

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

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи

на ступінь вищої освіти бакалавр

**на тему: «РОЗРОБКА УКРАЇНОМОВНОГО WEB-ДОДАТКУ
ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ ЗА
ДОПОМОГОЮ REACT.JS»**

Виконав: студент 4 курсу, групи ПД-44

спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Двірний Д.Ю.

(прізвище та ініціали)

Керівник

Яскевич В.О.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Нормоконтроль

(прізвище та ініціали)

Київ – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

_____ О.В. Негоденко

« ____ » _____ 2021 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ
ДВІРНОМУ ДМИТРУ ЮРІЙОВИЧУ

1. Тема роботи: «Розробка україномовного web-додатку для вивчення англійської мови за допомогою React.js»

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

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.1. Середовище розробки Visual Code

3.2. Структура веб додатку.

3.3. Науково-технічна література, пов'язана з розробкою.

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

4.1. Аналіз області досліджень

4.2. Вибір технологій і середовища

4.3. Розробка платформи

5. Перелік графічного матеріалу

5.1.1. Популярність платформ для дистанційного навчання

5.1.2. Переваги та недоліки найвідоміших платформ для дистанційного навчання

5.1.3. Програмні засоби реалізації

5.1.4. Огляд можливостей

5.1.5. Апробація результатів досліджень

6. Дата видачі завдання 19.04.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.21 – 22.04.21	
2	Аналіз існуючих аналогів	22.04.21 – 23.04.21	
3	Дослідження програмних засобів	23.04.21 – 01.05.21	
4	Моделювання об'єкту	01.05.21 – 05.05.21	
5	Розробка функціоналу веб додатку	05.05.21 – 07.05.21	
6	Вступ, висновки, реферат	07.05.21 – 08.05.21	
7	Розробка презентації застосунку	08.05.21 – 24.05.21	
8	Попередній захист роботи	25.05.21	

Студент

Керівник роботи

РЕФЕРАТ

Текстова частина бакалаврської роботи 46 с., 36 рис., 19 джерел.

Ключові слова: україномовна, англійської, дистанційне, платформа, вивчення, мови, додаток

Об'єкт дослідження – вивчення англійської мови.

Предмет дослідження – україномовна платформа для вивчення англійської, розроблена на основі React.JS.

Мета роботи – покращення процесу вивчення англійської мови за допомогою україномовного додатку на основі React.JS.

Методи дослідження – Теоретичні методи класифікації, аналізу. Емпіричні методи.

Наукова новизна даної роботи полягає в наступному:

В роботі виконано аналіз існуючих аналогів платформ для дистанційного навчання. Встановлено їх переваги та недоліки. В результаті аналізу було визначено основні проблеми додатків аналогів та потреби користувачів. Проаналізовано можливості мов програмування PHP та JavaScript, її фреймвока ReactJS, середовища розробки VisualCode, СУБД MySQL, додатка phpMyAdmin.

Розроблено логіку, алгоритм та базовий функціонал додатку, описано чому до нього були поставлені саме такі завдання.

Галузь використання – завдяки тому, що платформа для дистанційного навчання має не комерційну основу, вона може широко застосовуватися для навчання в школах, для вивчення англійської мови для дітей та людей старшого віку в поза робочій час.

Зміст

1 АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕНЬ.....	9
1.1 Платформи для дистанційного навчання.....	9
1.2 Огляд аналогів	10
1.2.1 Busuu.....	10
1.2.2 Duolingo.....	11
1.2.3 LinguaLeo	13
1.3 Формування вимог до додатку.....	14
Висновки до першого розділу.....	14
2 ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ	15
2.1 Обрані технології розробки.....	15
2.1.1 JavaScript	15
2.1.2 REACT.JS	17
2.1.3 MySQL.....	20
2.1.4 PhpMyAdmin	23
2.1.5 HTML.....	24
2.1.6 CSS.....	25
2.1.7 PHP.....	26
2.2 Середовище розробки	26
2.2.1 Visual Code	26
2.3 Проектування додатку	27
3.1 Загальні відомості	29
3.2 Серверна частина.....	30
3.3 База даних.....	34

3.4 Клієнська частина.....	38
3.5 Тестування.....	42
3.5.1 Інтерфейсу користувача.....	42
3.5.2 Функціональне тестування.....	43
Висновки до третього розділу.....	43
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	45
ДОДАТОК А.....	47
ДОДАТОК Б	49

ВСТУП

Протягом всієї історії людства знання були одним із надважливих елементів життя людини. І чим далі ми будемо рухатись по історичній прямій тим більшу роль будуть відігравати знання в житті людини. Саме тому, розробка навчальних додатків, являється одним із самих важливих завдань в ІТ.

Сьогодні в Україні можна помітити значний попит на вивчення англійської мови. Цей попит можна пояснити доволі природними причинами. Англійська є мовою міжнародного спілкування, тому її вивчення доволі актуальне для тих, хто збирається з туристичними цілями їхати за кордон.

Також вивчення англійської є актуальним для спілкування з іноземцями, адже в сучасному світі, можна не обмежувати своє спілкування, лише людьми з якими ти живеш в одній країні.

Англійська є основною мовою комунікації в ІТ індустрії, а вона одна із лідерів по темпу розвитку в Україні та всьому світі. Люди в індустрії віддають перевагу комунікації на англійській мові, провідні мови програмування використовують в своїй основі англійську мову, тому без знання англійської в цій індустрії - нікуди.

Також в Україні росте попит на дистанційне навчання. Завдяки досить серйозним перевагам такого виду навчання. Наприклад можливість отримувати знання в будь-який час доби є актуальним для дорослих людей, які можуть бути зайняті в часи очного навчання. Також, для нового покоління, отримання знань через інтернет вже являється звичною справою, і для них дистанційне навчання може бути набагато простіше і зрозуміліше ніж очне.

В українського ринку платформ для дистанційного навчання є декілька важливих проблем.

По-перше, недостатня кількість платформ, які підтримують навчання на українській мові.

По-друге, більшість існуючих платформ доволі погано працюють з повільним інтернетом.

Навчання на українській мові є дуже актуальною проблемою для України, адже попит на нього росте не пропорційно збільшенню можливостей для цього.

В Україні, на жаль, досі актуальна проблема повільного інтернет-з'єднання та застарілих комп'ютерів, особливо за межами великих міст, де ще не всюди є 4G, або швидкий інтернет від провайдера.

Веб-додаток, на основі фреймворку мови програмування JavaScript від назвою ReactJS, також дозволяє вирішити проблему повільного інтернет-з'єднання, оскільки однією з переваг ReactJS являється його «легкість», що несумнівно допоможе користувачам з повільним інтернет з'єднанням.

Метою випускної кваліфікаційної роботи є розробка україномовного веб додатку для вивчення англійської мови на основі мови програмування JavaScript та її фреймворку ReactJS.

Були поставлені наступні завдання, базуючись на встановленій меті:

- аналіз обраної предметної області;
- порівняння наявних аналогів платформ для дистанційного навчання;
- вибір технологій і середовища розробки;
- розробка веб додатку.

