективно повідомляючи про її унікальні особливості, цінності та тематичну спрямованість, а також надаючи

відвідувачам цінні ресурси та простий шлях до бронювання пробного уроку.

Таким чином MVP версія вебдодатку повинна мати наступні бізнес-функції:

- відображення банеру та заклику до запису на пробне заняття;
- відображення відео з презентацією школи;
- відображення інформації про основні переваги школи;
- відображення форми для запису на пробне заняття;
- повідомлення адміністраторам школи та вчителям про заявку на пробне заняття;

У своїй MVP версії система аналізу даних структурована так, щоб збирати та використовувати дані про події користувача, забезпечуючи їх доступність у сховищі даних для подальшого аналізу та генерації ідей. Ці дані про події користувача охоплюють різні взаємодії та поведінку користувачів на платформі вивчення мови.

Щоб зробити інсайти на основі даних більш наглядними, версія MVP інтегрує інструменти бізнес-аналітики зі сховищем даних. Ці інструменти дозволяють діставати, перетворювати та аналізувати дані, надаючи зацікавленим сторонам корисну інформацію про поведінку користувачів і продуктивність платформи у легкому для сприйняття візуальному форматі. Тобто система аналізу даних включає можливості візуалізації даних, що дозволяє створювати візуально переконливі діаграми та графіки на основі відстежуваних даних. Ці візуалізації служать потужними інструментами для передачі складної інформації в легко зрозумілому форматі, що дозволяє зацікавленим сторонам з першого погляду отримати цінну інформацію.

Зберігаючи дані про події користувача у сховищі даних і використовуючи інструменти бізнес-аналітики для аналізу та візуалізації, система аналізу даних дає змогу зацікавленим сторонам приймати керовані даними рішення, оптимізувати продуктивність платформи та покращити загальний досвід вивчення мови для користувачів.

Таким чином MVP версія системи аналізу даних повинна мати наступні бізнес-функції:

- зберігати дані про події користувача у сховищі даних;
- пропонувати візуалізацію ключових метрик на основі зібраних даних.

2.2 Визначення функцій інтерфейсу клієнтської частини вебсайту

Інтерфейс клієнтської частини для платформи вивчення іноземних мов служить основною точкою взаємодії для користувачів, які відвідують вебсайт. Ось ключові функції, які має містити клієнтський інтерфейс:

- презентація школи. Всі частини інтерфейсу мають демонструвати унікальні особливості, місію та цінності мовної школи, підкреслювати тематику або атмосферу школи, як-от американське ранчо та декор у ковбойському стилі, щоб створити для відвідувачів особливий досвід занурення;

- корисні матеріали. Має бути продемонстровано та надано доступ до підібраних освітніх ресурсів і підготовлених матеріалів, таких як граматичні посібники, списки лексики та культурні відомості; Відвідувачі мають мати можливість переглянути ці матеріали безкоштовно, тим самим підвищиться довіра до школи;

- форма заявки на пробний урок. Користувачі мають мати змогу легко заповнити та надіслати форму для запиту на пробний урок, вказавши своє ім'я, контактну інформацію, рівень володіння мовою та бажаний формат уроку (індивідуальний або груповий). Форма має бути інтуїтивно зрозумілою для користувача;

- відео презентація. Інтерфейс має містити рекламне відео, яке демонструє цікавий огляд мовної школи, показуючи всі ключові моменти, методи навчання, вчителів та унікальну атмосферу. Відео має висвітлювати переваги школи, наприклад її тематичне середовище, натхненне американським ранчо та ковбойською культурою, а також переваги навчання в такому захоплюючому середовищі. Використовуючи візуальні матеріали, розповіді і відгуки, за допомогою відео можливо ефективно передати ціннісну пропозицію школи та створити переконливу розповідь, яка резонуватиме з майбутніми учнями. Це відео

має бути розміщено на помітному місці на домашній сторінці вебсайту, щоб зробити його легко доступним для відвідувачів і зробити внесок у загальну презентацію школи;

- адаптивний дизайн. Клієнтський інтерфейс має бути оптимізованим для різних пристроїв і розмірів екранів, забезпечуючи безперебійну роботу на настільних ПК, планшетах і смартфонах. Для цього використовуються принципи адаптивного дизайну, щоб динамічно адаптувати макет і вміст відповідно до пристрою користувача та розміру вікна перегляду;

- навігація та доступність. Клієнтський інтерфейс має бути інтуїтивно зрозумілим, мати навігаційне меню та чітку структуру сайту, щоб допомогти користувачам легко знаходити потрібну інформацію. Інтерфейс повинен бути доступним для користувачів з обмеженими можливостями, включно з програмами зчитування з екрана, навігацією з клавіатури та контрастом кольорів;

- візуальний дизайн і брендинг. У всьому інтерфейсі має зберігатись узгодженість із брендингом та візуальною ідентичністю мовної школи, включаючи кольори, типографіку та зображення. Візуально привабливі елементи дизайну та графіка можуть бути використані для покращення загальної естетики та взаємодії з користувачем.

2.3 Створення дизайну вебдодатку

Розробляючи дизайн вебдодатку, було дотримано сучасного підходу, відомого як «mobile first», віддаючи пріоритет мобільному дизайну, щоб забезпечити оперативність і зручність використання на всіх пристроях. Цей метод є особливо актуальним, бо починаючи з 2016 року кількість трафіку з мобільних пристроїв перевищила кількість трафіку з комп'ютерів або ноутбуків [2]. Останніми роками мобільний трафік приблизно займав 57% усього вебтрафіку, при цьому трафік комп'ютерів становить приблизно 39%, а планшети займають решту відсотків [3].

Враховуючи ці статистичні дані, вкрай важливо віддавати перевагу мобільному дизайну, щоб ефективно задовольнити більшість користувачів. Використання «mobile first» підходу у вебдизайні має вирішальне значення для створення адаптивних, зручних вебсайтів, які відповідають потребам і вподобанням сучасної аудиторії, орієнтованої на мобільні пристрої. Пріоритет мобільного дизайну гарантує оптимізацію вебсайту для різних розмірів екрана та пристроїв, пропонуючи послідовну та зручну роботу на настільних комп'ютерах, планшетах і смартфонах. Принципи адаптивного дизайну дозволяють вмісту динамічно адаптуватися до різних розмірів вікна перегляду, забезпечуючи читабельність і зручність використання на невеликих екранах.

Зосередження на мобільному дизайні як на першому пріоритеті дозволяє дизайнерам оптимізувати користувацький інтерфейс, визначити пріоритетність важливого вмісту та оптимізувати навігацію. Зручні для мобільних пристроїв макети покращують взаємодію з користувачем, мінімізують час завантаження та роблять навігацію простою та зрозумілою, що призводить до більшого залучення та задоволення серед мобільних користувачів.

Крім того, дизайн, орієнтований на мобільні пристрої, сприяє доступності, віддаючи перевагу простоті та ясності в елементах дизайну, полегшуючи навігацію та взаємодію з вебсайтом для користувачів з обмеженими можливостями чи вадами. Розробка з урахуванням доступності гарантує, що всі користувачі, незалежно від свого пристрою чи здібностей, зможуть отримати доступ до вмісту сайту та ефективно працювати з ним.

Додатково підхід, орієнтований на мобільні пристрої, заохочує дизайнерів надавати пріоритет оптимізації продуктивності вебдодатку, оптимізацію зображень і мінімізацію ресурсомістких елементів. Зосереджуючись на швидкій і легкій мобільній роботі, вебсайти можуть покращити час завантаження і підвищити загальну продуктивність системи на всіх пристроях.

Оскільки використання кількість мобільних пристроїв перевищує кількість настільних комп'ютерів та ноутбуків, і продовжує зростати, то застосування підходу, орієнтованого на мобільні пристрої, гарантує, що вебсайт мовної школи

залишиться актуальним та буде показувати гарні показники в середовищі, що все більше орієнтується на мобільні пристрої. Розробка в першу чергу для мобільних пристроїв дозволяє вебсайтам адаптуватися до змін поведінки користувачів і технологічних досягнень, випереджаючи очікування та вподобання користувачів.

Дизайн вебсайту було розроблено за допомогою Figma, універсального інструменту дизайну, відомого своїм повним набором функцій, призначених для створення макетів. Figma пропонує низку корисних функцій, які спрощують процес проектування, забезпечуючи ефективну колаборацію та створення макетів.

Figma забезпечує можливість ефективної співпраці в команді в режимі реального часу, дозволяючи кільком членам команди працювати одночасно над одним дизайнерським проектом [4].

За допомогою Figma дизайнери можуть легко створювати інтерактивні прототипи для імітації взаємодії користувачів і перевірки зручності використання вебсайту. Це допомагає визначити потенційні вдосконалення дизайну та протестувати макет перед впровадженням.

Figma підтримує принципи адаптивного дизайну, що полегшує розробку макетів, які адаптуються до різних розмірів екрана та пристроїв. Це забезпечує послідовну та оптимізовану взаємодію з ноутбуками, смартфонами та планшетами.

Figma пропонує широкий спектр плагінів та інтеграцій з іншими інструментами та службами, розширюючи свої функціональні можливості та підвищуючи продуктивність для дизайнерів.

Враховуючи всі переваги було обрано Figma для створення візуально привабливого і зручного макету вебсайту, який віддає перевагу адаптивному дизайну та “mobile first” підходу, водночас пропонуючи безперебійну роботу на всіх інших пристроях.

На рисунку 2.1 зображено перший екран мобільної версії вебдодатку мовної школи.



Рисунок 2.1 – Перший екран мобільної версії вебдодатку

На першому екрані мобільної версії вебдодатку демонструється логотип, банер, кнопка із закликом записатись на пробне заняття та мобільна версія меню у згорнутому вигляді. На рисунку 2.2 зображено відкрите мобільне меню вебдодатку.



Рисунок 2.2 – Відкрите мобільне меню веб додатку

Мобільне меню має посилання на всі розділи вебсторінки. На рисунку 2.3 показано другу секцію мобільної версії вебдодатку.

На рисунку 2.4 показано третю секцію мобільної версії вебдодатку. Це секція із корисними матеріалами, яка в подальшому буде вести на окремий розділ із безкоштовними матеріалами.



Рисунок 2.4 – Секція з корисними матеріалами мобільної версії вебдодатку

Також в секції з корисними матеріалами показуються картки із найпопулярнішими темами, які будуть вести на окремі сторінки із відповідними матеріалами.

На рисунку 2.5 зображено мобільну версію форми для запису на пробне заняття.



The image shows a mobile application interface for a free trial class sign-up. The background is a dark grey color. At the top, the text "GET FREE A TRIAL CLASS" is displayed in a light, serif font. Below this, the instruction "Fill out this form to get your free trial class" is centered. The form consists of several input fields, each with a light grey border and rounded corners. The fields are labeled: "First Name", "Last Name", "Email", "Phone Number", and "A messenger where we can contact you". Below these fields is a larger text area with the prompt "Describe your experience and the main purpose of learning English". At the bottom of the form is a prominent "SEND" button. A small, dark, leafy plant is visible in the bottom right corner of the form's background.

Рисунок 2.5 – Мобільна версія форми для запису на пробне заняття

Форма для запису на пробне заняття має наступні поля:

- ім'я користувача;
- прізвище користувача;
- email адреса;

- телефон;
- бажаний месенджер в якому користувач очікує отримати повідомлення;
- поле для опису мети вивчення мови, досвіду та додаткової інформації про студента.

На рисунку 2.6 зображено футер мобільної версії вебдодатку, який включає в себе логотип, посилання на соціальні мережі та меню навігації.

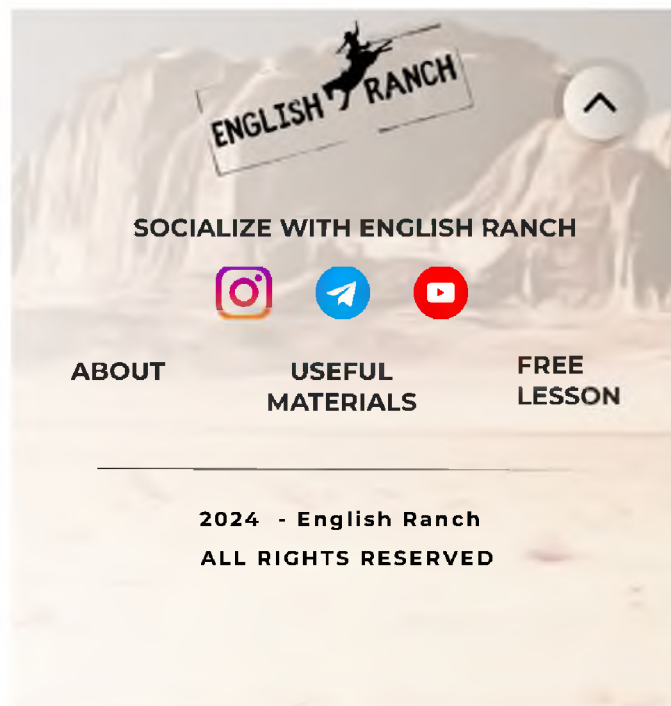


Рисунок 2.6 – Футер мобільної версії вебдодатку

Наступним кроком було створення комп'ютерної версії вебдодатку. На рисунку 2.7 зображено перший екран комп'ютерної версії вебдодатку мовної школи.

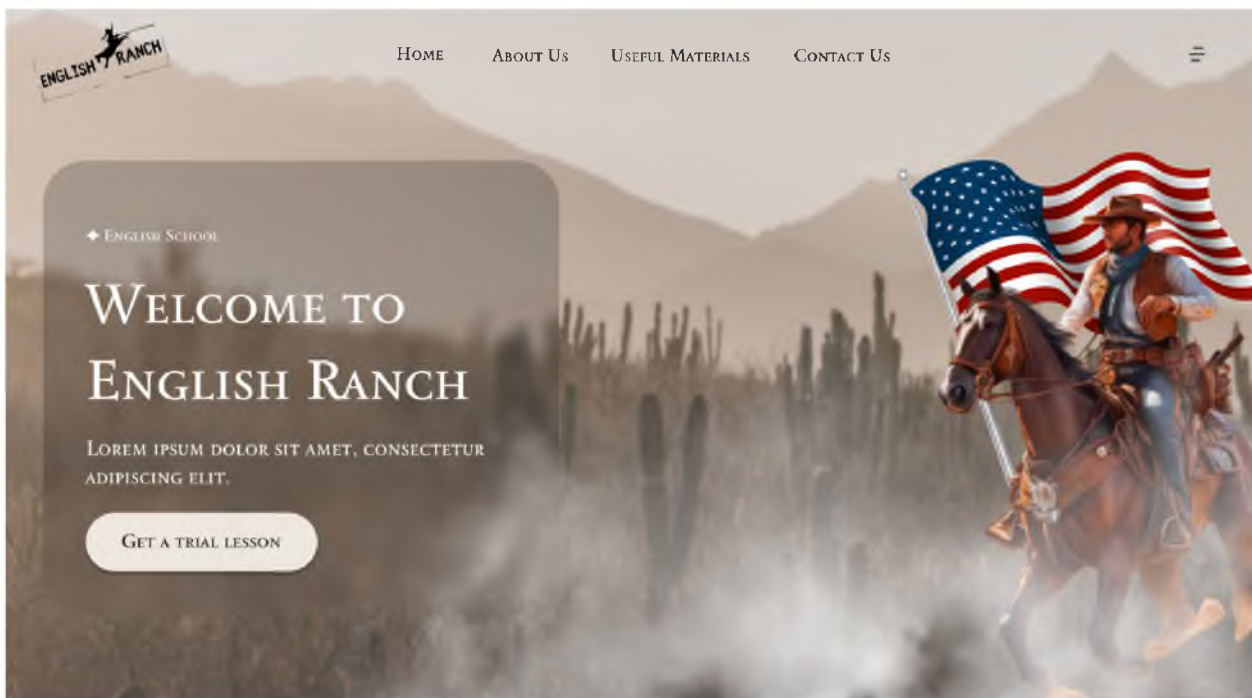


Рисунок 2.7 – Перший екран комп’ютерної версії вебдодатку

На першому екрані комп’ютерної версії вебдодатку демонструється логотип, меню, банер, кнопка із закликом записатись на пробне заняття та привітальна фраза.

На рисунку 2.8 показано другу секцію комп’ютерної версії вебдодатку.

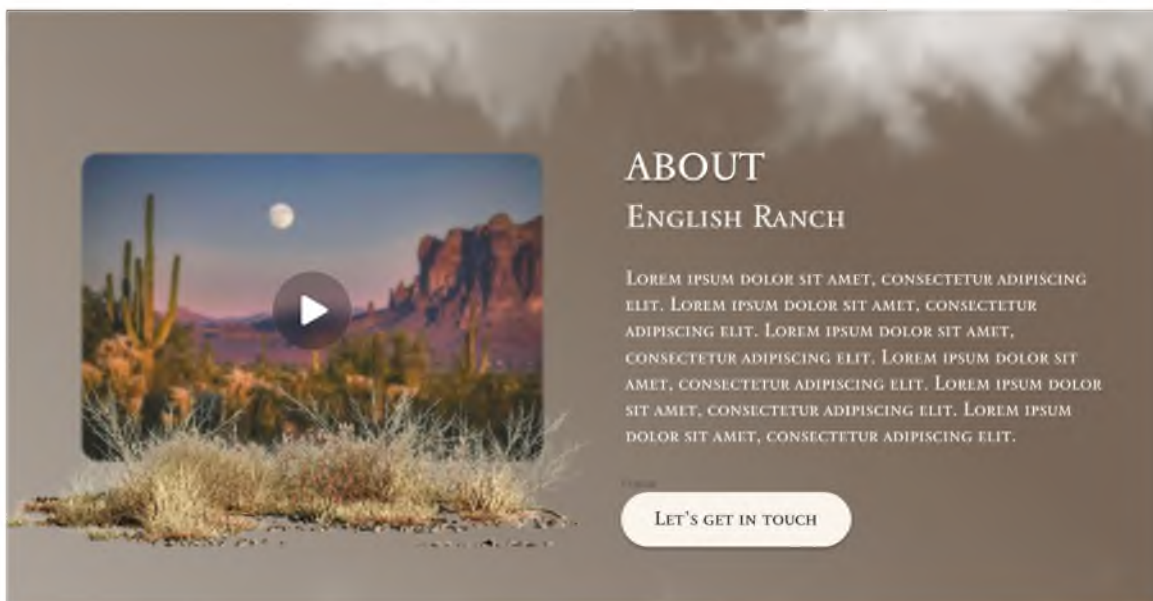


Рисунок 2.8 – Друга секція комп’ютерної версії вебдодатку

Друга секція комп'ютерної версії вебдодатку демонструє рекламне відео з основною інформацією про школу, текстовий опис і заклик до запису на пробний урок.