1 АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕНЬ

1.1 Платформи для дистанційного навчання

Платформи для дистанційного навчання це досить перспективна галузь для розвитку інформаційних технологій. Вони мають високу актуальність навчання людей старшого віку.

На жаль, на даний момент, існує дефіцит україномовних платформ для дистанційного навчання. Також існуючі платформи вимагають встановлення різноманітних додатків на пристрій користувача або потребують швидкого інтернет з'єднання, чого мовах іноді застарілих пристроїв у користувачів та школах та повільного інтернет – з'єднання може створювати деякі проблеми.

Також велика частина існуючих послуг дистанційного навчання має платну основу або вбудовану рекламу, тому розробка україномовної безкоштовної платформи для дистанційного навчання є досить актуальною.

Платформа буде реалізована у формі веб додатку. Було обрано саме таку форму, через наступні конкурентні переваги: завдяки тому що веб додатки не потребують завантаження на пристрій користувача, та тому що мережа інтернет зараз доступна практично всюди, буде досягнена максимальна доступність платформи для користувачів. Також ж веб додатків немає обмежень щодо технічних характеристик пристрою користувача.

В Україні можна помітити попит, на підвищення кількості громадян, які хочуть вчитися саме українською мовою. На рисунку 1.1 можна помітити цей тренд.

Рік	1955	1986	1991	1992	1994	1996	1998	2005	2006	2007	2011	2012	2017	2018
Відсоток української мови	72,8%	40,6%	49,3%	51,4%	56,5%	60,0%	65,0%	77,0%	78,0%	79,0%	82,2%	81,9	90,0%	92,0%

Рисунок 1.1 – дані, щодо росту популярності навчання англійською.

Верхня шкала – рік, за який взяті данні.

Нижня шкала – відсоток людей, які навчаються українською мовою [1].

Одним із переваг дистанційного методу навчання є те що такий вид навчання значно дешевший та набагато більш зручний, завдяки тому, що людині необхідно навчатися самій це розвиває в ній самостійність та само організованість.

У суспільстві існує великий запит на платформи для дистанційного навчання, завдяки масовому попиту на дистанційну освіту серед людей старшого віку. Сучасне суспільство вимагає безупинного підвищення власної кваліфікації, в чому, без заперечного, можуть допомогти платформи для дистанційної освіти людей.

Розробка веб-додатку, який буде враховувати це та, відповідно, буде розроблений так, щоб усунути вище названі проблеми та відповісти на описані потреби є актуальною.

1.2 Огляд аналогів

1.2.1 Busuu

Першим аналогом для розгляду було обрано додаток Busuu. Компанія є одним із лідерів на ринку платформ для дистанційного навчання.

Має приємний інтерфейс, велике різноманіття мов для вивчення. Завдяки тому що вона існує з 2008 року, в неї є велика база завдань перевірених часом.

На рисунку 1.2 зображено інтерфейс Busuu.

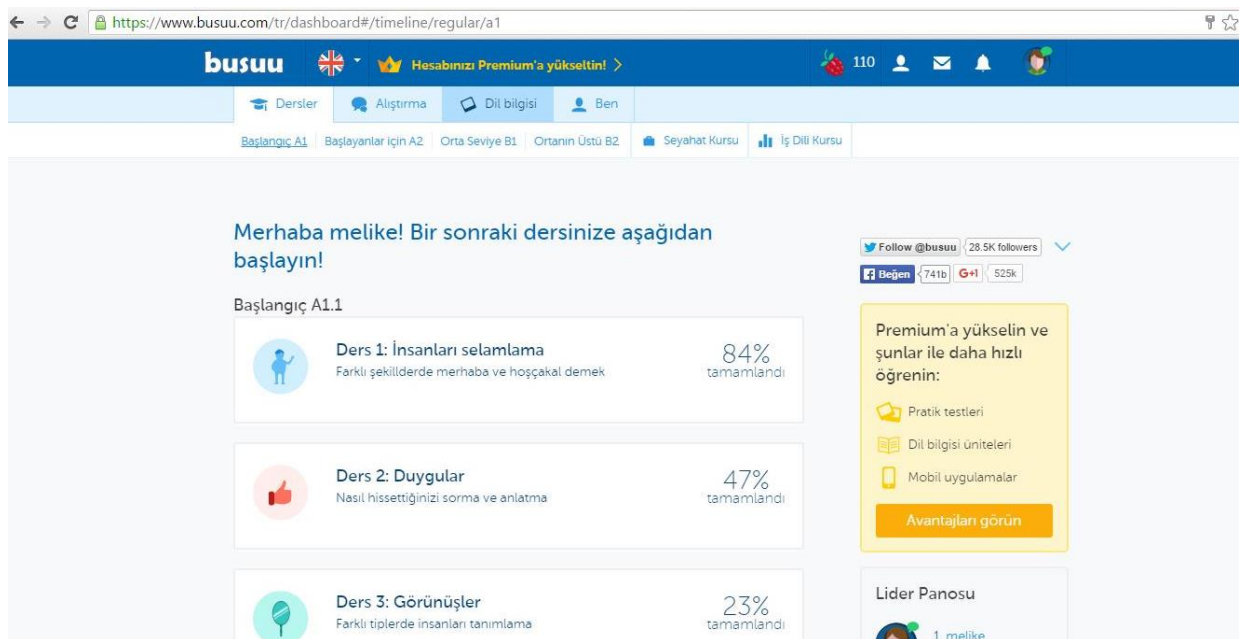


Рисунок 1.2 – Інтерфейс Busuu [2]

Головним недоліком цього додатку було визначено відсутність підтримки української мови.

Також мінусом платформи є присутність платних уроків, більшість матеріалів

Таблиця 1 – Переваги та недоліки платформи Busuu

Переваги	Недоліки
Приємний інтерфейс	Відсутність підтримки української мови
Декілька мов для вивчення	Наявність платних завдань
Перевірена часом база завдань	

1.2.2 Duolingo

Перша бета версія додатку **Duolingo** вийшла 30 листопада 2011 року. Сервіс є одним із лідерів на ринку.

Навчання на платформі проводиться лише в ігровій формі, що може бути недоліком для дорослих користувачів.

На рисунку 1.3 зображено інтерфейс Duolingo.



Рисунок 1.3 – інтерфейс Duolingo [3]

Сервіс має платні функції, що може становити проблему для використання в школах.

Також у платформи досить багато графічних зображень, що може становити проблему для використання при повільному інтернет – з'єднання або на слабкому за технічними характеристиками пристрої.

Позитивними сторонами сервісу можна визначити велику та перевірену базу завдань та широкий вибір мов.

Таблиця 2 – Переваги та недоліки платформи Duolingo

Переваги	Недоліки
Широкий вибір мов	Завантаженість графічними зображеннями
Велика база завдань	Лише ігрова форма навчання
	Наявність платних завдань

1.2.3 LinguaLeo

LinguaLeo – платформа була заснована в 2013 році. Має ігрову форму навчання, що може бути недоліком для деяких користувачів.

Додаток локалізована українською мовою.

Доступний, як в веб режимі так і у вигляді додатку на телефон. Є однією з найбільших платформ для дистанційного навчання.

На рисунку 1.4 зображено інтерфейс LinguaLeo.