На рисунку 2.9 показано третю секцію комп'ютерної версії вебдодатку, це секція із корисними матеріалами. Ця секція має ті самі компоненти, що і мобільна версія додатку, тільки має візуально оптимізований макет для привабливого вигляду на великих екранах.

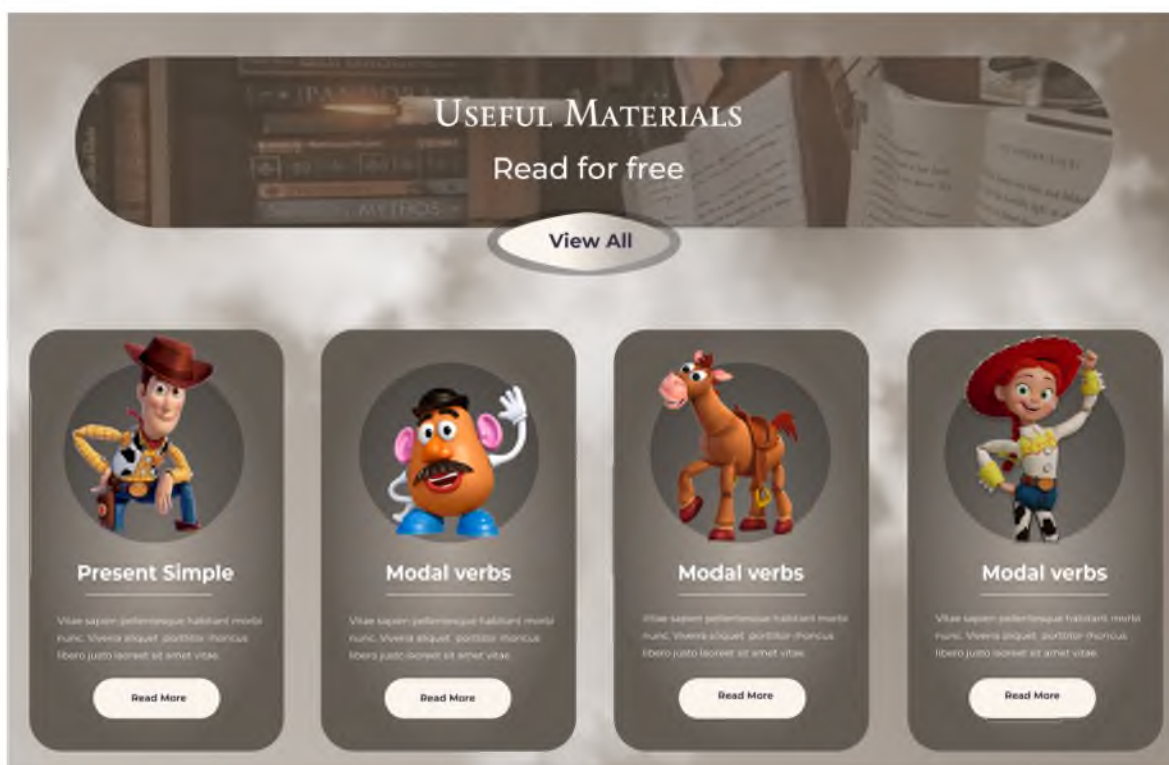
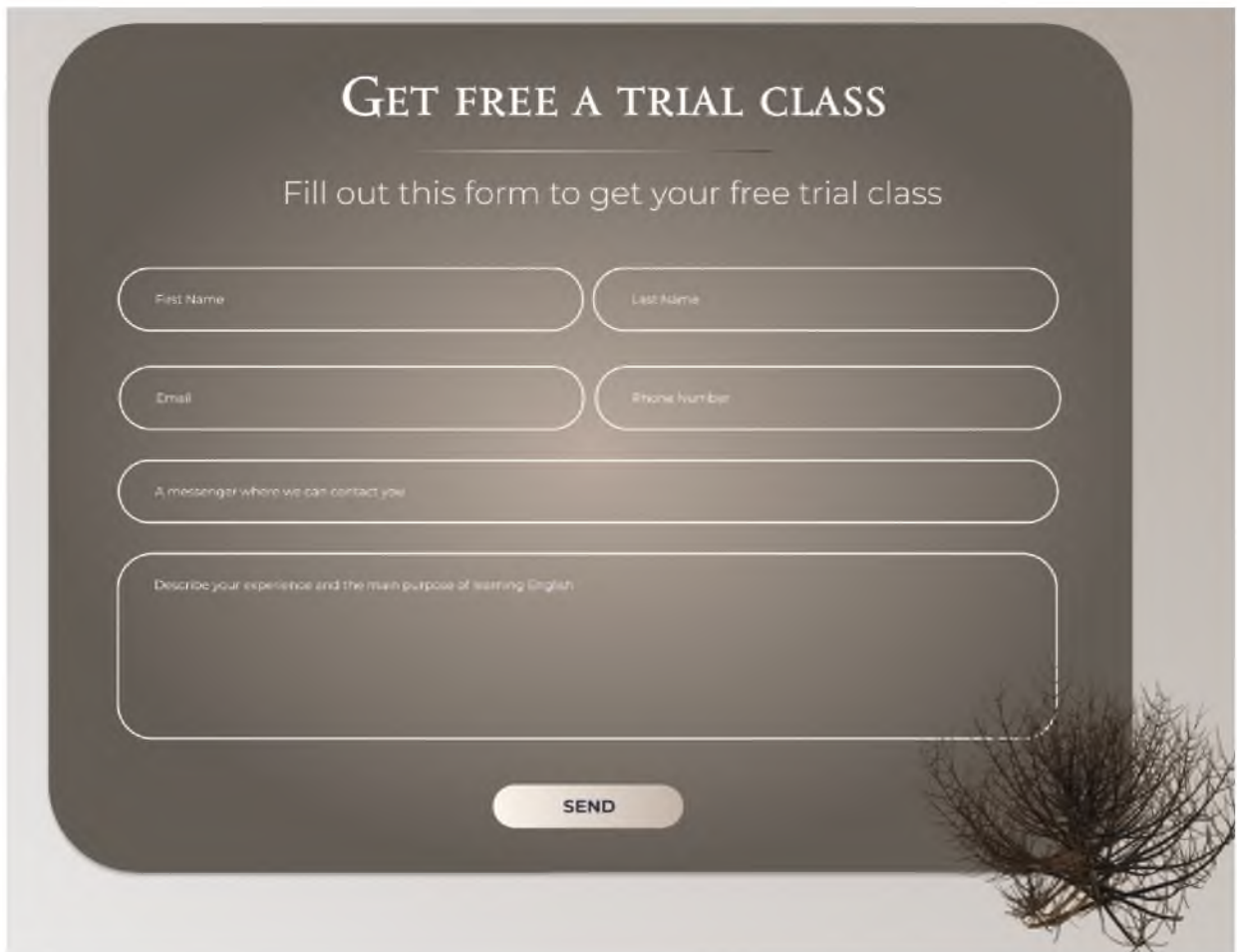


Рисунок 2.9 – Секція з корисними матеріалами комп'ютерної версії вебдодатку

На рисунку 2.10 зображено комп'ютерну версію форми для запису на пробне заняття. Форма має такі самі поля, як і в мобільній версії, змінено тільки візуальний вигляд форми.



GET FREE A TRIAL CLASS

Fill out this form to get your free trial class

First Name

Last Name

Email

Phone Number

A messenger where we can contact you

Describe your experience and the main purpose of learning English

SEND

Рисунок 2.10 – Комп’ютерна версія форми для запису на пробне заняття

На рисунку 2.11 зображено футер комп’ютерної версії вебдодатку, який включає в себе ті самі компоненти, що і мобільна версія.

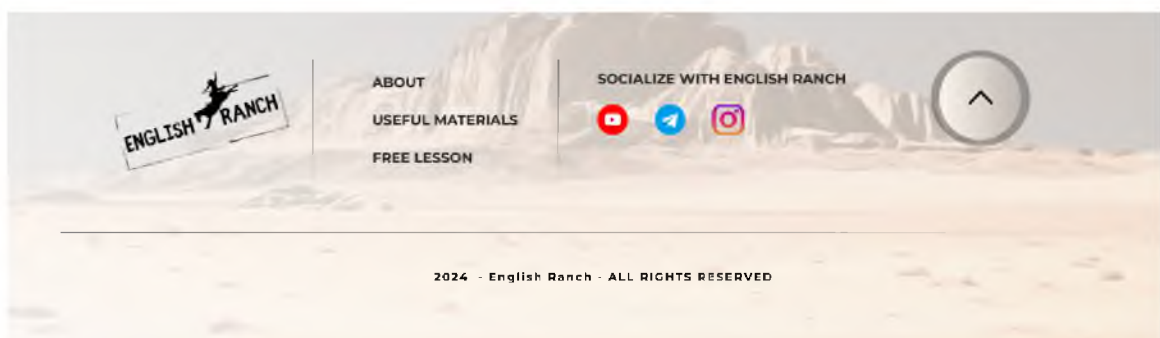


Рисунок 2.11 – Футер комп’ютерної версії вебдодатку

2.4 Прогнозування сценаріїв взаємодії користувачів із вебдодатком

У MVP версії вебдодатку взаємодія користувача із вебдодатком є обмеженою мінімально закладеним функціоналом. Цю взаємодію можна передбачити наступним чином.

Відкривши вебсайт, користувачі переходять на головну сторінку, де зустрічають короткий опис мовної школи. Тут вони дізнаються про пропозиції, цінності та унікальні особливості школи. Користувачі можуть взаємодіяти з вмістом, прокручуючи текст і натискаючи гіперпосилання для отримання більш детальної інформації.

Для тих, хто бажає заглибитися, доступне вступне відео, яке надає візуальний огляд атмосфери школи та пропозицій. Користувачі, які вибрали перегляд відео, взаємодіють із елементами керування відеопрогравача, такими як відтворення, пауза, гучність і повноекранний режим. Вони також можуть перемотувати назад або пропускати частини відео відповідно до своїх уподобань.

Переглянувши опис школи та, можливо, переглянувши вступне відео, користувачі можуть вирішити вжити заходів, відповівши на заклик до дії. Зазвичай це включає запит на пробний урок, натиснувши відповідну кнопку чи посилання. Потім користувачі перенаправляються до форми, де вони можуть ввести свою контактну інформацію та вказати свою зацікавленість у пробному уроці. Тут вони взаємодіють із полями форми, вводячи такі дані, як своє ім'я, адреса електронної пошти та інше.

По суті, ці передбачені сценарії окреслюють основні шляхи взаємодії користувачів із MVP версією вебдодатку. Незважаючи на обмежений обсяг, ці взаємодії призначені для того, щоб направляти користувачів через важливі дії, такі як перегляд вмісту, перегляд вступного матеріалу та ініціювання запиту на пробний урок. У міру розвитку вебплатформи можуть з'являтися додаткові функції для збагачення користувацького досвіду.

На рисунку 2.12 зображено use case діаграму MVP версії вебдодатку мовної школи.

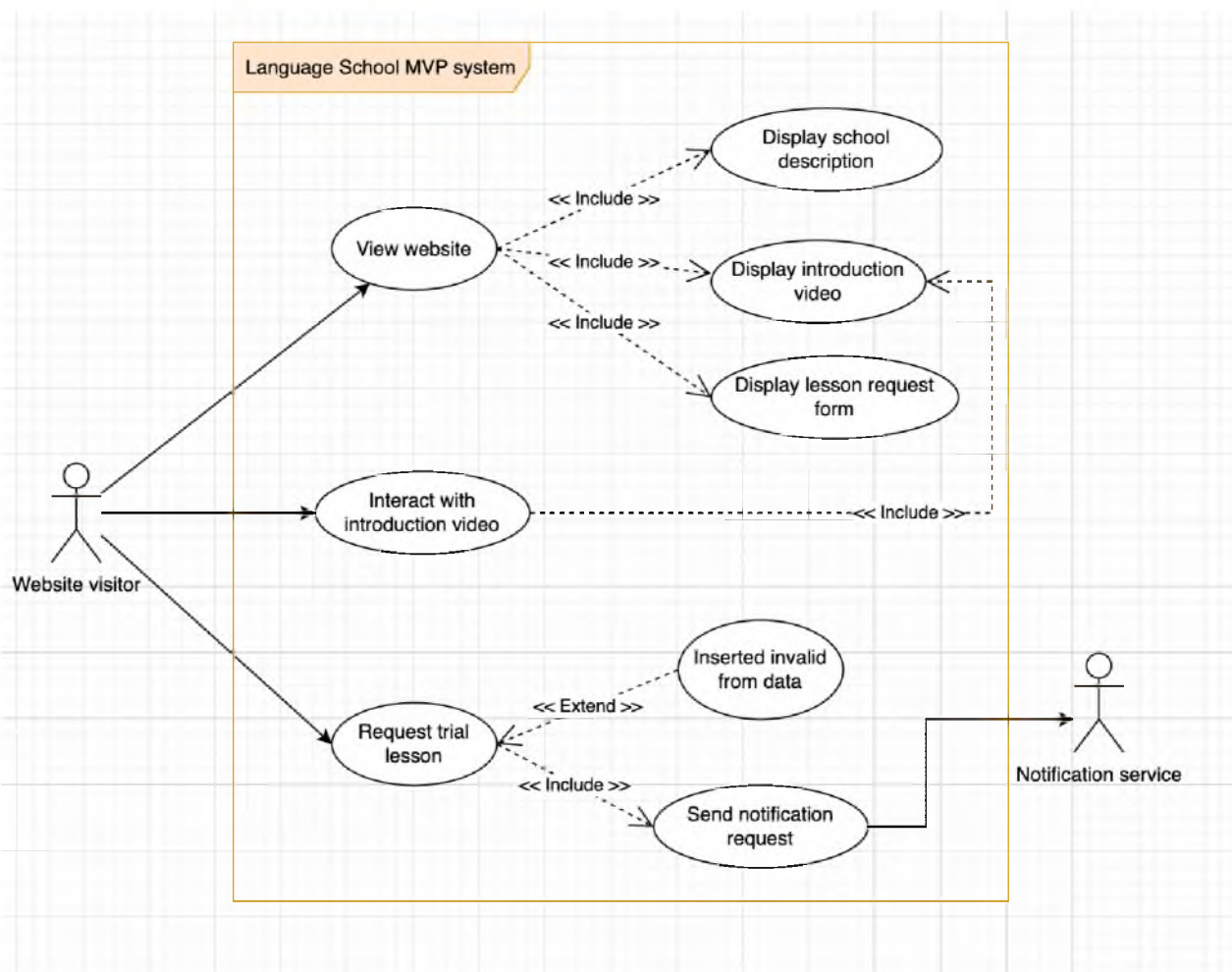


Рисунок 2.12 – Use case діаграма MVP версії вебдодатку мовної школи

На цій діаграмі зображено двох акторів: відвідувача вебсторінки та окремий сервіс для оповіщення [5]. Відвідувач вебсторінки може переглянути вебсторінку, що включає в себе відображення опису школи, відображення відео та форми для запису на пробне заняття. Також відвідувач може взаємодіяти з відео елементом, що включає в себе відображення цього відео елементу. Відвідувач вебсторінки може записатись на пробне заняття заповнивши форму, під час заповнення форми є вірогідність, що користувач введе неправильні дані, після чого йому доведеться виправити помилки та відправити форму знову. Успішне заповнення форми та запит пробного уроку стане тригером для відправки повідомлення в окремий сервіс доставки повідомлень для інформування працівників школи про новий запит на пробне заняття.

2.5 Категоризація користувачів і визначення їхніх дозволів і привілеїв у системі

У MVP версії системи існує лише одна категорія користувачів: гість (неавторизовані користувачі). Ці користувачі мають обмежені дозволи, зокрема вони можуть:

- перейти на головну сторінку вебсайту, щоб дізнатися про мовну школу та її пропозиції;
- відправити запит на пробний урок;
- зв'язатися зі школою, надіславши запитання через контактну форму, щоб дізнатися про конкретні деталі чи проблеми.

З впровадженням нових функцій, таких як онлайн-освітня платформа, гості отримають додаткові функції. Зокрема, вони матимуть можливість зареєструватися на сайті. Після реєстрації всім користувачам за замовчуванням буде призначена роль студента. Це означає, що зареєстровані користувачі матимуть можливість:

- зареєструвати обліковий запис на вебсайті, вказавши свої дані для створення профілю;
- отримати доступ до навчальних матеріалів, які пропонуються тільки зареєстрованим користувачам;
- взяти участь в безкоштовних інтерактивних навчальних заходах, вебінарах або курсах, які пропонуються через платформу.

Розширюючи можливості гостьових користувачів, включаючи реєстрацію та призначення ролі студента за замовчуванням, система покращує залучення користувачів і полегшує безперешкодний перехід для осіб, зацікавлених у записі на мовні курси або доступі до навчальних матеріалів, які пропонує школа.

Після розширення MVP новими функціями система буде мати як мінімум дві додаткові ролі, кожна з яких має окремі права доступу.

Перша категорія це студенти. Студентом може стати будь-який гостьовий користувач після загальнодоступної реєстрації, бо всі користувачі після реєстрації

мають статус студента. Вони, як авторизовані користувачі, отримують доступ до своїх профілів для оновлення особистої інформації та налаштування параметрів облікового запису відповідно до своїх уподобань. Крім того, студенти матимуть можливість переходити на сторінку «Мої курси», де вони зможуть переглядати курси на які вони записані та відстежувати свій прогрес. Студенти матимуть доступ до сторінок конкретних курсів, щоб переглядати матеріали курсу та вивчати зміст уроків на спеціальних сторінках уроків. Додати студента до курсу може вчитель або адміністратор, самотійно студент не може записатись на курс.

Роль вчителя призначається адміністратором і мають цю роль винятково викладачі школи. Вони матимуть ширший діапазон дозволів, а саме:

- можливість створювати нові курси на платформі;
- змінювати наявний зміст курсу та розробляти індивідуальні уроки;
- зараховувати учнів на курс;
- переглядати статистику по курсу та результати студентів;
- подібно до студентів, викладачі матимуть можливість оновлювати свої профілі, щоб відобразити будь-які зміни в особистій інформації та керувати налаштуваннями свого облікового запису за потреби.

Впроваджуючи ці нові ролі та привілеї доступу, система зможе адаптувати функціональні можливості для задоволення потреб як студентів, так і викладачів у межах платформи вивчення мови. Цей стратегічний підхід гарантує, що користувачі зможуть ефективно взаємодіяти з платформою відповідно до своїх ролей і рівнів доступу.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Проектування архітектури системи аналізу даних та вебдодатку

Загальна архітектура системи для аналізу даних на платформі вивчення іноземних мов розділена на дві основні частини: архітектура вебдодатку та архітектура системи аналізу даних. Архітектура вебдодатку побудована навколо моделі клієнт-сервер, широко поширеної архітектури для створення інтерактивних вебдодатків. Ця модель розділяє систему на два основні компоненти: клієнтську сторону та серверну [6].

На стороні клієнта інтерфейс користувача та логіка програми розроблені для роботи у веббраузері користувача. Цей компонент відповідає за візуалізацію інтерфейсу користувача, обробку взаємодії користувача та надсилання запитів на сервер щодо даних або ресурсів. Для розробки на стороні клієнта зазвичай використовуються такі технології, як HTML, CSS і JavaScript.

На стороні сервера серверна інфраструктура керує запитами від клієнтів, обробляє дані та взаємодіє із зовнішніми системами. Цей компонент реалізує бізнес-логіку, обробку даних, аутентифікацію та надає дані клієнтам. Для розробки на стороні сервера зазвичай використовуються такі технології, як Node.js, Python, Java або Ruby on Rails.

Клієнт-серверна архітектура може бути дворівнева та трьохрівнева. На рисунку 3.1 зображено те, як може виглядати дворівнева архітектура, яка включає в себе клієнтів та сервер, дані між якими передаються мережею. В тривірневій архітектурі поруч з клієнтами та сервером також присутня база даних.

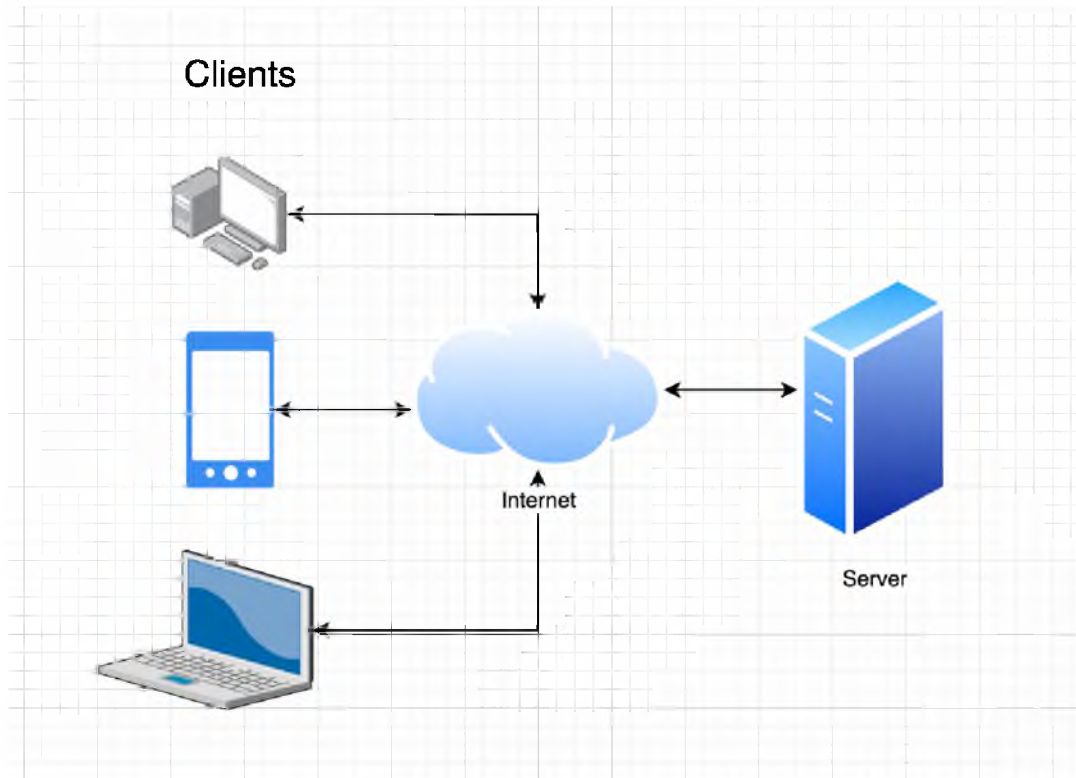


Рисунок 3.1 – Дворівнева клієнт-серверна архітектура

На рисунку 3.2 зображено трирівневу клієнт-серверну архітектуру.

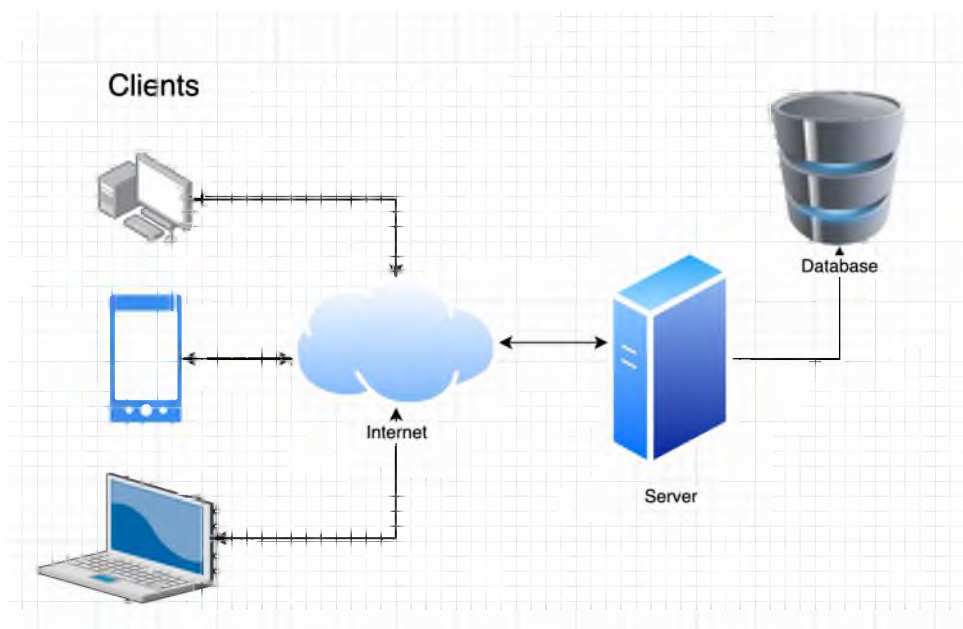


Рисунок 3.2 – Трирівнева клієнт-серверна архітектура

Архітектура системи аналізу даних для платформи вивчення мови відповідає класичній моделі ETL (Extract, Transform, Load). Ця архітектура побудована для полегшення обробки даних із різних джерел, їх перетворення у відповідний формат і завантаження в сховище даних для подальшої обробки та аналізу.

ETL процес можна описати наступним чином [7]:

- Extract: дані дістаються з різних джерел, включаючи вебсервера, бази даних і зовнішні API. Ці необроблені дані збираються та готуються для обробки;
- Transform: витягнуті дані піддаються трансформації, щоб зробити їх придатними для аналізу. Це може передбачати очищення, фільтрування, агрегування та збагачення даних для покращення їх якості;
- Load: перетворені дані завантажуються в сховище або озеро даних, централізоване сховище, призначене для зберігання великих обсягів структурованих і неструктурованих даних і керування ними. Це сховище даних служить основою для побудови подальших аналітичних звітів [8].

Для побудови аналітичних звітів дані отримуються із сховища даних. Різноманітні аналітичні інструменти, у тому числі SQL-запити, бібліотеки візуалізації даних і алгоритми машинного навчання, використовуються для аналізу даних і отримання корисної інформації. Потім ця інформація представлена зацікавленим сторонам через інформаційні панелі, звіти та інтерактивні візуалізації, що дає їм змогу приймати обґрунтовані рішення на основі даних. Ця архітектура системи аналізу даних на основі ETL дозволяє ефективно збирати, обробляти та аналізувати дані з платформи вивчення мови, надаючи зацікавленим сторонам цінну інформацію про поведінку користувачів, результативність курсів, рекламних компаній та загальне використання платформи.

Концептуально архітектура системи загалом, включаючи веб додаток та систему аналізу даних, зображено на рисунку 3.3, діаграму архітектури з конкретними технологіями буде розглянуто в наступних розділах роботи.

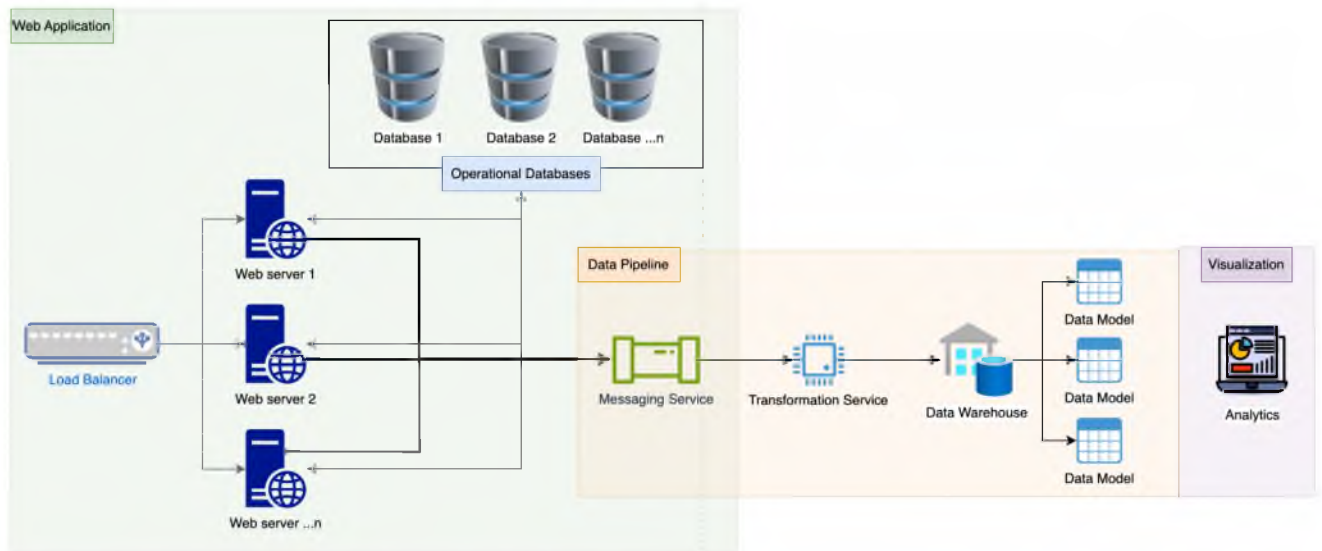


Рисунок 3.3 – Загальна архітектура системи аналізу даних платформи для вивчення іноземних мов

3.2 Дослідження та вибір найбільш прийнятних технологій для розробки ІС

Дослідження та вибір технологій для інформаційної системи привели до рішення розмістити як вебдодаток, так і конвеєр даних на Google Cloud Platform (GCP). Цей вибір був зумовлений кількома факторами. По-перше, GCP пропонує широкий набір технологій і послуг, призначених для розробки вебдодатків і обробки даних. Від сервісів масштабованої інфраструктури, таких як Compute Engine і Kubernetes Engine, до керованих сервісів, таких як Cloud SQL, BigQuery, Dataflow, Cloud Functions та Pub/Sub, GCP надає універсальну платформу для створення та розгортання складних систем [9].

Крім того, GCP оптимізує процес розробки та розгортання за допомогою своїх інструментів і послуг. Наприклад, Google App Engine та Cloud Run дозволяють легко розгорнути вебпрограми, а Cloud Dataflow спрощує розробку конвеєрів даних за допомогою керованого сервісу Apache Beam [10].

Крім того, GCP пропонує щедрий безкоштовний пробний період, що дозволяє розробникам досліджувати та експериментувати з різними службами без

значних витрат. Це дає змогу командам оцінити відповідність GCP до вимог їхнього проекту, перш ніж оформити платну підписку.

Використовуючи Google Cloud Platform як для вебдодатку, так і для системи аналізу даних, ми маємо змогу скористатися її надійною інфраструктурою, масштабованими службами та зручними для розробників інструментами для створення надійної та ефективної інформаційної системи. Крім того, обширна документація GCP і підтримка спільноти додатково полегшують процес розробки, забезпечуючи успішне впровадження та розгортання системи.

Для створення вебдодатку було розглянуто різноманітні технології, включаючи Django на основі Python, і React з JavaScript. Однак, після ретельної оцінки фреймворк Next.js був обраний як кращий варіант. На рішення вплинуло кілька факторів.

Django, відомий своєю можливістю створювати надійній бекенд і тим, що багато розповсюдженого функціоналу, такого як аутентифікація в Django працює out of the box [11]. Але також Django має суттєві обмеження у реалізації динамічного клієнтського інтерфейсу. З іншого боку React вирішує цю проблему. Але те, що код React додатку виконується в браузері, вимагає окремого бекенд сервера для зв'язку з базою даних та іншими компонентами системи. Крім того, генерація динамічного контенту за допомогою React створює проблеми для оптимізації SEO, важливого аспекту для вебдодатків. Хоча для React існують рішення для генерації сторінок на стороні сервера, впровадження цих рішень потребує суттєвих часових затрат.

Next.js став відповідним рішенням, використовуючи React, JavaScript та Node.js для створення full-stack вебдодатків як з клієнтським, так і з серверним кодом [12]. Такий підхід оптимізує процеси розробки та розгортання, зменшуючи потребу в окремих бекенд серверах. Крім того, Next.js вирішує проблему SEO за допомогою рендерингу на стороні сервера, покращуючи видимість вебсайту та рейтинг у пошуковій системі. Крім того, Next.js полегшує розробку API, забезпечуючи можливість мати вебдодаток та API в одному місці.

Таким чином, Next.js було обрано через його здатність оптимізувати розробку, вирішувати проблеми оптимізації SEO та підтримувати розробку full-stack додатку з вбудованим серверним рендерингом і можливостями побудови API. Це рішення узгоджується з цілями проекту щодо ефективності, масштабованості та видимості в пошукових системах.

Вебдодаток Next.js було контейнеризовано за допомогою Docker і розгорнуто в сервісі Cloud Run на Google Cloud Platform (GCP). Cloud Run дозволяє запускати контейнери Docker і автоматично масштабувати їх у відповідності до навантаження [13].

Контейнеризація програми Next.js за допомогою Docker забезпечує створення портативного середовища для розгортання. Контейнери Docker інкапсулюють усі залежності та конфігурації, забезпечуючи впевненість у тому, що програма буде успішно виконуватись в різних середовищах.

Розгортання контейнеризованої програми в Cloud Run дозволить використати масштабованість і гнучкість безсерверних обчислень. Cloud Run автоматично керує основною інфраструктурою, дозволяючи програмі динамічно масштабуватися на основі вхідного трафіку або коливань робочого навантаження.

Використовуючи Cloud Run, додаток Next.js отримує переваги від ефективного використання ресурсів, автоматичного масштабування та спрощеного керування інфраструктурою. Такий підхід до розгортання відповідає сучасним практикам DevOps і дозволяє програмі обробляти різні рівні трафіку, одночасно оптимізуючи економічну ефективність і продуктивність.

Операційна база даних для проекту буде розміщена в Cloud SQL на Google Cloud Platform (GCP) з використанням PostgreSQL як системи керування реляційною базою даних (RDBMS).

Cloud SQL пропонує повністю керовану службу бази даних, яка спрощує завдання адміністрування бази даних, такі як масштабування та обслуговування [14]. Розміщуючи оперативну базу даних у Cloud SQL, ми отримали вигоду від високої доступності, автоматичного резервного копіювання та бездоганної інтеграції з іншими службами GCP.

PostgreSQL — це потужна та багатofункціональна реляційна СУБД з відкритим вихідним кодом, яка широко використовується завдяки своїй надійності, продуктивності та стійкості [15]. Завдяки підтримці розширених функцій SQL, цілісності даних і керуванню транзакціями PostgreSQL добре підходить для критично важливих операційних баз даних.

Використовуючи Cloud SQL з PostgreSQL, ми забезпечили надійність, масштабованість і безпеку своєї операційної бази даних, мінімізуючи адміністративні витрати.

Для системи аналізу даних необхідна центральна точка входу, куди вебдодаток може надійно надсилати створені користувачем події без ризику втрати даних. Для втілення цієї потреби було обрано сервіс обміну повідомленнями Google Pub/Sub.

Google Pub/Sub є надійною та масштабованою платформою обміну повідомленнями, яка забезпечує надійну передачу даних між різними компонентами системи [16]. Асинхронне та роз'єднане спілкування втілюється в життя через теми та підписки, що забезпечує ефективну доставку та обробку повідомлень.

Завдяки інтеграції Google Pub/Sub в архітектуру конвеєра даних вебдодаток може публікувати події за визначеними темами, знаючи, що вони будуть безпечно зберігатися та доставлятися в чергу для подальшої обробки. Наступним кроком сервіси обробки даних можуть підписатися на ці теми, щоб споживати та обробляти вхідні події вчасно.

Загалом Google Pub/Sub діє як надійна точка входу в конвеєр даних, сприяючи безперебійній відправці даних з вебдодатку до інших компонентів системи, забезпечуючи при цьому надійність і цілісність даних. Цей вибір узгоджується з вимогами системи щодо масштабованості, надійності та обробки подій у реальному часі.

Як важливий компонент для обробки даних системи було обрано Google Cloud Functions. Ці функції можна викликати у відповідь на події, такі як дані про події користувача, надіслані до теми Pub/Sub. Вони пропонують безсерверну

архітектуру, яка автоматично масштабується відповідно до вимог робочого навантаження. Крім того, їх легко реалізувати, що робить їх ідеальним вибором для нашого проекту.