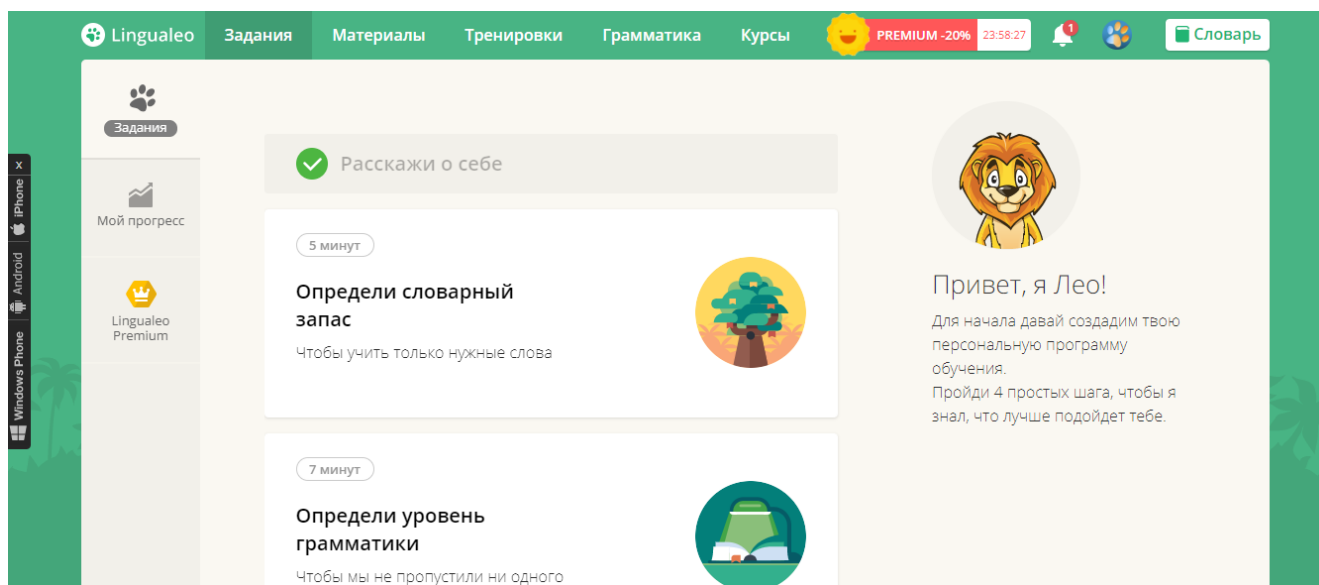


Рисунок 1.4– Інтерфейс LinguaLeo [4]

Недоліком можна вважати те, що деякі матеріали доступні лише у формі відео, що може становити проблему для користувачів з повільним інтернет – з'єднанням або обмеженим трафіком.

Також більшість матеріалів платформи недоступні у безкоштовній версії додатку, що може бути недоліком для деяких користувачів.

Через особливості своєї форми навчання у платформи доволі графічного перевантажений інтерфейс, це може бути неприємно деяким користувачам.

Таблиця 3 – Переваги та недоліки платформи LinguaLeo

Переваги	Недоліки
Повна локалізація українською	Деякі матеріали доступні лише у формі відео
Доступність у вигляді веб додатка та у вигляді додатка для Android і IOS.	Наявність платних завдань
	Перевантажений інтерфейс

1.3 Формування вимог до додатку

Виходячи з результатів аналізу додатків аналогів можна сформулювати наступні вимоги для розроблювального додатку:

1. Додаток повинен підтримувати українську мову, на ринку гостро спостерігається дефіцит україномовних платформ для дистанційного навчання;
2. Додаток повинен мати не комерційну основу, адже всі розглянуті аналоги мали платні функції, що створює дефіцит додатків для людей, що не готові або не можуть платити за них;
3. Додаток повинен мати простий і графічний інтерфейс, по-перше, графічно перевантажений інтерфейс створює проблеми для людей з повільним інтернет з'єднанням та застарілими пристроями, по-друге, занадто перевантажений інтерфейс може бути неприємним деяким користувачам.

Висновки до першого розділу

В першому розділі були розглянуті наступні теми: що таке дистанційна освіта та платформи для дистанційної освіти, існуючі аналоги платформ для дистанційної освіти їх недоліки та переваги.

На основі аналізу можна отримати такі результати. Дистанційна освіта та платформи для дистанційної освіти є досить перспективними сферами розвитку.

Жоден з розглянутих аналогів не мав таких переваг, як можливість навчання на українській мові та повністю безкоштовна основа одночасно. Саме тому розробка цієї платформи для дистанційного навчання є актуальною.

Отже, після проведення цих теоретичних досліджень, наступний розділ буде присвячено вибору технології середовища розробки.

2 ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ

2.1 Обрані технології розробки

2.1.1 JavaScript

JavaScript — це мова скриптів, яка використовується для розробки вбудованих додатків, які виконуються як на стороні клієнта, так і на стороні сервера [5].

На рисунку 2.1 зображено зразок коду JavaScript.

```
1  $(".move_block").swipe( {
2      swipeLeft:leftSwipe,
3      swipeRight:rightSwipe,
4      threshold:0
5  });
6  function leftSwipe(event){
7      console.log ("yes");
8  }
9  function rightSwipe(event){
10     alert('swipe right');
11 }
```

Рисунок 2.1 – Фрагмент коду JavaScript.

Зазвичай JavaScript влаштований в веб сторінку або завантажується з сервера файлом у форматі «.js».

Однією з особливостей мови програмування є те що вона працює на стороні клієнта. Тобто, скрипт завантажується на комп'ютер клієнта, а не виконується на сервері, що відрізняє його від інших мов програмування.

Таблиця 4 – Переваги та недоліки JavaScript [6]

Переваги JavaScript	Недоліки JavaScript
відносно не складно виявити і виправити помилки	фрагменти коду можуть бути великими
JavaScript працює на різних платформах та браузерах	на різних платформах може відображатися по різному
робота з ним не потребує компілятора	відсутня типізація даних.
він простий та швидкий	мова компілюється в момент виконання коду, що іноді збільшує час виконання програми

Додати код JavaScript на сторінку можна використовуючи тег «<script>». Зробити це можна двома способами:

1. Додаючи код в сам тег;

Спосіб підключення зображений на рисунку 2.2.

```

1  <script>
2      alert (1) ;
3  </script>
```

Рисунок 2.2 – Перший спосіб підключення JS.

2. Додаючи в тег посилання на файл «.js» з кодом.

Спосіб підключення зображений на рисунку 2.3.

```

1 <script src="/js/script1.js">
2 </script>

```

Рисунок 2.3 – Другий спосіб підключення JS.

2.1.2 REACT.JS

React.JS – це декларативна, ефективна і гнучка JavaScript – бібліотека, призначена для створення інтерфейсів користувача. Вона дозволяє компонувати складні інтерфейси з невеликих окремих частин коду — «компонентів» [7].

Переваги та недоліки React.JS у порівнянні з його найближчими конкурентами.

Розробники React.js піклуються про те щоб перехід на більш нову версію цієї бібліотеки JavaScript проходив досить легко, що значно спрощує підтримку додатку.