Google Cloud Functions викликаються у відповідь на події забезпечуючи безперебійну обробку даних в реальному часі [17]. Використовуючи Python, обрану мною мову програмування, ці функції ефективно вирішують завдання обробки даних, такі як збагачення даних, перетворення та завантаження у сховище даних.

Завдяки використанню Google Cloud Functions система отримує переваги безсерверного, масштабованого та економічно ефективного підходу до обробки даних. Завдяки автоматичному масштабуванню та керованому подіями виконанню ці функції ефективно справляються з коливаннями робочого навантаження, мінімізуючи накладні витрати. Цей вибір відповідає вимогам проекту щодо гнучкості, масштабованості та простоти впровадження.

BigQuery було обрано у якості сховища даних через його надійність та бездоганну інтеграцію з іншими сервісами в середині Google Cloud Platform. Як повністю кероване безсерверне сховище даних BigQuery пропонує неперевершену масштабованість, що дозволяє організаціям аналізувати величезні обсяги даних без потреби підтримувати інфраструктуру [18]. Розподілена архітектура та стовпчастий формат зберігання даних забезпечують швидку роботу запитів, надаючи користувачам змогу отримувати інформацію з даних у режимі реального часу. Крім того, повна інтеграція BigQuery з іншими службами Google Cloud і популярними інструментами аналізу даних спрощує робочі процеси і полегшує розробку. Завдяки надійним функціям безпеки та сертифікатам відповідності BigQuery забезпечує безпеку даних і їх відповідність нормативним вимогам.

Щоб полегшити перетворення даних у BigQuery, було прийнято рішення використовувати dbt (інструмент створення трансформацій даних), який виконуватиметься та керуватиметься в dbt Cloud. Dbt відповідатиме за створення моделей даних, адаптованих до вимог проекту.

Dbt — це потужний інструмент, розроблений спеціально для задач трансформації даних і моделювання [19]. Використовуючи можливості dbt, інженери даних можуть визначати та створювати моделі даних безпосередньо в BigQuery, спрощуючи процес перетворення сирих даних у структуровані набори даних, готові до аналізу.

Dbt Cloud забезпечує кероване середовище для виконання та планування завдань dbt, пропонуючи такі функції, як контроль версій, документація та інструменти для співпраці [20]. Це дозволяє командам ефективно співпрацювати над завданнями моделювання даних і забезпечувати послідовність і надійність у процесі трансформації.

Завдяки інтеграції dbt в архітектуру системи аналізу даних і використанню його функцій у хмарі dbt, проект отримує переваги від спрощених робочих процесів моделювання даних, версіонування, а також покращеної якості та узгодженості даних. Цей вибір узгоджується з цілями проекту щодо ефективності, масштабованості та надійності процесів перетворення даних і моделювання.

Google Looker Studio було обрано для створення інтерактивних аналітичних дашбордів на основі моделей даних, створених у BigQuery після перетворень dbt.

Looker Studio пропонує комплексну платформу для створення візуалізацій даних, звітів і інформаційних панелей [21]. Завдяки бездоганній інтеграції з BigQuery Looker Studio дозволяє користувачам використовувати перетворені моделі даних для створення візуалізацій і отримання розуміння своїх даних.

За допомогою Looker Studio користувачі можуть легко створювати та налаштовувати інтерактивні інформаційні дашборди відповідно до потреб свого бізнесу. Платформа надає широкий спектр опцій візуалізації, включаючи діаграми, графіки та карти, що дозволяє користувачам досліджувати дані з різних точок зору та знаходити цінну інформацію.

Використовуючи Looker Studio для створення інтерактивних аналітичних панелей, організації можуть розкрити повний потенціал своїх даних, дозволяючи зацікавленим сторонам приймати обґрунтовані рішення та стимулювати розвиток бізнесу. Цей вибір узгоджується з цілями проекту, які стосуються прийняття

рішень на основі даних і надання користувачам можливості отримувати практичні висновки зі своїх даних.

На рисунку 3.4 зображено архітектуру системи з урахуванням обраних та описаних вище технологій.

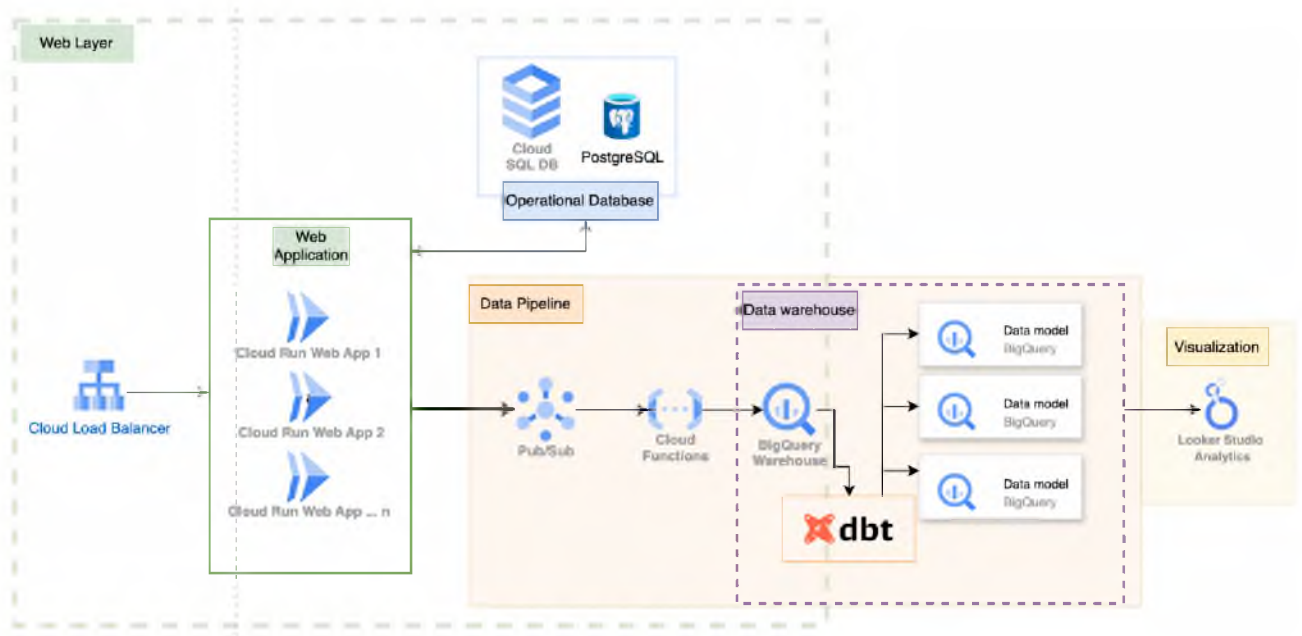


Рисунок 3.4 – Архітектура системи з обраними технологіями

3.3 Визначення сутностей і зв'язків у системі та проектування операційної бази даних

У MVP версії системи визначено три таблиці для керування сутностями та їхніми зв'язками.

Таблиця `messengers` зберігає інформацію про різні месенджери, які використовуються для спілкування, містить поля для ідентифікатора та імені месенджера.

Таблиця `lesson_request_statuses` зберігає різні статуси, які може мати запит на пробний урок, включаючи поля для ідентифікатора статусу, імені та опису.

Таблиця `trial_lesson_requests` представляє запити на пробні уроки, надіслані користувачами. Він містить поля для ідентифікатора запиту, позначки часу

надсилання, ім'я, прізвища, електронної пошти, номера телефону, додаткової інформації, а також зовнішніх ключів, які посилаються на таблицю з месенджерами та таблицю `lesson_request_statuses` для встановлення зв'язків. Зокрема, ідентифікатор месенджера вказує на месенджер, який використовується для спілкування, тоді як ідентифікатор статусу вказує на поточний статус запиту.

У міру розвитку системи архітектура бази даних буде розширюватися та змінюватись, щоб включати такі сутності, як користувачі, уроки, курси, дозволи тощо. Ці таблиці зберігатимуть відповідну інформацію про користувачів, уроки, курси, дозволи, дозволяючи системі ефективно керувати доступом користувачів, реєстрацією на курси, створенням уроків та дозволами. У міру додавання нових функцій база даних продовжуватиме рости й адаптуватись відповідно до змінних потреб системи.

Ці таблиці спільно визначають сутності та їхні зв'язки в операційній базі даних. На рисунку 3.5 зображено ER-діаграму описаних вище зв'язків та сутностей [22].

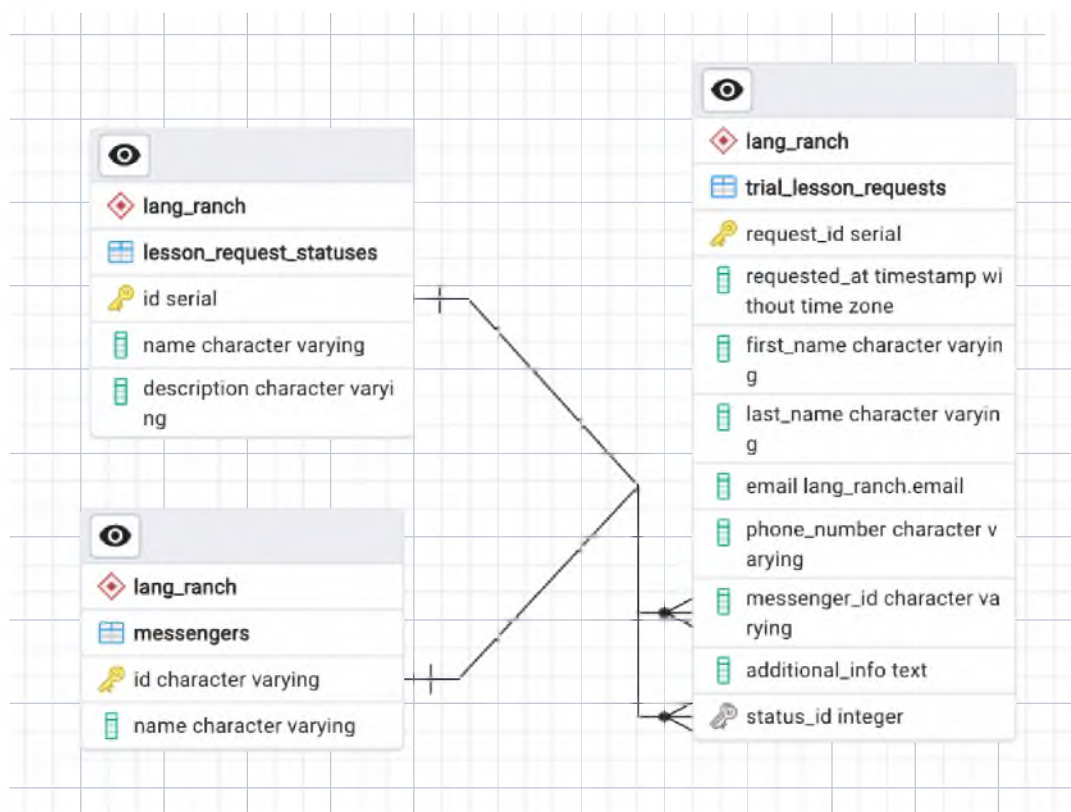


Рисунок 3.5 – ER-діаграма MVP версії операційної бази даних

3.4 Визначення подій користувача для відправки в систему аналізу даних

У MVP версії системи було визначено три ключові події користувача, які відправляються в систему аналізу даних для подальшої обробки, валідації, запису в сховище даних та візуалізації.

По-перше, подія `session_start` позначає початок сесії користувача після відвідування платформи. Ця подія відправляється, коли користувач вперше відкриває вебсайт, ця подія не має специфічних саме для неї полів, але містить метадані, отримані із HTTP запиту, які є частиною інших подій також. Ці метадані інкапсулюють важливу інформацію про користувача, зокрема `session_id`, IP адресу, географічне розташування (країна, регіон, місто), деталі пристрою (браузер, операційна система, тип пристрою, виробник, модель) і те, чи є відвідувач ботом.

Подія `page_view` відображає шлях навігації користувача нашим вебсайтом. Ця подія відправляється щоразу, коли користувач відкриває певну сторінку, включає такі властивості, як `path_name`, що вказує на назву сторінки, і `path_changes_count`, ціле число, що вказує послідовність завантаження сторінки під час сеансу користувача. Завдяки запису цих подій про відвідування сторінок ми можемо створити цілісне розуміння шляху та поведінки користувача під час відвідування.

Остання подія `trial_lesson_request` відправляється коли користувач надсилає форму запити на пробний урок. Ця подія наповнена набором властивостей, що охоплюють усі введені користувачем дані, починаючи від особистих даних, таких як ім'я, електронна пошта та номер телефону, до таких ідентифікаторів, як `request_id` і `requested_at timestamp`.

Завдяки відслідковуванню цих подій та деталей закладено основу для надійної системи аналізу даних, яка дає змогу отримувати корисну інформацію, покращувати взаємодію з користувачами та сприяти прийняттю обґрунтованих рішень.

4 РОЗРОБКА ТА ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Програмна реалізація вебдодатку

Вебдодаток було реалізовано за допомогою JavaScript фреймворку Next.js, який дозволяє будувати динамічні full-stack вебдодатки. На рисунку 4.1 можна побачити структуру коду вебсайту.

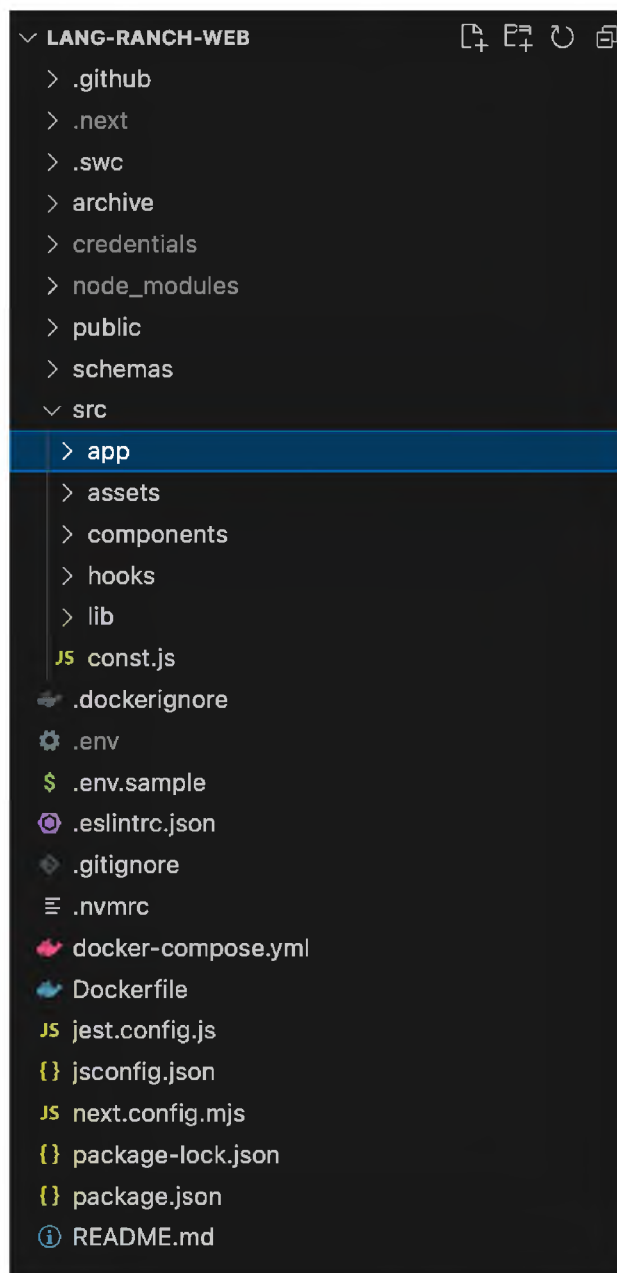


Рисунок 4.1 – Структура коду вебдодатку

Конфігурація проекту зберігається в файлах `next.config.mjs` та `jsconfig.json`. В директорії `src` зберігається основний код проекту [23]. В директорії `schemas` зберігаються json схеми, які використовуються для валідації подій користувача і не тільки. На рисунку 4.2 можна побачити структуру директорії зі схемами та приклад схеми для валідації події `page_view`.

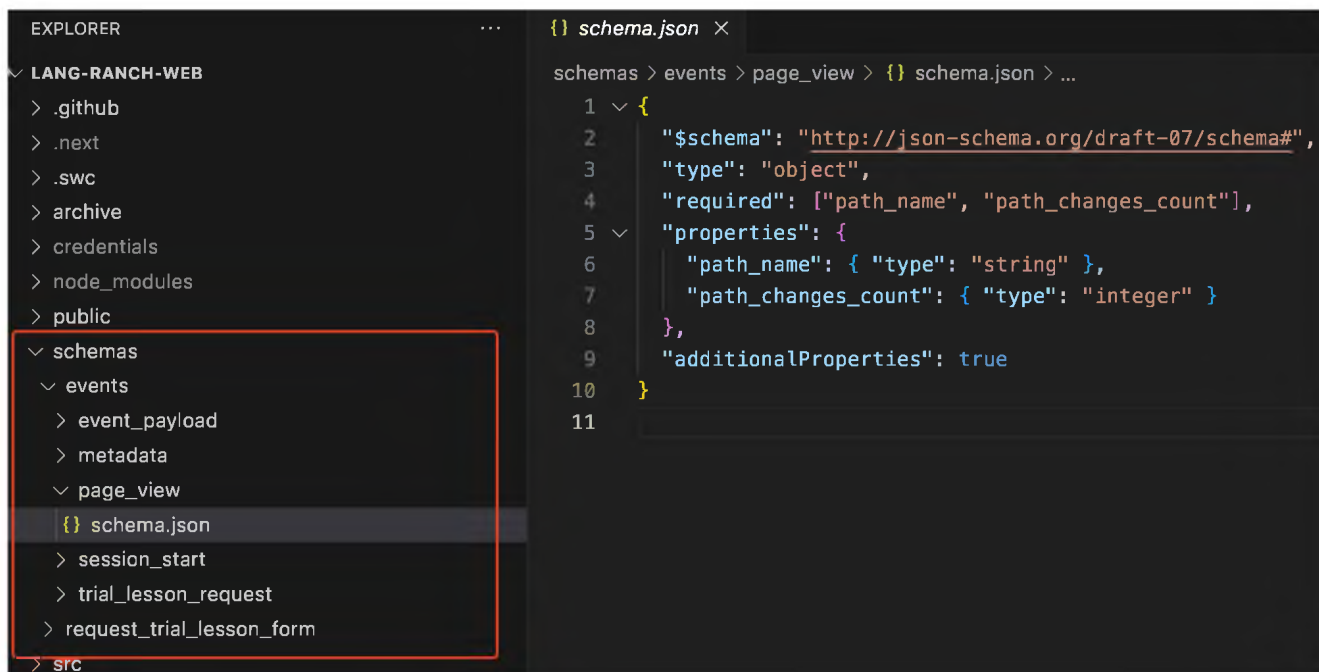


Рисунок 4.2 – Схема для валідації події `page_view`

На цій схемі можна побачити, що подія `page_view` повинна мати два поля: `path_name` та `path_changes_count`. Поле `path_name` має бути строкою, а `path_changes_count` має бути цілим числом.

В середині директорії `src` є директорія `app`, яка створена у відповідності до вимог Next.js. Файл `globals.css` зберігає глобальні стилі, які застосовуються для всіх сторінок вебсайту. На рисунку 4.3 можна побачити структуру цієї директорії `app`.

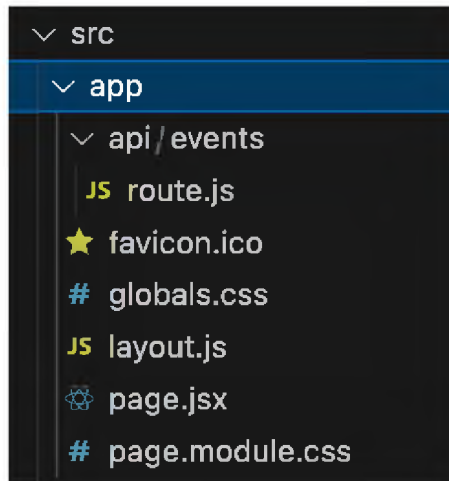


Рисунок 4.3 – Структура директорії app

У файлі `page.jsx` написана структура головної сторінки вебсайту, вміст цього файлу можна побачити на рисунку 4.4.

```

page.jsx
src > app > 🌀 page.jsx > ...
  5  import SectionIntro from "@components/landing/sectionIntro/sectionIntro";
  7  import SectionUsefulMaterials from "@components/landing/sectionUsefulMaterials/sectionUsefulMaterials";
  8  import styles from "./page.module.css";
  9
 10  import ToastContainerWrapper from "@components/ui/ToastContainerWrapper";
 11
 12  export default function Home() {
 13    return (
 14      <>
 15        <ToastContainerWrapper />
 16        <HeaderBackground>
 17          <Header />
 18          <SectionIntro />
 19        </HeaderBackground>
 20        <main className={styles.mainSection}>
 21          <div className={styles.gradientBackground}></div>
 22          <SectionAbout />
 23          <SectionUsefulMaterials />
 24          <SectionContact />
 25          <Footer />
 26        </main>
 27      </>
 28    );
 29  }
 30
  
```

Рисунок 4.4 – Вміст файлу page.jsx

Головна сторінка складається із header, вступної секції, секції з інформацією про мовну школу, секції із корисними матеріалами, секції з формою для запису на пробне заняття та footer.

У директорії assets, яку зображено на рисунку 4.5, зберігаються картинки, які використані на деяких сторінках.

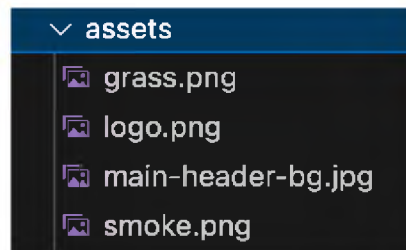


Рисунок 4.5 – Вміст директорії assets

В директорії components зберігаються React компоненти за допомогою, яких побудовано вебсторінку. Як приклад, на рисунку 4.6 наведено компонент Analytics.js.

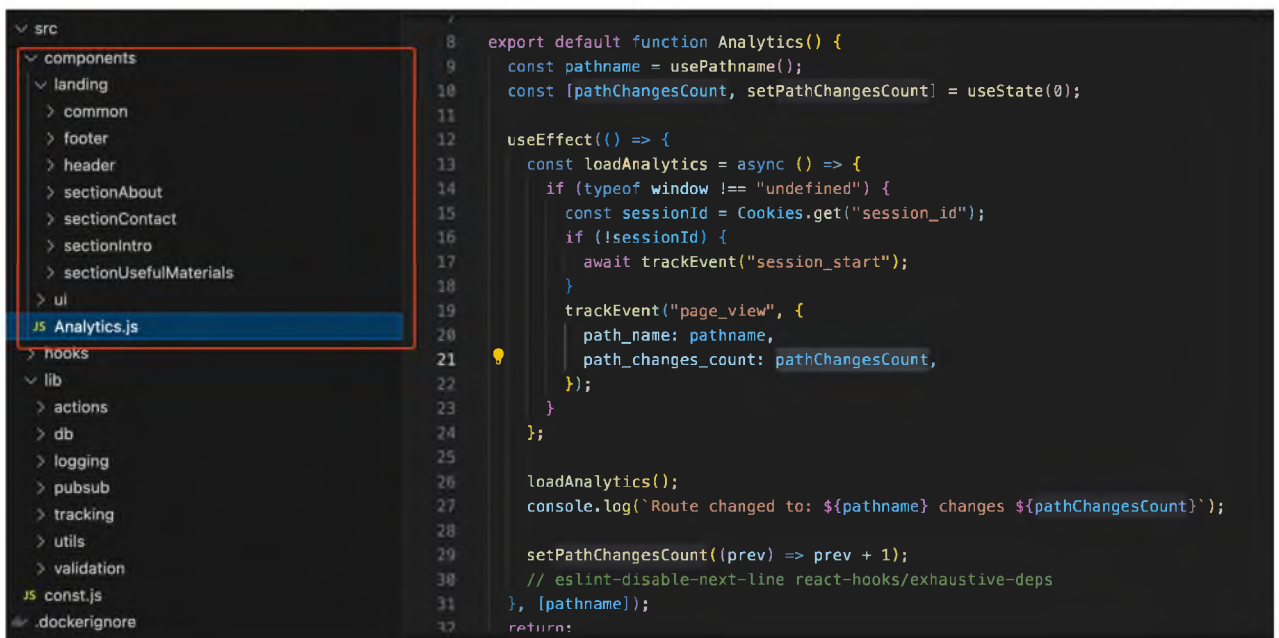


Рисунок 4.6 – React компонент Analytics.js

Цей компонент завантажується коли користувач відкриває вебсторінку, та, якщо в cookies збережено `session_id`, відправляє тільки подію `page_view`, а якщо `session_id` немає, то це означає ще користувач вперше відвідує вебсайт і в такому випадку також відправляється подія `session_start`.

У директорії `lib`, яку зображено на рисунку 4.7, зберігається різний функціонал вебсерверу, який не стосується на пряму до відображення елементів на сторінках.

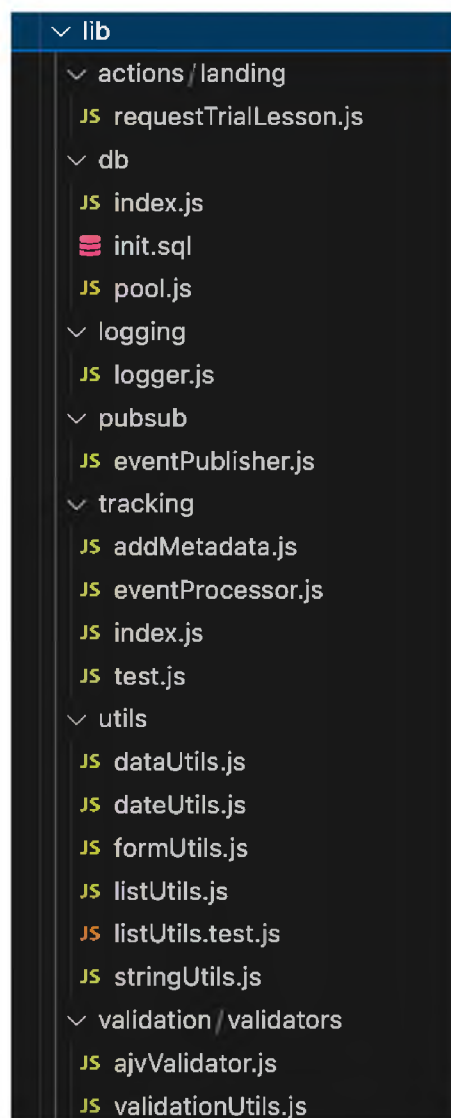


Рисунок 4.7 – Вміст директорії `lib`

У директорії `lib/actions`, зберігається код для `server actions`, це спеціальний функціонал, який пропонує Next.js. `Server actions` — це асинхронні функції, які

виконуються на сервері [24]. Їх можна використовувати в серверних і клієнтських компонентах для обробки надсилання форм і мутацій даних у програмах Next.js. В цьому випадку було написано серверну функцію `requestTrialLesson` для обробки результату надсилання користувачем форми для запису на заняття.

У директорії `lib/db`, зберігається код для взаємодії з базою даних PostgreSQL. Для цього було обрано бібліотеку `pg` для `node.js`. На рисунку 4.8 зображено модуль зі створенням підключення до бази даних.

```
import { Pool } from "pg";

import logger from "../logging/logger";

const pool = new Pool({
  user: process.env.POSTGRES_USER,
  database: process.env.POSTGRES_DB,
  password: process.env.POSTGRES_PASSWORD,
  host:
    process.env.NODE_ENV === "development"
      ? "127.0.0.1"
      : `/${cloudsql}/${process.env.CLOUD_SQL_CONNECTION_NAME}`,
  port: 5432,
  options: `-c search_path=${process.env.POSTGRES_SCHEMA}`,
});
logger.info("Initialized db connection pool");

export default pool;
```

Рисунок 4.8 – Вміст `lib/db/pool.js`

У директорії `lib/logging`, зберігається код для логування інформаційних повідомлень, помилок та інших повідомлень.

У директорії `lib/pubsub`, зберігається код пов'язаний із взаємодією із сервісом Google Pub/Sub. На рисунку 4.9 зображено код функції `publishMessage`, яка відповідає за відправку повідомлень в тему Pub/Sub.

```
JS eventPublisher.js ×
src > lib > pubsub > JS eventPublisher.js > publishMessage
1  import { PubSub } from "@google-cloud/pubsub";
2
3  import { PUB_SUB_TOPICS } from "@/const";
4  import logger from "../logging/logger";
5
6  const pubSubClient = new PubSub();
7  ⚡
8  export async function publishMessage(
9    data,
10   topicNameOrId = PUB_SUB_TOPICS.userEventsTopic
11 ) {
12   const dataBuffer = Buffer.from(
13     typeof data === "string" ? data : JSON.stringify(data)
14   );
15
16   try {
17     const messageId = await pubSubClient
18       .topic(topicNameOrId)
19       .publishMessage({ data: dataBuffer });
20     logger.debug(`Message ${messageId} published.`);
21   } catch (error) {
22     logger.error(`Received error while publishing: ${error.message}`);
23   }
24 }
```

Рисунок 4.9 – Функція publishMessage для відправки повідомлень в Pub/Sub

У директорії lib/tracking, зберігається код пов'язаний із обробкою та відправкою подій в систему аналізу даних. На рисунку 4.10 продемонстровано функцію postEvent, яка відправляє подію з клієнта на API endpoint на сервері для подальшої обробки, валідації, трансформації та відправки в систему аналізу даних.

```

c > lib > tracking > JS index.js > postEvent
1  import { v4 as uuidv4 } from "uuid";
2
3  import processEvent from "./eventProcessor";
4
5  async function postEvent(eventPayload) {
6      const res = await fetch("/api/events", {
7          method: "POST",
8          headers: {
9              "Content-Type": "application/json",
10         },
11         body: JSON.stringify(eventPayload),
12     });
13     console.log(res);
14     return await res.json();
15 }

```

Рисунок 4.10 – Функція postEvent

У директорії lib/validation, зберігається код пов'язаний із валідацією даних у вебдодатку відносно json схем. На рисунку 4.11 зображено код класу SchemasHolder, якій відповідає за зчитування та зберігання в пам'яті схем.

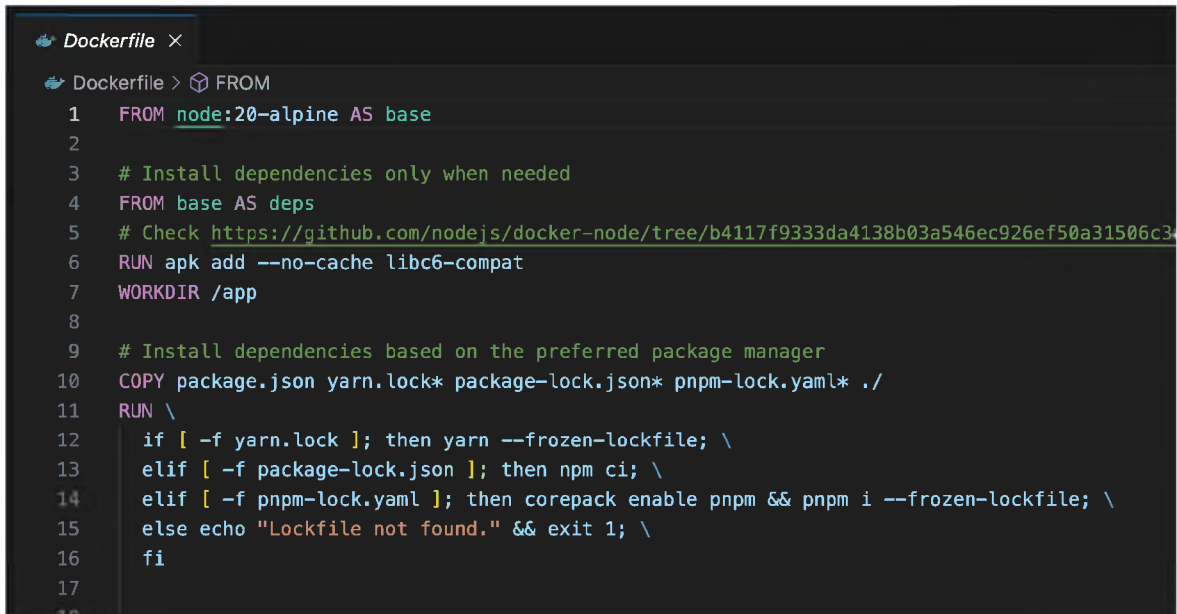
```

11 class SchemasHolder {
12     constructor() {
13         if (SchemasHolder._instance) {
14             return SchemasHolder._instance;
15         }
16         SchemasHolder._instance = this;
17
18         this.schemas = {};
19         logger.info("Initialized Validation Schemas Holder.");
20     }
21
22     getSchema(schemaName) {
23         schemaName = removeTrailingSlash(schemaName);
24         let schema = this.schemas[schemaName];
25         if (!schema) {
26             schema = readJsonSchema(schemaName);
27             this.schemas[schemaName] = schema;
28         }
29         return schema;
30     }
31 }

```

Рисунок 4.11 – Клас SchemasHolder

Також проєкт було контейнеризовано для виконання в docker, бо він буде вивантажений у сервіс Cloud Run. На рисунку 4.12 продемонстровано фрагмент Dockerfile, який зберігає інструкції для побудови docker image.



```
Dockerfile x
Dockerfile > FROM
1 FROM node:20-alpine AS base
2
3 # Install dependencies only when needed
4 FROM base AS deps
5 # Check https://github.com/nodejs/docker-node/tree/b4117f9333da4138b03a546ec926ef50a31506c3
6 RUN apk add --no-cache libc6-compat
7 WORKDIR /app
8
9 # Install dependencies based on the preferred package manager
10 COPY package.json yarn.lock* package-lock.json* pnpm-lock.yaml* ./
11 RUN \
12   if [ -f yarn.lock ]; then yarn --frozen-lockfile; \
13   elif [ -f package-lock.json ]; then npm ci; \
14   elif [ -f pnpm-lock.yaml ]; then corepack enable pnpm && pnpm i --frozen-lockfile; \
15   else echo "Lockfile not found." && exit 1; \
16   fi
17
18
```

Рисунок 4.12 – Фрагмент Dockerfile

4.2 Розгортання вебдодатку

Після завершення етапу розробки вебдодатку наступним важливим кроком стало його розгортання. Щоб розпочати цей процес, було створено репозиторій на GitHub для розміщення вихідного коду програми. Цей репозиторій служить центральним репозиторієм для контролю версій і співпраці між членами команди.

Після створення репозиторію, він постійно оновлюється новим кодом, щоб всі відповідні файли, залежності та конфігурації зберігались в одному місці і були постійно доступними. Цей крок не лише полегшив керування кодом, але й уможливив continuous integration/continuous deployment (CI/CD) [25].

Використовуючи платформу GitHub, було створено спрощений робочий процес для керування кодом і співпраці. Це стало основою для подальших кроків розгортання, включаючи конфігурацію CI/CD і розгортання в хмарі Google Cloud

Platform (GCP). На рисунку 4.13 зображено репозиторій lang-ranch-web, який зберігає код вебдодатку.

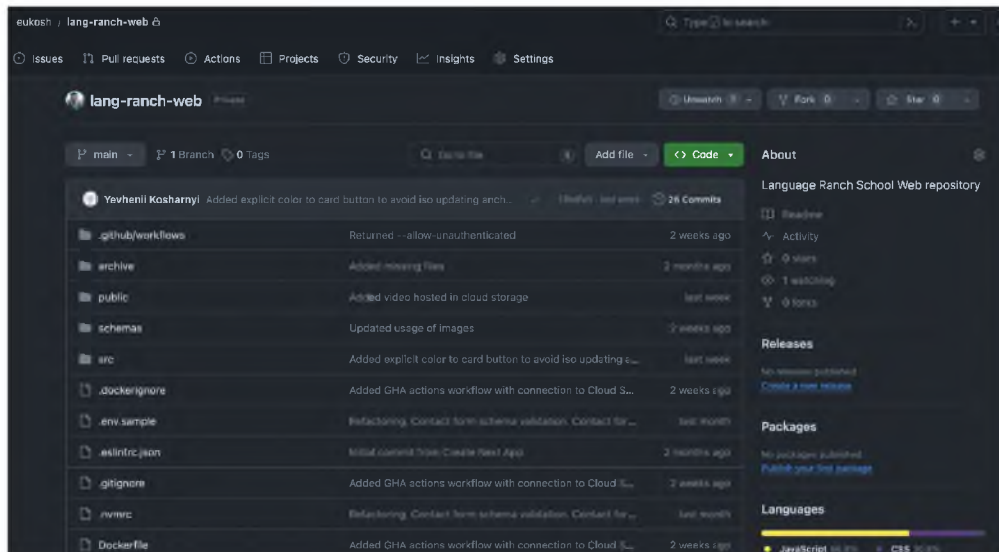


Рисунок 4.13 – Репозиторій lang-ranch-web

Всі компоненти системи було вирішено вивантажувати в хмару GCP, вебдодаток було запущено за допомогою сервісу Cloud Run. Для цього було створено акаунт в GCP, на рисунку 4.14 зображено головний екран платформи.