На рисунку 2.4 зображено фрагмент коду React.JS.

```

1 import React, { Component } from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4
5 class App extends Component {
6   render() {
7     return (
8       <div className="App">
9         <header className="App-header">
10          <img src={logo} className="App-logo" alt="logo" />
11          <p>
12            Edit <code>src/App.js</code> and save to reload.
13          </p>
14          <a
15            className="App-link"
16            href="https://reactjs.org"
17            target="_blank"
18            rel="noopener noreferrer"
19          >
20            Learn React
21          </a>
22        </header>
23      </div>
24    );
25  }
26 }
27
28 export default App;
29

```

Рисунок 2.4 – Фрагмент коду Reacts.JS

Також він не займає занадто багато інтернет трафіку, завдяки своїй простоті, що буде доречним при поганому інтернет – з'єднання. Також серед переваг ReactJS можна назвати підготовленість та адаптивність до високих навантажень, підтримку віртуального DOM, що сильно спрощує розробку, та має високу гнучкість, що буде доречним при вдосконаленні веб додатку.

Angular, був відсіяний через складну міграцію між версіями, що досить сильно ускладнило, підтримку веб додатку, та через свою складність, що може створити перепони під час розробки додатку. Ці причини є критичними в рішенні відмовитися від його використання, навіть не дивлячись на те що в Angular є декілька досить важливих конкурентних переваг, таких як, наприклад, мінімальний ризик отримати помилку [8].

На рисунку 2.5 зображено фрагмент коду Angular.

```
• app.ts
1 import {Component} from 'angular2/core'
2
3 @Component({
4   template: `<h1 [style]='font'></h1>
5   <div>{{person.address.street}}</div>
6   <span (click)='updatePerson("test")'></span>`
7 })
8 class Foo {
9   person = {
10    /** Full name */
11    name: "John Doe",
12    address: {
13      /** Street number and address */
14      street: "1 Happy St",
15      zip: 98118
16    }
17  };
18
19  font: string;
20  updatePerson(name: string){ /* TODO */}
21 }
22
```

Рисунок 2.5 – Фрагмент коду Angular

У Vue.js є наступні недоліки: менш функціональна система рендерингу, що означає менше простору та можливостей для налагодження проекту, менш гнучка шаблонізація у порівнянні з React.JS. Це досить важливо для більш – менш масштабного проекту, тому vue.js, більше підходить для менших проектів ніж той який викладені в цій роботі [9].

На рисунку 2.6 зображено фрагмент коду Vue.js

```
▼ Hello.vue x
amr.swalha.code, a few seconds ago | 1 author (amr.swalha.code)
1 <template>
2   <div class="hello">
3     <h1>{{ msg }}</h1>
4     <h2>{{ employee }}</h2>
5     <h2>Essential Links</h2>
6     <ul>
7       <li><a href="https://vuejs.org" target="_blank">Core Docs</a></li>
8       <li><a href="https://forum.vuejs.org" target="_blank">Forum</a></li>
9       <li><a href="https://gitter.im/vuejs/vue" target="_blank">Gitter Chat</a></li>
10      <li><a href="https://twitter.com/vuejs" target="_blank">Twitter</a></li>
11      <br>
12      <li><a href="http://vuejs-templates.github.io/webpack/" target="_blank">Docs for This Template</a></li>
13    </ul>
14    <h2>Ecosystem</h2>
15    <ul>
16      <li><a href="http://router.vuejs.org/" target="_blank">vue-router</a></li>
17      <li><a href="http://vuex.vuejs.org/" target="blank">vuex</a></li>
18      <li><a href="http://vue-loader.vuejs.org/" target="_blank">vue-loader</a></li>
19      <li><a href="https://github.com/vuejs/awesome-vue" target="_blank">awesome-vue</a></li>
20    </ul>
21  </div>
22 </template>
23
24 <script>
25   export default {
26     name: 'hello',
27     data() {
28       return {
29         msg: 'Welcome to Your Vue.js App',
30         employee: 'David James'
31       }
32     }
33   }
34 </script>
35
```

Рисунок 2.6 – Фрагмент коду Vue.js

Для роботи було вибрану саме React.JS завдяки тому він, що він має більш широкий функціонал ніж Angular та більше підходить для масштабних проектів ніж Vue.js.

2.1.3 MySQL

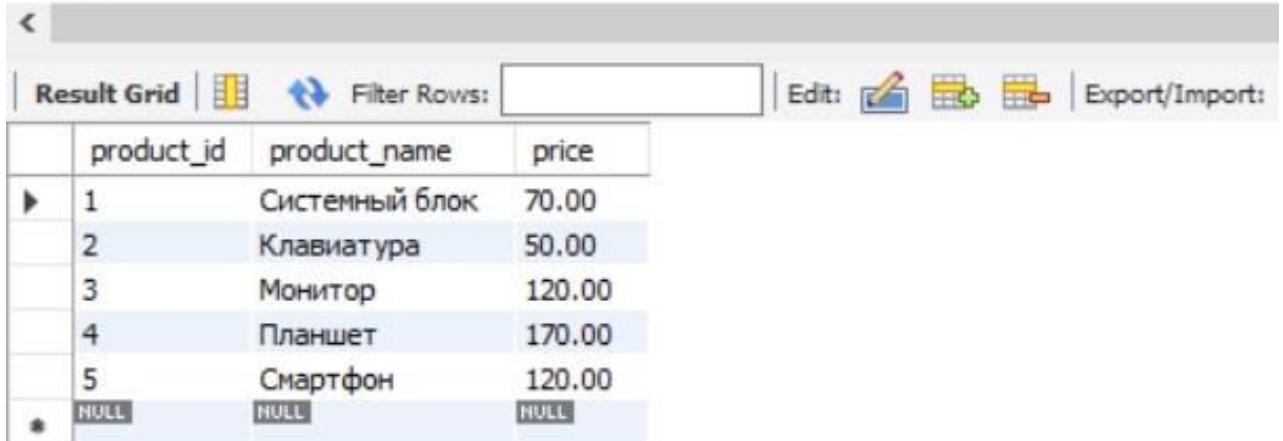
MySQL – це система для управління реляційними базами даних з відкритим кодом. Реляційні бази даних організовані між собою у вигляді таблиць [10].

На рисунку 2.7 зображено фрагмент коду MySQL

```

4
5 • UPDATE goods SET price = price + 10
6   LIMIT 1000;
7
8 • SELECT * FROM goods;
9

```



	product_id	product_name	price
▶	1	Системный блок	70.00
	2	Клавиатура	50.00
	3	Монитор	120.00
	4	Планшет	170.00
	5	Смартфон	120.00
•	NULL	NULL	NULL

Рисунок 2.7 – Фрагмент коду MySQL

Гнучкість цієї системи забезпечується підтримкою великого числа різних таблиць типу MyISAM, таблиці цього типу підтримують пошук по повному тексту. Так і таблиці InnoDB, таблиці цього типу підтримують транзакції на рівні окремих записів. Також в цій системі постійно з'являються нові типи таблиць це відбувається завдяки її відкритій архітектурі.