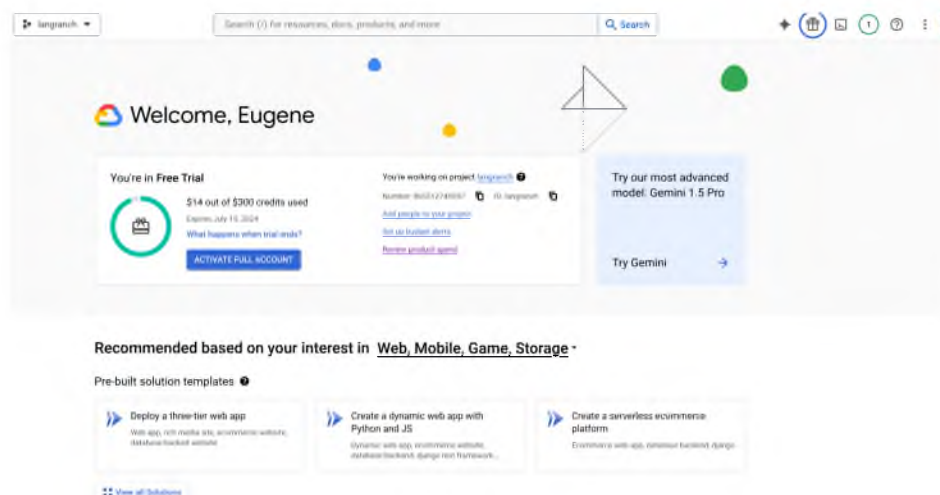
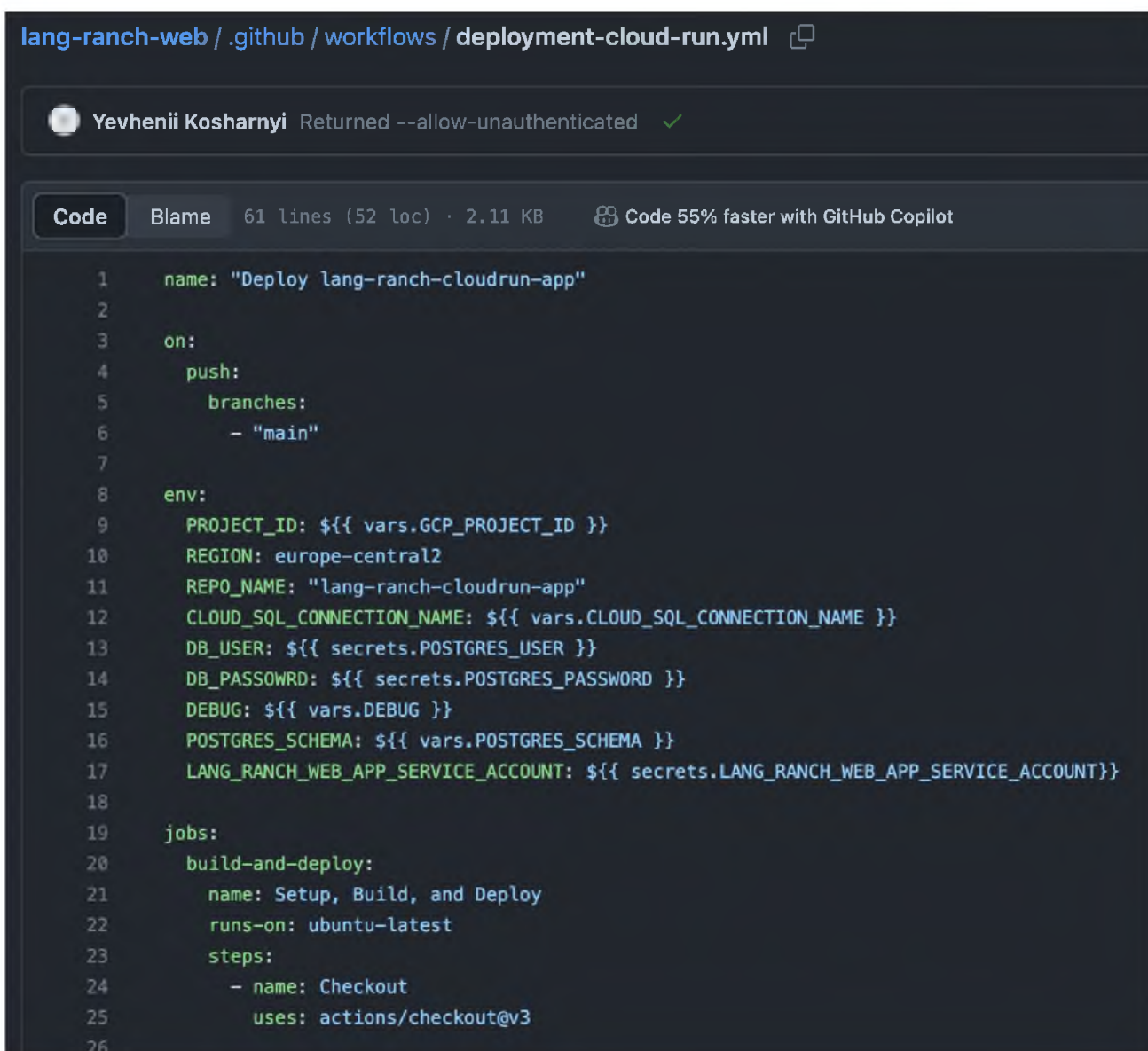


Рисунок 4.14 – Головний екран платформи GCP

Було використано GitHub workflows, щоб реалізувати процес continuous deployment (CD) для вебдодатку. Цей workflow автоматизує розгортання сервісу

Cloud Run, забезпечуючи безперервне оновлення щоразу, коли зміни надсилаються до репозиторію. Використовуючи GitHub actions, workflow організовує створення програми, її контейнеризацію за допомогою Docker і розгортання в Cloud Run. Цей автоматизований процес спрощує оновлення, зменшує ручне втручання та підтримує надійний конвеєр розгортання. На рисунку 4.15 продемонстровано фрагмент відповідного workflow.

The image shows a screenshot of a GitHub repository file named 'deployment-cloud-run.yml'. The file is a workflow configuration in YAML format. It is triggered on a 'push' to the 'main' branch. The environment variables include project ID, region, repository name, database connection details, and service account. A job named 'build-and-deploy' is defined, which runs on 'ubuntu-latest' and includes a 'Checkout' step using 'actions/checkout@v3'.

```
lang-ranch-web / .github / workflows / deployment-cloud-run.yml
Yevhenii Kosharnyi Returned --allow-unauthenticated ✓
Code Blame 61 lines (52 loc) · 2.11 KB Code 55% faster with GitHub Copilot
1 name: "Deploy lang-ranch-cloudrun-app"
2
3 on:
4   push:
5     branches:
6       - "main"
7
8   env:
9     PROJECT_ID: "${{ vars.GCP_PROJECT_ID }}"
10    REGION: europe-central2
11    REPO_NAME: "lang-ranch-cloudrun-app"
12    CLOUD_SQL_CONNECTION_NAME: "${{ vars.CLOUD_SQL_CONNECTION_NAME }}"
13    DB_USER: "${{ secrets.POSTGRES_USER }}"
14    DB_PASSWRD: "${{ secrets.POSTGRES_PASSWORD }}"
15    DEBUG: "${{ vars.DEBUG }}"
16    POSTGRES_SCHEMA: "${{ vars.POSTGRES_SCHEMA }}"
17    LANG_RANCH_WEB_APP_SERVICE_ACCOUNT: "${{ secrets.LANG_RANCH_WEB_APP_SERVICE_ACCOUNT }}"
18
19   jobs:
20     build-and-deploy:
21       name: Setup, Build, and Deploy
22       runs-on: ubuntu-latest
23       steps:
24         - name: Checkout
25           uses: actions/checkout@v3
26
```

Рисунок 4.15 – Фрагмент workflow для розгортання Cloud Run

Як тільки нові зміни потрапляють в main гілку Github репозиторію, цей workflow активується і розгортає нову версію вебдодатку. На рисунку 4.16 продемонстровано розгорнутий вебдодаток на платформі Cloud Run.

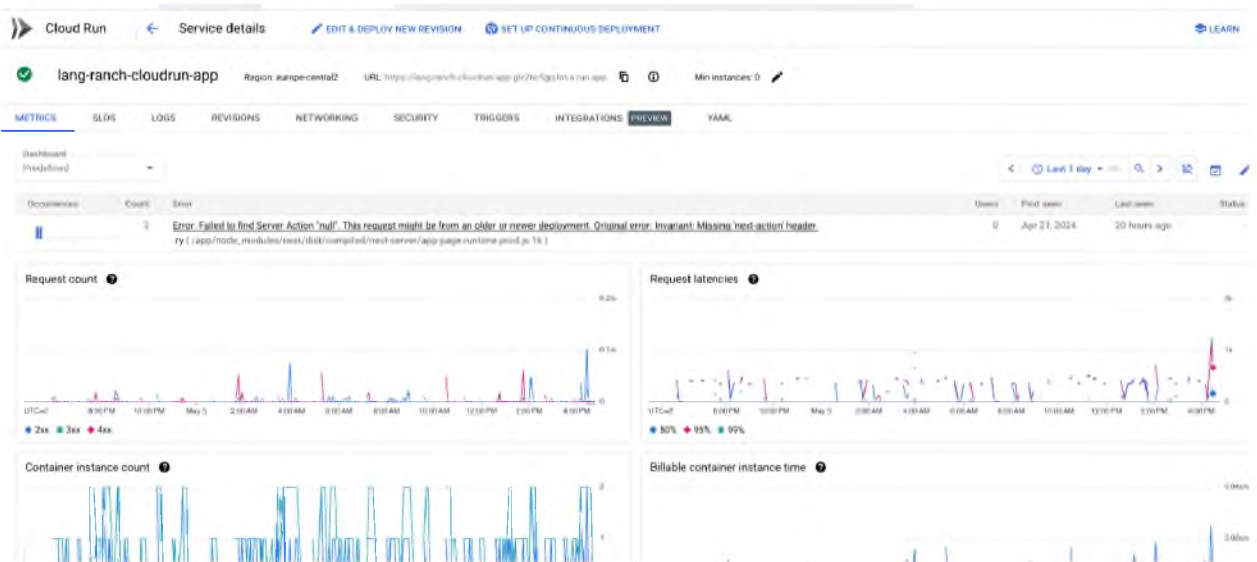


Рисунок 4.16 – Розгорнутий вебдодаток на платформі Cloud Run

На рисунку 4.17 зображено успішні запуски раніше створеного GitHub workflow.

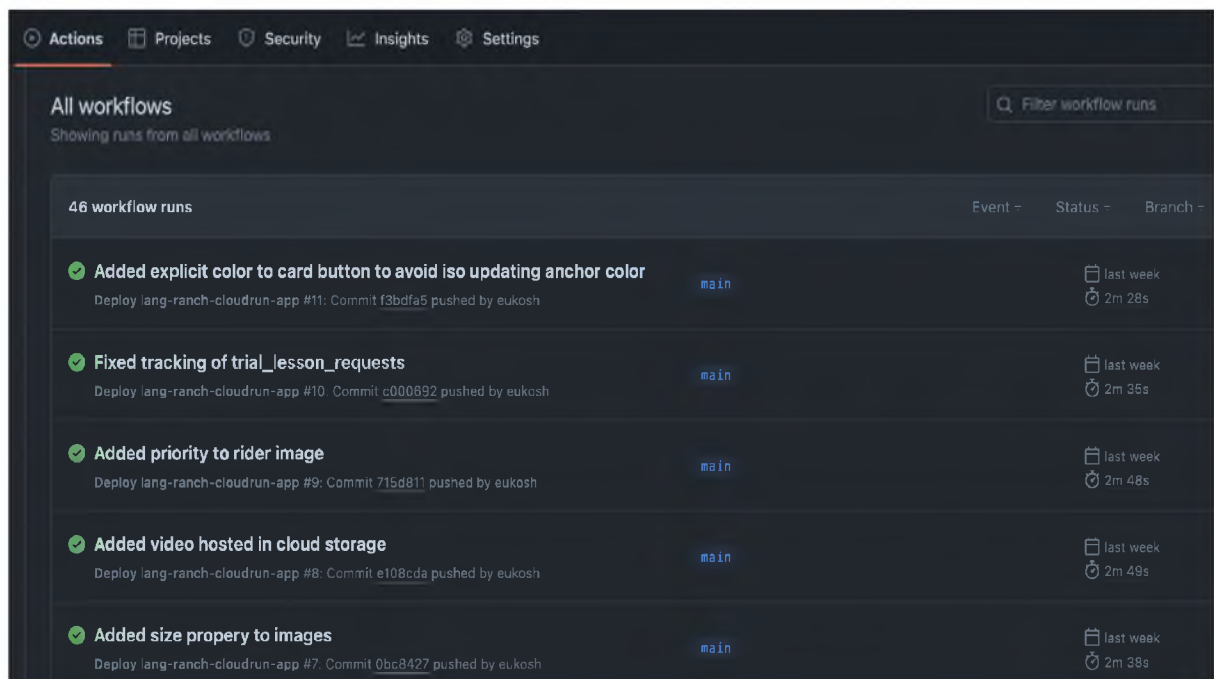


Рисунок 4.17 – Успішні запуски GitHub workflow

Після ретельного дослідження було вирішено придбати доменне ім'я languageranch.com у Namecheap, авторитетного постачальника доменів. Це доменне ім'я було обрано після ретельного розгляду, щоб якнайкраще представити наш вебсайт і відповідати нашій стратегії брендингу. Завдяки надійним послугам Namecheap ми забезпечили бажане доменне ім'я, заклавши основу для створення сильної присутності нашої платформи в Інтернеті.

На рисунку 4.18 зображено інтерфейс провайдера Namecheap та придбаний домен із відповідним IP адресом налаштованим на платформі GCP.

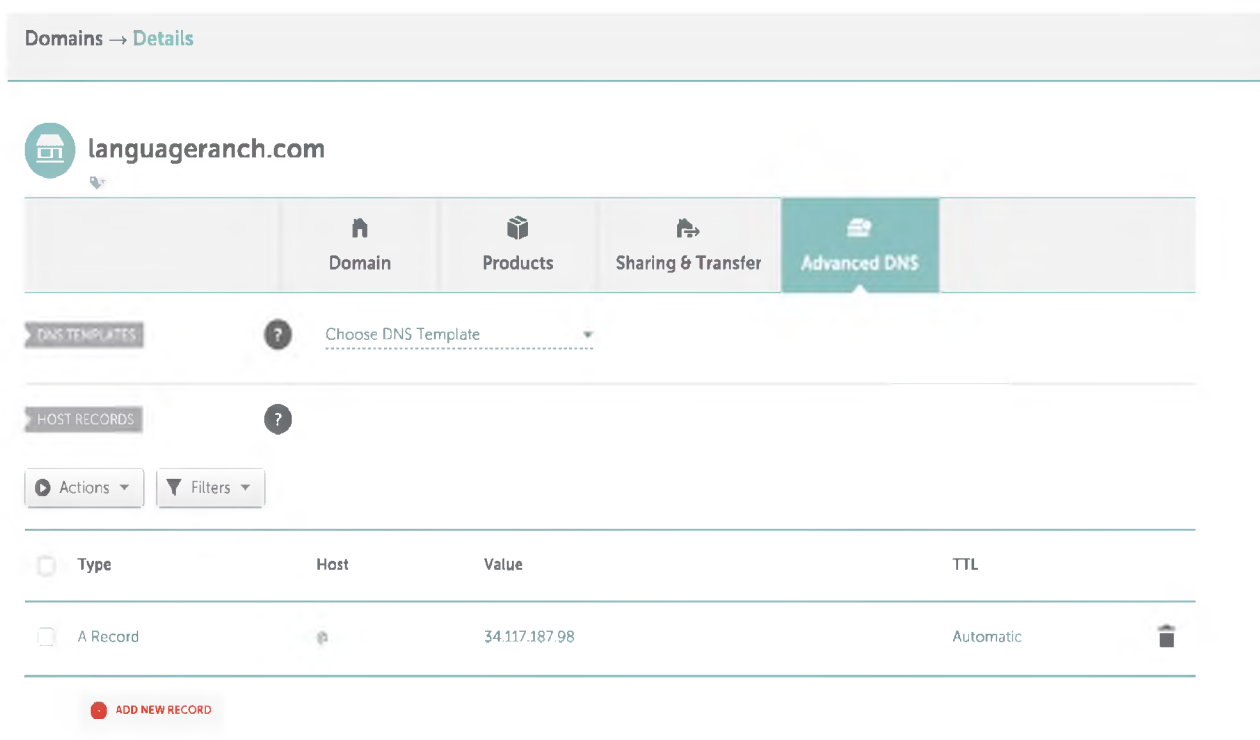


Рисунок 4.18 – Придбане доменне ім'я

Для підключення доменного ім'я до вебдодатку в сервісі Cloud Run було створено статичну IP адресу та налаштовано відповідним чином Cloud Load Balancer. Після цих налаштувань вебсайт став доступним за своїм доменним ім'ям, це продемонстровано на рисунку 4.19.

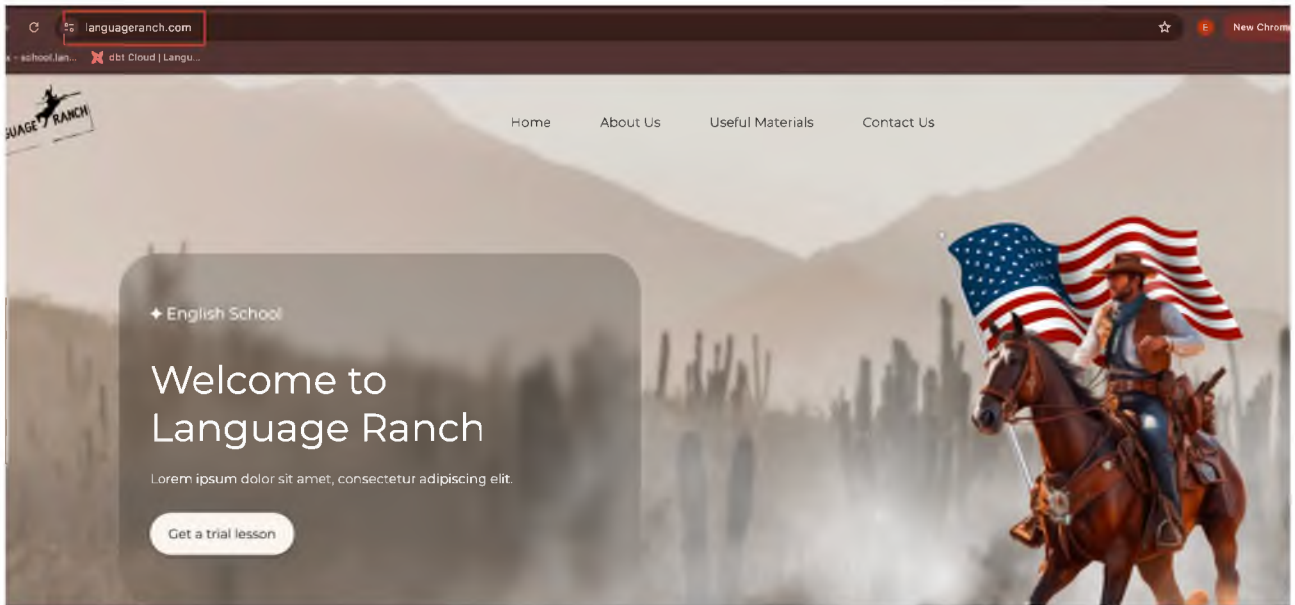


Рисунок 4.19 – Вебдодаток публічно доступний за доменним ім'ям

4.3 Побудова та розгортання системи аналізу даних

Центральною точкою входу в конвеєр даних є шина повідомлень Google Pub/Sub. Щоб забезпечити безперервний потік даних, було створено відповідну тему pub/sub під назвою «user_events».

Ця тема служить місцем, куди публікуються події користувача та згодом використовуються для подальшої обробки та аналізу. Завдяки Pub/Sub систему аналізу даних можна вважати надійною, бо застосування цього сервісу дозволяє бути впевненим, що дані не будуть втрачені, у випадку системного збою чи помилки на стороні споживача даних із цієї теми.

В Google Pub/Sub можна налаштувати гнучкий період утримання даних, що дозволяє утримувати дані певний період і зробити потворну обробку, якщо, наприклад, сервіс, що є споживачем даних, буде недоступний певний час.

На рисунку 4.20 можна побачити створену тему user_events.

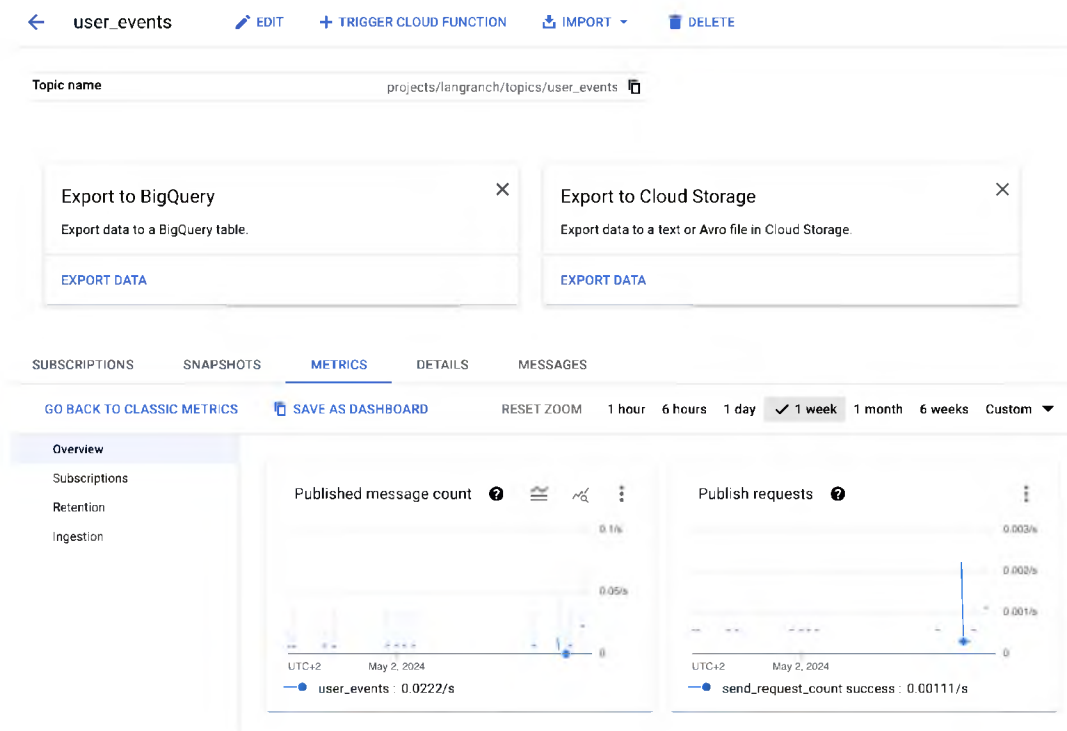


Рисунок 4.20 – Створена Pub/Sub тема user_events

Щоб ефективно обробляти події користувача з теми Pub/Sub, було розроблено хмарну функцію під назвою "events_consumer". Цей сервіс є підписаний на тему Pub/Sub "user_events", що забезпечує безперервне отримання вхідних подій.

У своїй поточній реалізації сервіс "events_consumer" отримує кожну подію з теми та записує її у відповідну таблицю в нашому сховищі даних BigQuery. Код для цієї хмарної функції написаний за допомогою мови програмування Python.

Цей процес полегшує обробку та зберігання даних у реальному часі, дозволяючи нам отримувати корисну інформацію під час взаємодії користувачів із нашою платформою.

На рисунку 4.21 зображено структуру проекту обробника подій та скрипт, який використовується для вивантаження в сервіс Google Cloud Functions.

```

EXPLORER
LANGRANCH-EV...
bin
$ deploy
utils
__init__.py
bq.py
helpers.py
venv
.env.production.yaml
.flake8
.gcloudignore
.gitignore
.python-version
const.py
main.py
README.md
requirements.txt

bin > $ deploy
1 #!/bin/bash
2
3 gcloud functions deploy events_consumer \
4   --no-gen2 \
5   --runtime=python310 \
6   --trigger-topic=user_events \
7   --region=europe-west1 \
8   --env-vars-file=.env.production.yaml \
9   --max-instances=100 \
10  --memory=128MB \
11  --timeout=30s \
12  --service-account=events-processor-bot@langranch.iam.gserviceaccount.com \
13

```

Рисунок 4.21 – Структура проекту хмарної функції `events_consumer`

На рисунку 4.22 продемонстровано вивантажену хмарну функцію `events_consumer` на платформі Google Cloud.

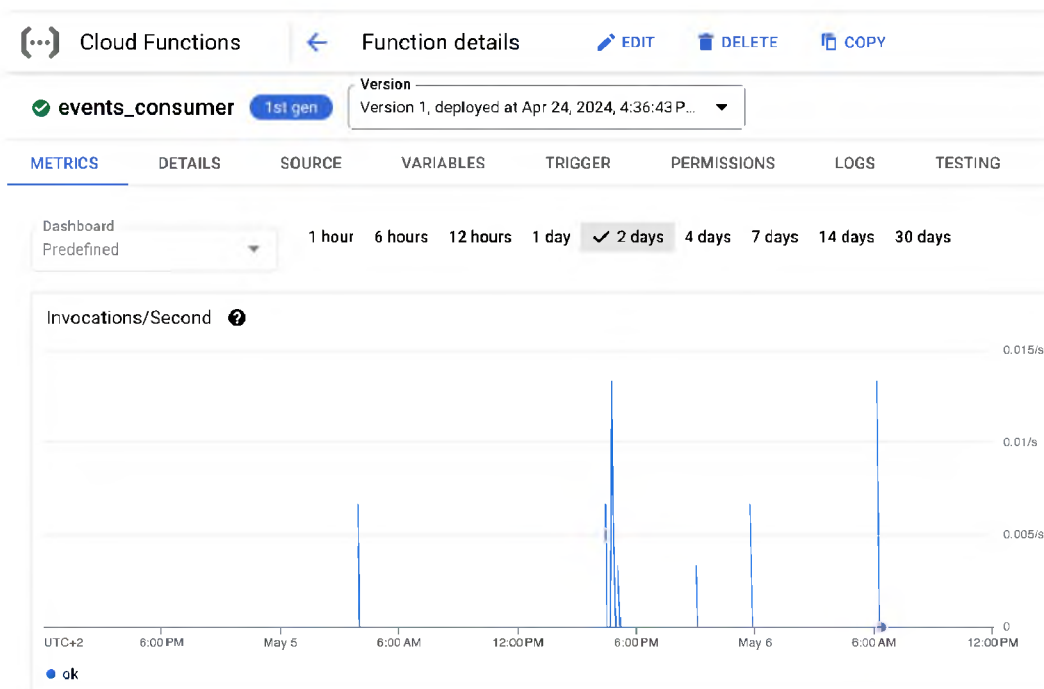


Рисунок 4.22 – Вивантажена хмарна функцію `events_consumer`

Хмарна функція `events_consumer` записує дані у відповідну таблицю в сховище даних. Для цього було включено BigQuery та створено відповідний

датасет під назвою "raw_data" для зберігання сирих даних. У датасеті було створено таблицю «user_events» для зберігання оброблених подій користувача. Ця конфігурація гарантує, що дані, отримані з теми Pub/Sub, надійно зберігаються для подальшої обробки в сховищі даних BigQuery, закладаючи основу для подальшого аналізу. На рисунку 4.23 зображено інформацію про створену таблицю user_events.

The screenshot displays the BigQuery interface for the table `user_events`. The left sidebar shows the project structure with `raw_data` selected. The main area shows the table details:

Table Properties:

Table Type	Partitioned
Partitioned by	DAY
Partitioned on field	timestamp
Partition expiration	Partitions do not expire
Partition filter	Not required

Storage info:

Number of rows	240
Number of partitions	12
Total logical bytes	94.87 KB
Active logical bytes	94.87 KB
Long term logical bytes	0 B
Total physical bytes	113.21 KB
Active physical bytes	113.21 KB
Long term physical bytes	0 B
Time travel physical bytes	0 B

Summary:

Last modified	May 6, 2024, 7:06:31 AM UTC+2
Data location	EU

Рисунок 4.23 – Таблиця user_events в BigQuery

На рисунку 4.24 продемонстровано приклад даних, які вже були завантажені в таблицю user_events на момент написання.

This is a partitioned table. [Learn more](#)

Row	event_id	insert_id	event_name	timestamp	properties	metadata.session_id	metadata
1	925820ba-9915-4f8a-a61e-235...	458f02c042016a36a4801e65 04c33363fb20a8100bb83205 04af9d50c6b19a7e	page_view	2024-04-23 19:39:22.446000 U...	{'path_changes_count': 0, 'path...	f5fbb21e-7242-4b4a-a06f-6beb...	188.157.2
2	faa905b3-1d23-491d-b0d4-886...	762332358de4706081e2e61d c29e77f9121b234be263ac00 6f8f13fe246c81	session_start	2024-04-23 20:16:24.534000 U...	{}	52222269-bc79-4043-9667-b3...	35.203.24
3	814727dc-7d95-4e3e-8fd5-218...	d107251c1aaa932079487e43 2db7361e5891cfa16cab9c37 cb29be9e710846a	page_view	2024-04-23 20:16:25.130000 U...	{'path_changes_count': 0, 'path...	52222269-bc79-4043-9667-b3...	35.203.24

Рисунок 4.24 – Приклад даних завантажених в таблицю user_events

Наступним кроком стала інтеграція DBT (Data Build Tool) у систему аналізу даних. Це передбачало створення DBT проєкту "langranch-data-models", розробленого для використання потужностей BigQuery для трансформації сирих даних в цінну інформацію. На рисунку 4.25 зображено GitHub репозиторій із моделями даних.

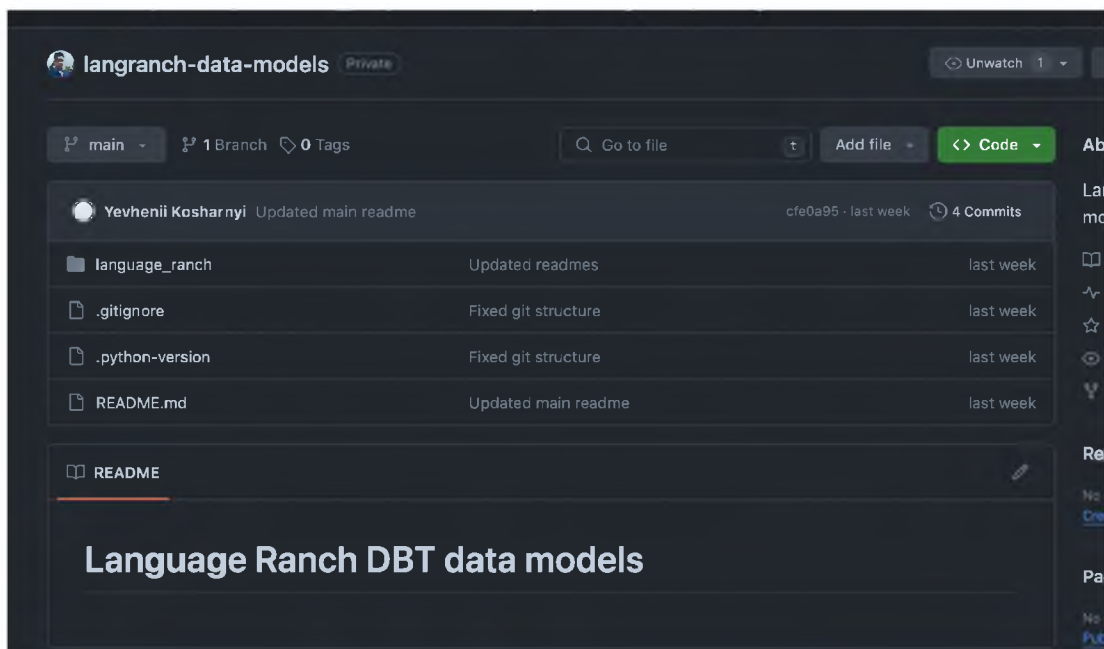


Рисунок 4.25 – GitHub репозиторій langranch-data-models

DBT дозволяє підключатись до сховища даних і за допомогою SQL описувати трансформації над даними. Таким чином було створено три моделі

даних для отримання відповідної інформації на основі подій користувача. Метою було зробити доступними для візуалізації дані про кількість унікальних сесій на веб сторінці, завантаження сторінок та запити на пробні заняття. На рисунку 4.26 продемонстровано приклад базових трансформацій над даними про сесії користувачів.

```

language_ranch > models > base > base_sessions.sql > ...
  Run | New Tab | JSON | Copy | Active Connection
1  WITH base_table AS (
2      SELECT
3          event_id,
4          metadata.session_id AS session_id,
5          timestamp,
6          metadata.ip_address AS ip_address,
7          metadata.country AS country,
8          metadata.city AS city,
9          metadata.is_bot AS is_bot,
10         metadata.os AS os,
11         metadata.device_type AS device_type,
12         metadata.device_vendor AS device_vendor,
13         metadata.device_model AS device_model
14     FROM {{ source('raw_data', 'user_events') }}
15     WHERE event_name = 'session_start'
16 ),
17
18     filtered_sessions AS (
19         SELECT * FROM base_table where session_id IS NOT NULL
20     )
21
22     SELECT * FROM filtered_sessions
  
```

Рисунок 4.26 – DBT модель base_sessions.sql

В цій DBT моделі дістаються строки та відповідні поля із подій з назвою “session_start”. Далі відфільтровуються дані, де поле session_id є NULL, бо такі дані вважаються не валідними.

У файлі dbt_project.yml є можливість конфігурувати всі моделі, наприклад вказати відповідну схему для зберігання таблиць, а також вказати способи матеріалізації. На рисунку 4.27 зображено таку конфігурацію.

```

models:
  language_ranch:
    # Config indicated by + and applies to all files under models/base/
    base:
      +materialized: view
      schema: "staging"
      tags: ["base"]
    core:
      +materialized: table
      tags: ["core"]

```

Рисунок 4.27 – Конфігурація DBT моделей

Після створення DBT моделей було налаштовано запуск трансформацій на платформі DBT cloud. На цій платформі можна гнучко налаштувати запуск трансформацій та вказати розклад для запусків. Було створено відповідний проєкт, налаштовано підключення до BigQuery та налаштовано і протестовано запуск DBT моделей. Безкоштовної версії DBT cloud цілком достатньо для невеликих підприємств та навчальних цілей. При бажанні від DBT cloud можна відмовитись і налаштувати запуск моделей за допомогою GitHub workflows, локально або в інших сервісах.

На рисунку 4.28 зображено налаштований запуск DBT моделей.