Таблиця 5 – Переваги та недоліки MySQL

Переваги	Недоліки
Безпека	Ненадійність
Швидкість	Повільно отримує оновлення
Функціональність	
Відкритість коду	

Можливим конкурентом для MySQL являється SQLite. На відміну від своїх конкурентів вона являється вбудованим рушієм баз даних. Це являється його відмінністю від його конкурентів.

На рисунку 2.8 зображено фрагмент коду SQLite.

```
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE "albums"
(
  [AlbumId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
  [Title] NVARCHAR(160) NOT NULL,
  [ArtistId] INTEGER NOT NULL,
  FOREIGN KEY ([ArtistId]) REFERENCES "artists" ([ArtistId])
  ON DELETE NO ACTION ON UPDATE NO ACTION
);
INSERT INTO "albums" VALUES(1,'For Those About To Rock We Salute You',1);
INSERT INTO "albums" VALUES(2,'Balls to the Wall',2);
INSERT INTO "albums" VALUES(3,'Restless and Wild',2);
INSERT INTO "albums" VALUES(4,'Let There Be Rock',1);
INSERT INTO "albums" VALUES(5,'Big Ones',3);
INSERT INTO "albums" VALUES(6,'Jagged Little Pill',4);
INSERT INTO "albums" VALUES(7,'Facelift',5);
INSERT INTO "albums" VALUES(8,'Warner 25 Anos',6);
```

Рисунок 2.8 – Фрагмент коду SQLite

Його особливість роботи полягає в тому що ця база даних не потребує окремого серверу, а він працює за допомогою інтеграції в мову програмування спеціальної бібліотеки [11].

В цьому його потенційний недолік, хоча подібні бібліотеки існують для всіх популярних мов програмування для мало популярної мови програмування бібліотеки може не знайтись.

Ця система, що можна зрозуміти з її опису, підходить для зовсім маленьких проектів, або коли немає можливості підняти повноцінний сервер з базою даних, що не підходить для даного проекту.

MongoDB – це документ – орієнтована система управління базами даних. Система написана на мові програмування C++. Є одним із прикладів No – SQL систем [12].

Такі бази даних не є реляційними, тобто вони не зберігають дані у вигляді таблиці. Основною перевагою таких баз даних є гнучкість та простота роботи з великим обсягом даних. Однією з проблем являється високий рівень вразливості [13].

На рисунку 2.9 зображено інтерфейс MongoDB.

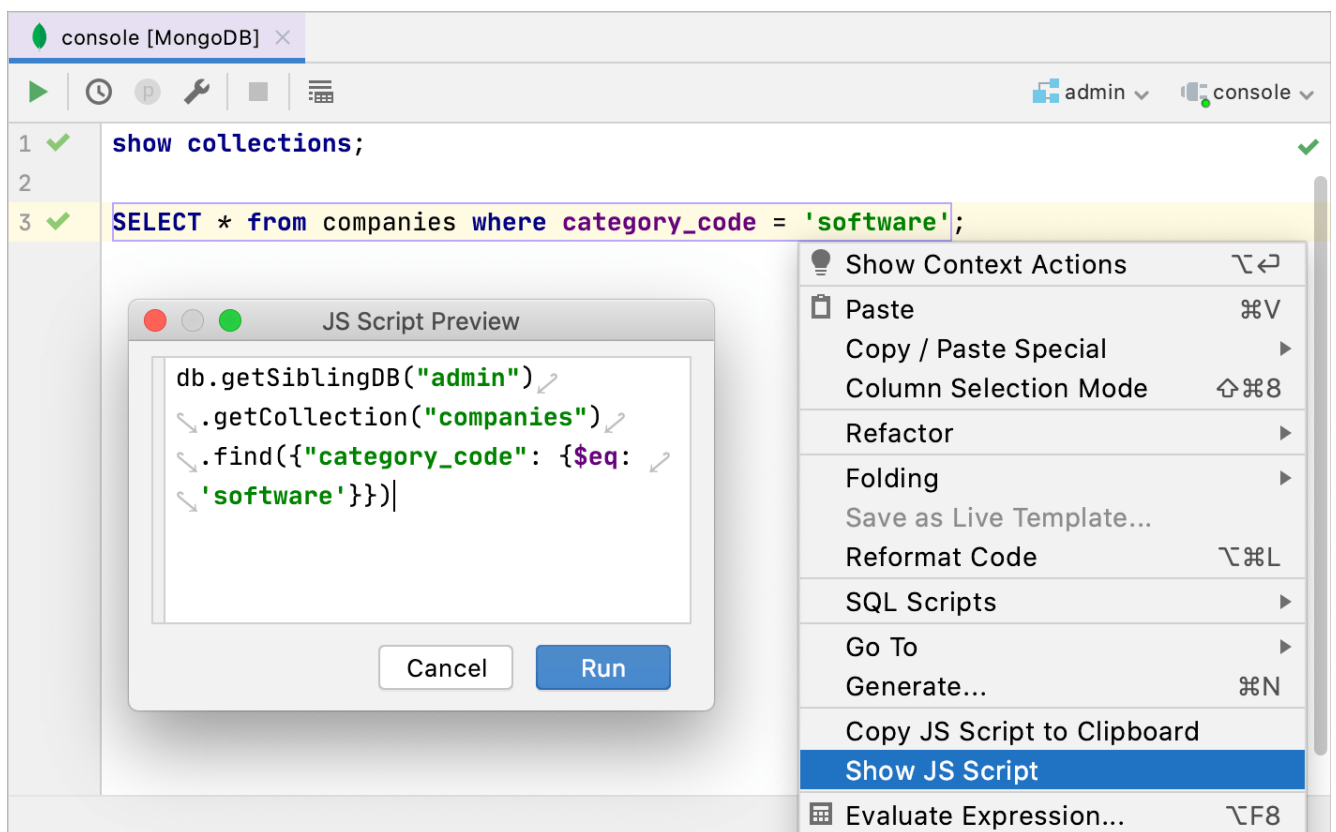


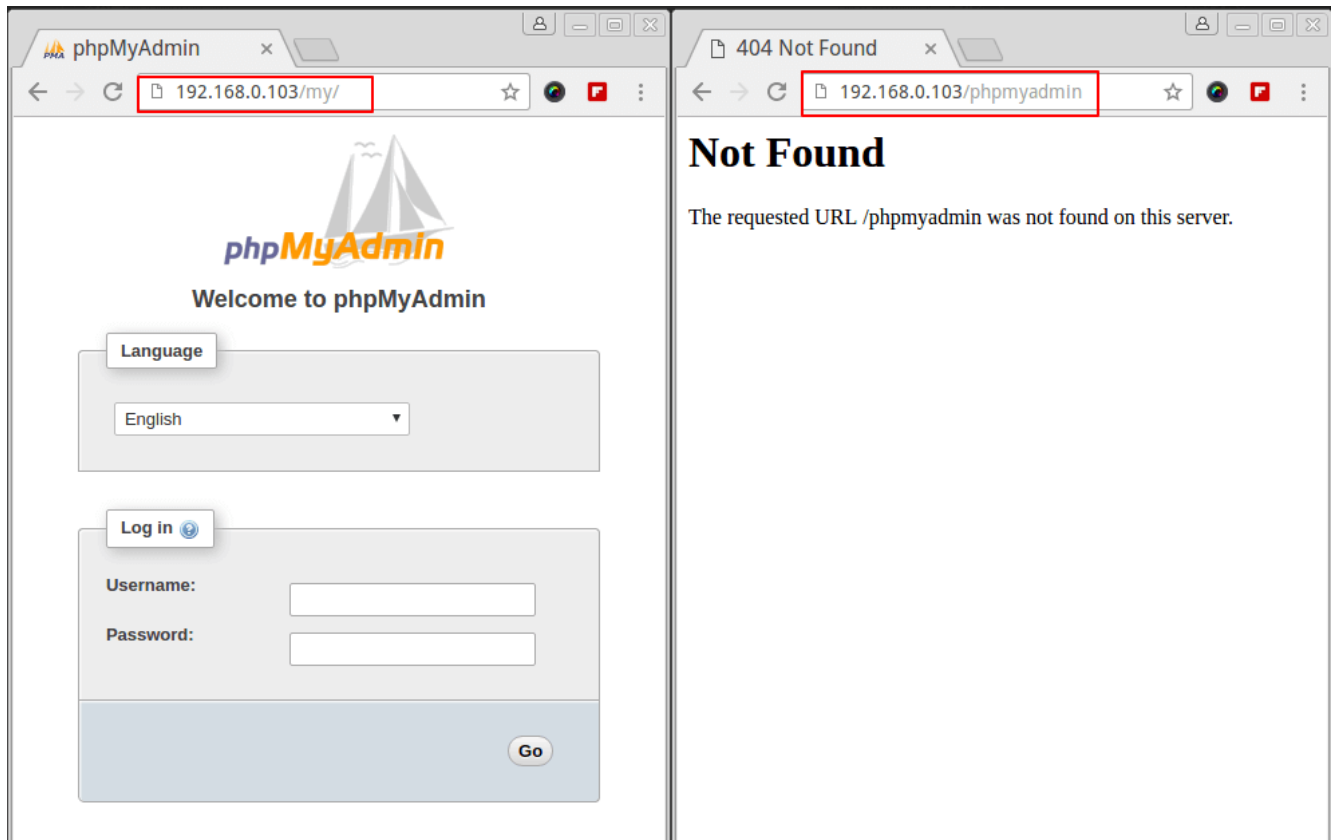
Рисунок 2.9 – Інтерфейс MongoDB

Вона була відсіяна через відсутність необхідності використовувати нереляційну систему баз даних.

2.1.4 PhpMyAdmin

phpMyAdmin – це додаток написаний на мові програмування PHP, з відкритим кодом, дозволяє виконувати повноцінне та дистанційне адміністрування бази даних [14].

На рисунку 2.10 зображено інтерфейс phpMyAdmin



На рисунку 2.10 зображено Інтерфейс phpMyAdmin

Дозволяє адмініструвати бази даних за допомогою графічного інтерфейсу, що може бути перевагою для людей, які слабо розуміють MySQL та, власне, самому MySQL.

Програма розповсюджується під ліцензією GNU General Public License та підтримує ініціативу GoPHP5.

Таблиця 6 – Переваги та недоліки phpMyAdmin.

Переваги	Недоліки
Широкий функціонал	Складність у використанні
Робота з БД графічно і за допомогою MySQL	Необхідність працювати над швидкістю роботи
Підтримує різні типи таблиць	Недостатня надійність
Підтримується більшістю CMS	Складний інтерфейс

2.1.5 HTML

HTML – це безальтернативна мова тегів, перша версія цієї мови була розроблена в 1992 році. Файли типу HTML мають розширення «.html» та «.htm» («.htm» застарілий стандарт). Структура HTML документу складається з:

1. Декларації типу документа, тут визначається тип документа;
2. Шапка документа, знаходиться під елементом «head», тут описана вся технічна інформація та певні додаткові відомості про документ, які не відображаються в браузері;
3. Тіло документа, знаходиться під елементом «body», тут міститься вся інформація яку потрібно відобразити у браузері [15].

На рисунку 2.11 зображений фрагмент коду HTML

```
<font size="5">Давньогрецькі боги</font>
<br />
<b>Посейдон</b> – володар світових вод, Океану,
в латинян йому відповідав <i>Нептун</i>, у слов'ян –
<i>цар Моря, Цар Морський, Водяник</i>.
```

Рисунок 2.11– приклад HTML розмітки

HTML документі пишуться за допомогою тегів.

Тег це елемент розмітки гіпертексту, текст відображається відповідно

властивостям тегу в якій він заключений.

Теги бувають відкритими та закритими, або одиночними.

На рисунку 2.12 зображено приклад закритого та відкритого тега

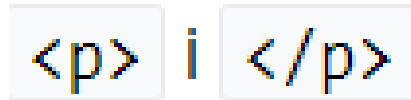


Рисунок 2.12 – приклад закритого та відкритого тега

Відкриті та закриті теги записуються наступним чином:

1. Написання відкриваючого тегу «<p>»;
2. Запис тексту, який повинен знаходитись всередині тегу;
3. Написання закриваючого тегу «</p>».

На рисунку 2.13 зображено приклад одиночного тега.



Рисунок 2.13 – приклад одиночного тега

Відкриті та закриті теги записуються наступним чином:

1. Написання тегу «<p>»;
2. Запис тексту, до якого повинен відноситися тег.

2.1.6 CSS

Cascading Style Sheets – це мова стилю сторінок, вона використовується для опису зовнішнього вигляду веб сторінок [16].

Перша версія CSS була випущена в грудні 1996 року, вона прийшла на заміну табличній версії.

Переваги CSS:

1. Єдина інформація про стиль для всього сайту, що дозволяє швидко вносити зміни в його дизайн;
2. Можливість створення окремих стилів для певних користувачів;
3. Зменшення об'єму сторінки дозволило збільшити швидкість її завантаження, та покращити її структуру;
4. У сучасних браузерів з'явилася можливість кешувати стилі.

2.1.7 PHP

PHP – скриптова мова програмування, використовується для створення веб – сторінок зі сторони сервера. PHP є однією із найпопулярніших мов в світі.

Метод роботи PHP: клієнту передається HTML – код інтерпретований сервером з відповідного PHP – коду. Це підвищує безпеку даних, але від цього страждає інтерактивність веб – сторінки [17].

Мінуси PHP

1. Незручність синтаксису мови;
2. Іноді нові версії PHP мають кардинальні зміни у порівнянні з попередніми;
3. Відсутність Unicode у версіях до 6.0 [18].

2.2 Середовище розробки

2.2.1 Visual Code

Visual Code – Редактор коду розроблений компанією Microsoft для Windows, Mac OS та Linux.

Позиціонується як легкий редактор для розробки веб та хмарних додатків. Програма підтримує роботу з Git, має вбудований відладчик та виділяється легкою системою інтеграції розширень [19].

Перша версія програми була представлена в 2015 році, на конференції Build 2015.