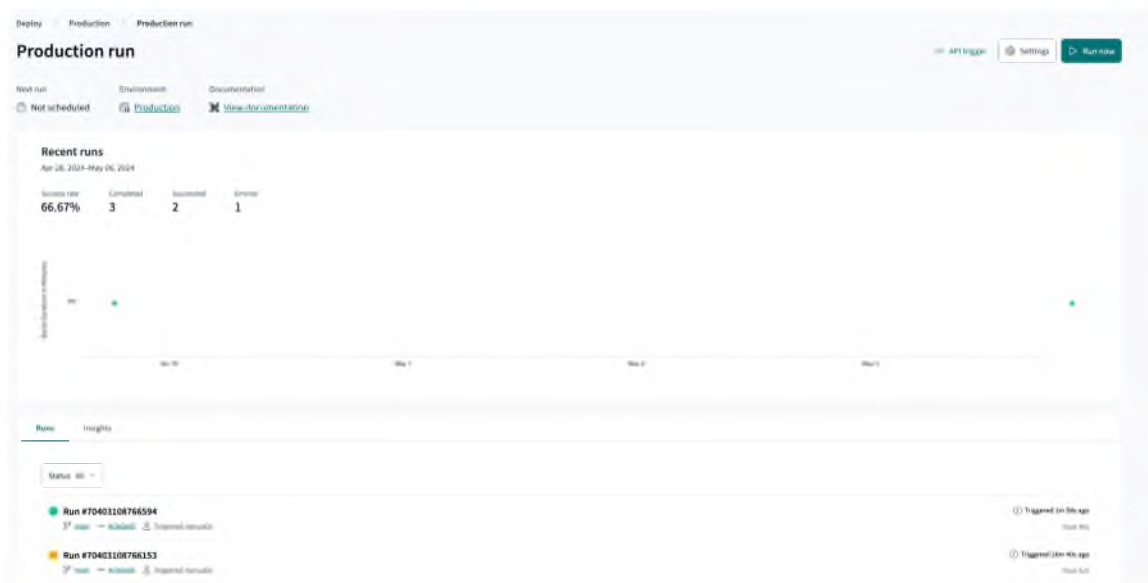


Рисунок 4.28 – Production run DBT моделей

Також DBT cloud пропонує зручний спосіб хостингу документації проєкту DBT, що зробить її доступною для всіх членів команди без необхідності генерувати її локально. На рисунку 4.29 продемонстровано граф залежностей між моделями, що є частиною документації DBT проєкту.

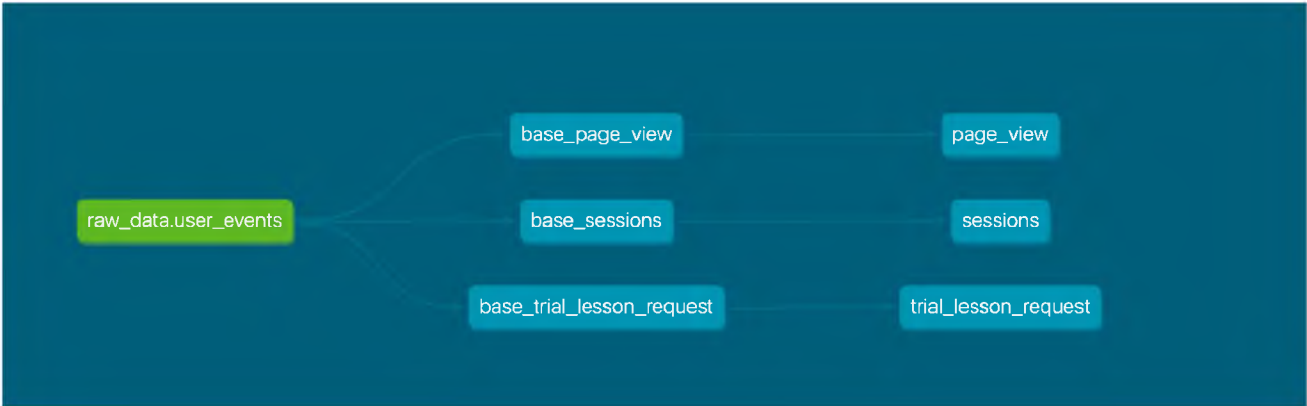


Рисунок 4.29 – Граф залежностей між DBT моделями

Після успішного запуску DBT моделей відповідні таблиці було створені в BigQuery. На рисунку 4.30 зображено створені таблиці та результат запиту до таблиці page_view.

Row	event_id	session_id	timestamp	path_name	path_changes_count
1	58c6cct6-25e4-44f9-8763-3f91...	5072efda-aect-4260-8958-055...	2024-05-06 04:05:25.461000 U...	/	0
2	45ac06e6-497c-44f2-9905-13e...	9f80fab6-6771-41fe-9b32-26c8...	2024-05-06 04:05:21.249000 U...	/	0
3	b30a7c66-3f92-4233-91f4-7b9...	01d392fd-9fa8-41eb-9f9c-edf4...	2024-05-05 18:56:55.089000 U...	/	0
4	950e4275-0105-4c65-9671-80...	fbfbb21e-7242-4b4a-a06f-6beb...	2024-05-05 15:01:27.571000 U...	/	0
5	e8d11834-1a8f-49eb-9769-40a...	fbfbb21e-7242-4b4a-a06f-6beb...	2024-05-05 14:45:06.897000 U...	/platform	0
6	20346184-4e4e-4f34-bb50-c55...	fbfbb21e-7242-4b4a-a06f-6beb...	2024-05-05 14:44:54.324000 U...	/platform	0
7	8ce9a495-1df7-4b77-a6ae-5d8...	fbfbb21e-7242-4b4a-a06f-6beb...	2024-05-05 14:44:51.362000 U...	/	0
8	5283b0fc-7a6f-4301-aaaa-08a...	fbfbb21e-7242-4b4a-a06f-6beb...	2024-05-05 14:44:47.259000 U...	/platform	0
9	d32c644c-32d7-4cdb-b82d-76c...	fbfbb21e-7242-4b4a-a06f-6beb...	2024-05-05 14:44:28.425000 U...	/platform	0
10	1e7c2c19-53ec-471f-8fff-ef0f6...	fbfbb21e-7242-4b4a-a06f-6beb...	2024-05-05 14:44:01.463000 U...	/	0
11	rtheF4R-R-249f-441e-nPa9-9ca...	fbfbb21e-7242-4b4a-a06f-6beb...	2024-05-05 14:43:33.477000 U...	/	0

Рисунок 4.30 – Результат запиту до таблиці page_view

Останнім етапом є візуалізація даних та отримання інформації про активність користувачів. Для цього було використано Google Looker Studio. На рисунку 4.31 продемонстровано першу частину створеного звіту про відвідуваність вебсайту.



Рисунок 4.31 – Звіт про відвідуваність вебсайту, частина 1

Цей звіт демонструє, що в період із 23 квітня 2024 року по 6 травня 2024 вебдодаток було завантажено 137 разів, серед цих візитів 11 разів вебдодаток відвідували пошукові боти. Загалом було створено 101 сесію, що приблизно може свідчити про кількість унікальних відвідувачів. Також можна побачити розподіл завантажень вебсайту по датам.

На рисунку 3.32 зображено другу частину звіту, де є розподіл відвідувачів вебсторінки в залежності від виробника пристрою, за країною місцезнаходження, а також продемонстровано розподіл створених сесій за датами.



Рисунок 4.32 – Звіт про відвідуваність вебсайту, частина 2

На рисунку 3.33 зображено фінальну частину звіту, де є розподіл створених сесій в залежності від виробника пристрою, за країною місцезнаходження, а також продемонстровано розподіл створених сесій за датами.



Рисунок 4.33 – Звіт про відвідуваність вебсайту, фінальна частина

4.4 Покриття компонентів системи тестами

Тестування відіграє важливу роль у розробці програмного забезпечення, гарантуючи, що система працює належним чином і відповідає очікуванням користувачів. Одним із ключових аспектів тестування є модульне тестування, коли окремі частини коду перевіряються, щоб переконатися, що вони працюють правильно. Наприклад, якщо у нас є функція, яка обчислює загальну вартість товарів у кошику для покупок, ми повинні написати модульний тест, щоб переконатися, що він дає правильний результат для різних наборів товарів.

Інтеграційне тестування перевіряє, як різні частини системи працюють разом. Це гарантує безперебійну взаємодію компонентів і отримання очікуваних результатів у поєднанні. Наприклад, можна перевірити, як модуль обробки платежів взаємодіє з системою управління продуктами, щоб переконатися, що замовлення обробляються правильно.

End-to-end тестування оцінює всю систему від початку до кінця, імітуючи реальні взаємодії користувача. Цей тип тестування перевіряє, чи всі компоненти працюють разом, як очікувалося, і чи правильно поводить система в різних сценаріях. Наприклад, end-to-end тестування може включати імітацію користувача, який переглядає вебсайт, додає товари в кошик і завершує процес оформлення замовлення, щоб переконатися, що все працює як очікується.

Загалом, тестування має важливе значення для виявлення та виправлення помилок, забезпечення якості програмного забезпечення та формування впевненості в надійності та продуктивності системи. Ретельно протестувавши кожен компонент системи, розробники можуть створити стабільний і надійний програмний продукт, який відповідає потребам і очікуванням користувачів.

На рисунку 4.34 продемонстровано приклад написаного модульного тесту, що перевіряє правильність виконання функції `getMovingListSubset`, яка використовується для реалізації каруселі зображень або карток на вебсторінці.

```

JS listUtils.test.js x
src > lib > utils > JS listUtils.test.js > ...
1  const { getMovingListSubset } = require("../listUtils");
2
3  describe("getMovingListSubset function", () => {
4    const parentList = [1, 2, 3, 4, 5];
5
6    test("returns subset when referenceIndex + subsetItemsCount is within parentList length", () => {
7      expect(getMovingListSubset(parentList, 0, 3)).toEqual([1, 2, 3]);
8      expect(getMovingListSubset(parentList, 2, 2)).toEqual([3, 4]);
9      expect(getMovingListSubset(parentList, 1, 4)).toEqual([2, 3, 4, 5]);
10   });
11
12   test("throws error when subsetItemsCount is greater than parentList length", () => {
13     expect(() => {
14       getMovingListSubset(parentList, 0, 6);
15     }).toThrowError(
16       "subsetItemsCount must be less than parentList length. Received: 5 < 6"
17     );
18   });
19
20   test("returns subset with wrapped elements when referenceIndex + subsetItemsCount exceeds parentList length",
21     expect(getMovingListSubset(parentList, 4, 3)).toEqual([5, 1, 2]);
22     expect(getMovingListSubset(parentList, 3, 4)).toEqual([4, 5, 1, 2]);
23     expect(getMovingListSubset(parentList, 2, 5)).toEqual([3, 4, 5, 1, 2]);
24   });
25 });
26

```

Рисунок 4.34 – Тест функції getMovingListSubset

Також тестувати можна і дані, які вже завантаженні у сховище даних. Для цього є багато інструментів, але так як у проекті використовується DBT, то для цього було використано його функціонал. На рисунку 4.35 продемонстровано тести якості даних, які перевіряють що дані у відповідних колонках мають бути унікальними і не мати порожніх значень.

```

language_ranch > models > base > ! base.yml
1
2  version: 2
3
4  models:
5    - name: base_page_view
6      description: "Base page_view model"
7      columns:
8        - name: event_id
9          description: "The primary for this table."
10         tests:
11           - unique
12           - not_null
13         - name: session_id
14           description: "Session value. It is required to identify events generated by one user."
15           tests:
16             - not_null
17

```

Рисунок 4.35 – Тестування якості даних у DBT проекті

ВИСНОВКИ

В ході проходження передатестаційної практики було виконано розробку системи для аналізу даних з платформи для вивчення іноземних мов і було створено відповідний програмний продукт. В процесі створення інформаційної системи було виконано наступні етапи:

- аналіз предметної області, ситуації на ринку та конкурентів;
- формування вимог до інформаційної системи;
- визначення основних функціональних та нефункціональних вимог до вебдодатку та системи аналізу даних;
- визначення функцій інтерфейсу клієнтської частини вебсайту;
- створення дизайну вебдодатку;
- прогнозування сценаріїв взаємодії користувачів із вебдодатком;
- категоризація користувачів і визначення їхніх дозволів і привілеїв у системі;
- проектування архітектури системи аналізу даних та вебдодатку;
- дослідження та вибір найбільш прийнятних технологій для розробки інформаційної системи;
- визначення сутностей і зв'язків у системі та проектування операційної бази даних;
- визначення подій користувача для відправки в систему аналізу даних;
- програмна реалізація вебдодатку;
- розгортання вебдодатку;
- побудова та розгортання системи аналізу даних;
- покриття компонентів системи тестами

Вимоги для початкової версії системи виконані вчасно. Також вже визначено план на подальше вдосконалення системи. По-перше, потрібно фіналізувати landing сторінку вебдодатку, заповнивши її справжньою інформацією та корисними матеріалами. Також цей пункт включатиме в себе інтернаціоналізацію вебсторінки, щоб вона могла підтримувати як мінімум українську та англійську

мови. По-друге, треба розпочати розробку компонентів платформи на базі існуючого вебдодатку для провадження навчального процесу, це включає в себе додавання процесу реєстрації та ролей користувачів, створення адміністративної панелі, можливість створювати курси та уроки вчителями, керувати учнями в рамках курсу та інше. У ході цього вдосконалення будуть з'являтися нові події користувачів, які мають бути інтегровані в систему аналізу даних.

Використання системи аналізу даних для платформи вивчення іноземних мов дозволить надати зацікавленим сторонам всебічне розуміння продуктивності бізнесу за допомогою візуалізацій проаналізованих даних, що стосуються різних ключових показників, пов'язаних із операційною діяльністю вебсайту мовної школи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. CEO, COO, CFO, CIO, CMO, CTO, Who is Who? // ZIGURAT GLOBAL INSTITUTE OF TECHNOLOGY SL. – 2018. – Режим доступу до ресурсу: <https://www.e-zigurat.com/en/blog/ceo-coo-cfo-cio-cmo-cto-who-is-who/>. (дата звернення: 26.03.2024).
2. Xia V. What is Mobile First Design? Why It's Important & How To Make It? / Vincent Xia // Medium. – 2017. – Режим доступу до ресурсу: <https://medium.com/@Vincentxia77/what-is-mobile-first-design-why-its-important-how-to-make-it-7d3cf2e29d00> (дата звернення: 27.03.2024).
3. Desktop vs Mobile vs Tablet Market Share Worldwide - February 2024 // StatCounter. – 2024. – Режим доступу до ресурсу: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/#monthly-200901-202402> (дата звернення: 28.03.2024).
4. Figma. // Wikipedia, Wikimedia Foundation. – 2024. – Режим доступу до ресурсу: en.wikipedia.org/wiki/Figma (дата звернення: 28.03.2024).
5. Use Case Diagram. // Wikipedia, Wikimedia Foundation. – 2024. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Use_case_diagram (дата звернення: 28.03.2024).
6. Terra J. What is Client-Server Architecture? Everything You Should Know / John Terra // Simplilearn Solutions. – 2023. – Режим доступу до ресурсу: <https://www.simplilearn.com/what-is-client-server-architecture-article#:~:text=The%20client-server%20architecture%20refers,model%20or%20client%20server%20network.> (дата звернення: 01.04.2024).
7. Extract, Transform, Load. // Wikipedia, Wikimedia Foundation. – 2024. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Extract,_transform,_load (дата звернення: 03.04.2024).
8. Data warehouse // Wikipedia, Wikimedia Foundation. – 2024. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Data_warehouse (дата звернення: 03.04.2024).

9. Кошарний Є. Ю., Розробка системи для аналізу даних платформи вивчення іноземних мов: 28-й міжнар. молодіж. форум зб. матеріалів Т. 6., м. Харків, 16-18 квіт. 2024 р. Харків, 2024. С. 612–614 (дата звернення: 05.04.2024).
10. Dataflow overview // Google Cloud. – 2024. – Режим доступу до ресурсу: <https://cloud.google.com/dataflow/docs/overview> (дата звернення: 05.04.2024).
11. Django documentation // Django Software Foundation. – 2024. – Режим доступу до ресурсу: <https://docs.djangoproject.com/en/5.0/> (дата звернення: 05.04.2024).
12. What is Next.js? // Vercel. – 2024. – Режим доступу до ресурсу: <https://nextjs.org/docs#what-is-nextjs> (дата звернення: 05.04.2024).
13. What is Cloud Run // Google Cloud. – 2024. – Режим доступу до ресурсу: <https://cloud.google.com/run/docs/overview/what-is-cloud-run> (дата звернення: 05.04.2024).
14. Cloud SQL overview // Google Cloud. – 2024. – Режим доступу до ресурсу: <https://cloud.google.com/sql/docs/introduction> (дата звернення: 05.04.2024).
15. PostgreSQL 16.3 Documentation // The PostgreSQL Global Development Group. – 2024. – Режим доступу до ресурсу: <https://www.postgresql.org/docs/16/index.html> (дата звернення: 05.04.2024).
16. What is Pub/Sub? // Google Cloud. – 2024. – Режим доступу до ресурсу: <https://cloud.google.com/pubsub/docs/overview> (дата звернення: 06.04.2024).
17. Cloud Functions overview // Google Cloud. – 2024. – Режим доступу до ресурсу: <https://cloud.google.com/functions/docs/concepts/overview> (дата звернення: 06.04.2024).
18. BigQuery overview // Google Cloud. – 2024. – Режим доступу до ресурсу: <https://cloud.google.com/bigquery/docs/introduction> (дата звернення: 06.04.2024).
19. Handy T. What, exactly, is dbt? / Tristan Handy // dbt Labs. – 2017. – Режим доступу до ресурсу: <https://www.getdbt.com/blog/what-exactly-is-dbt> (дата звернення: 06.04.2024).

20. Dbt cloud // dbt Labs. – 2024. – Режим доступа до ресурсу: <https://www.getdbt.com/product/dbt-cloud> (дата звернення: 06.04.2024).
21. Looker Product Family Documentation // Google Cloud. – 2024. – Режим доступа до ресурсу: <https://cloud.google.com/looker/docs> (дата звернення: 06.04.2024).
22. Entity–Relationship Model // Wikipedia, Wikimedia Foundation. – 2024. – Режим доступа до ресурсу: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model (дата звернення: 07.04.2024).
23. Next.js Project Structure // Vercel. – 2024. – Режим доступа до ресурсу: <https://nextjs.org/docs/getting-started/project-structure> (дата звернення: 07.04.2024).
24. CI/CD // Wikipedia, Wikimedia Foundation. – 2024. – Режим доступа до ресурсу: <https://de.wikipedia.org/wiki/CI/CD> (дата звернення: 08.04.2024).
25. Server Actions and Mutations // Vercel. – 2024. – Режим доступа до ресурсу: <https://nextjs.org/docs/app/building-your-application/data-fetching/server-actions-and-mutations> (дата звернення: 08.04.2024).