Також Visual Code підтримує розробку для платформ ASP.NET та Node.js. На рисунку 2.14 зображено інтерфейс Visual Code

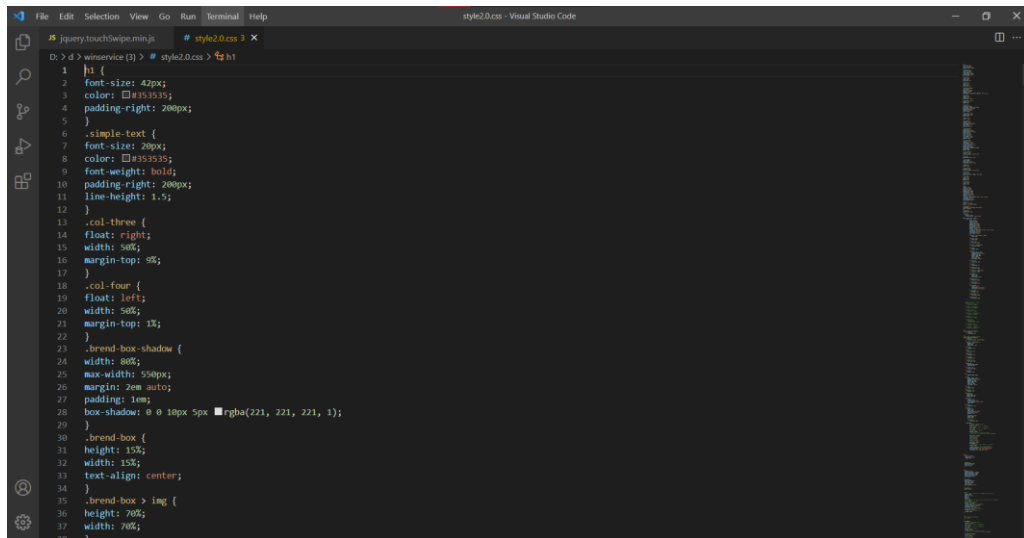


Рисунок 2.14 – Інтерфейс Visual Code

2.3 Проектування додатку

Перш за все, потрібно розробити структуру сайту. На цьому рисунку 2.15 зображено структуру сайту.

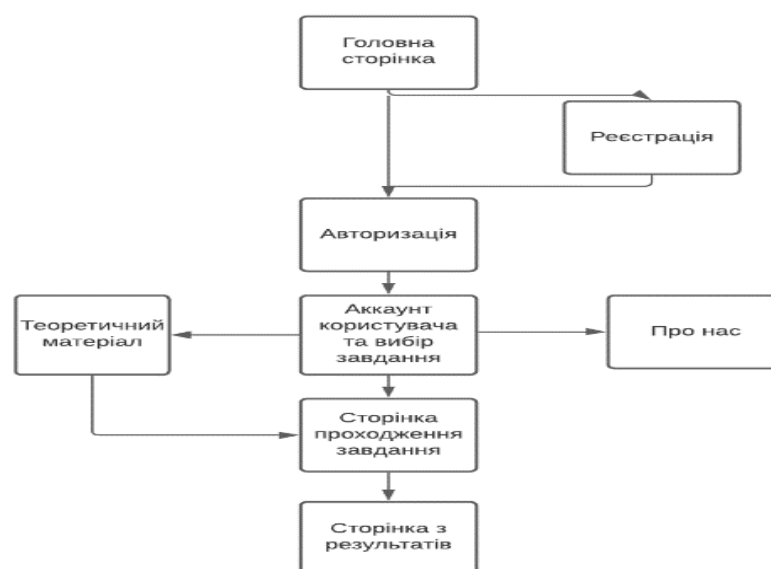


Рисунок 2.15 – структура сайту.

Для розробки додатку буде оптимальною структура взаємодії, коли дані, обробляються більше на клієнтській частині, а сервер лише зберігає базу даних.

Практично вся обробка даних здійснюється на стороні клієнта, за винятком процедур перевірки цілісності даних та іншої логіки, що відноситься до обслуговування бази даних, які краще виконувати на сервері.

За архітектурою, сайт має бути статичним.

На рисунку 2.16 зображено структуру взаємодії додатку.

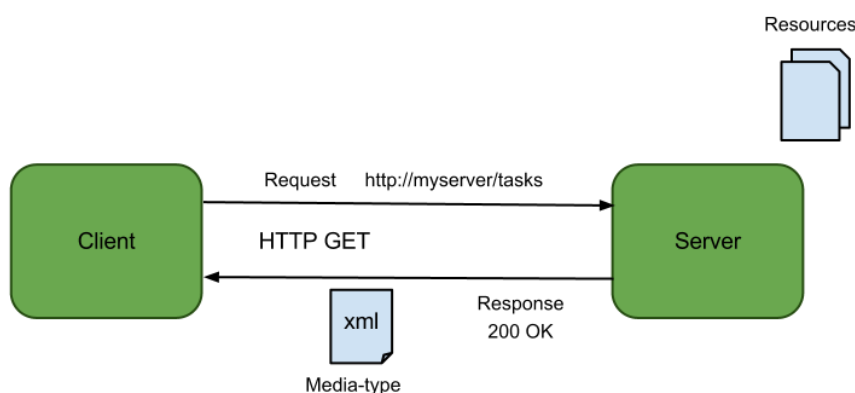


Рисунок 2.16 – структура взаємодії.

Висновки до другого розділу

У другому розділі було розглянуто питання технологій, які слід обрати скільки для розробки додатку. було розглянуто мови програмування та їх фреймворки, середовище розробки системи та додатки для управління базами даних.

На основі розглянутих даних було отримано наступні результати. Для розробки веб додатку було обрано мову програмування JavaScript та її фреймворк React.JS. Адміністрування бази даних буде виконано phpMyAdmin. А середовищем розробки був вибраний Visual Studio Code.

3 РОЗРОБКА ПЛАТФОРМИ

3.1 Загальні відомості

За результатами попереднього розділу для розробки веб додатків були вибрані наступні технології:

1. Клієнтська частина буде написана на HTML – відповідає за розмітку веб сторінки, CSS, JavaScript, ReactJS – відповідають за анімації та стилі;
2. База даних буде виконана за допомогою MySQL та phpMyAdmin;
3. Серверна частина буде виконана за допомогою PHP.

Фізичне місце розташування веб додатка – клієнт – сервер від приватного провайдера. На цьому рисунку 3.1 зображена конфігурація сервера.

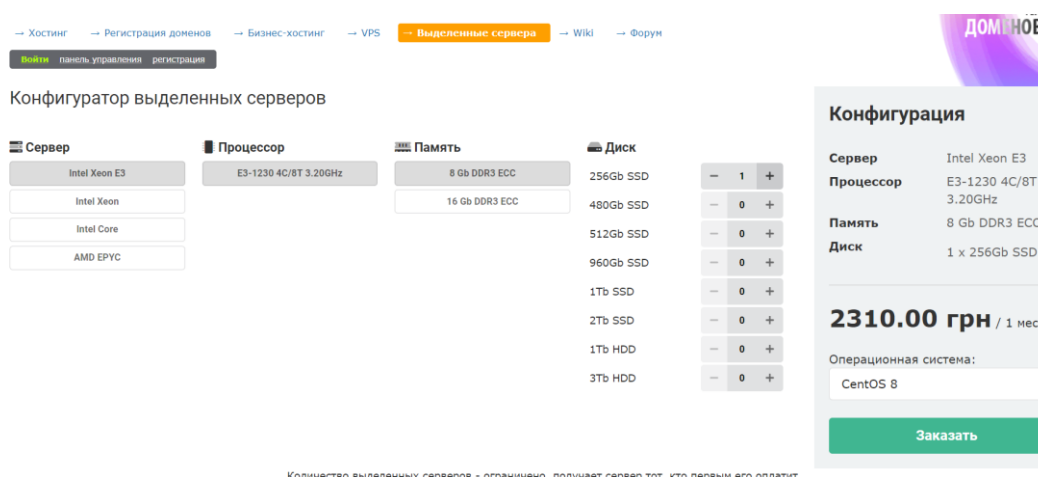


Рисунок 3.1 – Приклад оренди сервера

Серверна частина додатку відповідає за зв'язок між базою даних та користувацьким інтерфейсом.

База даних відповідає за зберігання та передачу даних користувачів веб додатку.

Клієнтська частина відповідає за демонстрацію, які демонструються користувачу.

3.2 Серверна частина

Серверна частина проекту була виконана за допомогою мови програмування PHP.

Вона повинна виконувати наступні завдання:

1. Отримувати та розпізнавати введені користувачем логін і пароль, перевіряти їх з даними в базі даних;
2. Отримувати з бази даних та відправляти користувачу дані завдань;
3. Отримувати та зберігати в базі даних завдання, які створюють користувачі та відправляти їх користувачам при запиті.

Для всіх цих завдань необхідно приєднуватися до бази даних, в PHP це виконується за допомогою функції `mysqli_connect()`.

`Mysqli_connect()` відкриває з'єднання з базою даних. Використовуючи 4 параметра.

1. Адресу сервера;
2. Ім'я користувача;
3. Пароль користувача;
4. Ім'я бази даних.

На рисунку 3.2 показаний приклад функції `mysqli_connect()`.

```
10
11  $link = mysqli_connect($host, $user, $password, $database)
12  |     or die("Помилка " . mysqli_error($link));
13
14
```

Рисунок 3.2 – Приклад функції `mysqli_connect()`

Далі нам потрібно отримати інформацію від бази даних. Для цього слід використовувати функцію `mysqli_query()`.

Іноді запит в цієї функції може бути доволі довгим, тому для полегшення роботи можна розділити одну велику строку на декілька малих, в такому випадку

для «склеювання» їх потрібно використовувати символ крапки «.».

На рисунку 3.3 показаний приклад функції `mysqli_query()`.

```
20
21     $result = mysqli_query($link, $query)
22     or die("Помилка " . mysqli_error($link));
23
```

Рисунок 3.3 – Приклад функції `mysqli_query()`

Також корисною для опрацювання помилок є функція `die()`. Вона зупиняє виконання сценарію, та показує повідомлення про код в якому сталася помилка.

Реалізація підключення до бази даних. Спочатку необхідно задати параметри підключення.

На рисунку 3.5 зображено приклад параметрів підключення.

```
$db_link = "localhost";
$db_login = "admin";
$db_password = "123asd";
$db_name = "english_db";
```

Рисунок 3.4 – Параметри підключення

Після задання параметрів, нам потрібно створити підключення. З цим справляється функція «`$db`». Зразу після неї ми задаємо коректне кодування.

Функція «`$data`» збирає всі необхідні дані, для запиту. Після збору даних, можна починати підготовку MySQL запиту за допомогою функції «`$query`». Після цього можна виконувати запит до бази даних за допомогою функції «`$query->execute($data)`».

На рисунку 3.6 зображено приклад коду, який робить запит до бази даних.

```

try {

    $db = new PDO("mysql:host=$db_host;dbname=$db_base", $db_user, $db_password);

    $db->exec("set names utf8");

    $data = array( 'login' => $login, 'password' => $password );

    $query = $db->prepare("INSERT INTO $db_table (login, password) values (:login, :password)");

    $query->execute($data);

    $result = true;
} catch (PDOException $e) {

    print "Error: " . $e->getMessage() . "<br/>";
}
}

```

Рисунок 3.5 – Фрагмент коду PHP.

Реєстрація відбувається за допомогою збереження даних користувача в базі даних. Дані передаються пароль шифрується, пароль шифрується за допомогою алгоритму «MD5», це забезпечує безпеку даних. Дані передаються на сервер, за допомогою PHP кода зображеного на рисунку 3.6.

```

<?php
include_once "bd.php";

if (isset($_POST['login']) && isset($_POST['password']) && isset($_POST['email'])) {
    if ($_POST['login'] != '') {
        if ($_POST['password'] != '') {
            if ($_POST['email'] != '') {
                $pass = md5($_POST['password']);

                if ($db->query("INSERT INTO `users` (`id`, `login`, `password`, `email`) VALUES (
                    'NULL',
                    '', $_POST['login'] .'',
                    '', $pass .'',
                    '', $_POST['email'] .''
                ))) {
                    echo '1';
                }
            } else {
                echo '0';
            }
        } else {
            echo '0';
        }
    } else {
        echo '0';
    }
} else {
    echo '0';
}
?>

```

Рисунок 3.6 – фрагмент кода для реєстрації.

Авторизація відбувається за допомогою отримання даних користувача від бази даних.

Дані передаються за допомогою PHP кода зображеного на рисунку 3.7.

```

<?php
include_once "bd.php";

if (isset($_POST['login']) && isset($_POST['password'])) {
    if ($_POST['login'] != '') {
        if ($_POST['password'] != '') {
            $pass = md5($_POST['password']);

            $log = $db->query("SELECT `id` FROM `users` WHERE `login` = '" . $_POST['login'] . "' && `password` = '" . $pass . "'");
            if ($log->num_rows != 0) {
                echo '1';
            } else {
                echo '0';
            }
        } else {
            echo '0';
        }
    } else {
        echo '0';
    }
} else {
    echo '0';
}
}

```

Рисунок 3.7 – фрагмент кода для авторизації.

На цій сторінці отримуються результати від користувача. Сервер розпізнає, з якого розділу йому відправили відповіді за допомогою «id». Далі в циклі порівнюється відповідь користувача з правильною відповіддю. Після цього в таблицю результатів записується «id» користувача та кількість правильних відповідей. Якщо, це було це було перше проходження цього тесту, то за допомогою команди «INSERT INTO» і відповідно, якщо повторне, то за допомогою команди «UPDATE». На рисунку 3.8 зображено відповідний фрагмент коду.

```

<?php
include_once "bd.php";

if (isset($_POST['login']) && isset($_POST['password']) && isset($_POST['result_arr'])) {
    if ($_POST['login'] != '') {
        if ($_POST['password'] != '') {
            $pass = md5($_POST['password']);

            $u = $db->query("SELECT `id` FROM `users` WHERE `login` = '" . $_POST['login'] . "' && `password` = '" . $pass . "'");
            if ($u->num_rows != 0) {
                $u = $u->fetch_array(MYSQLI_ASSOC);

                $good_answers = 0;

                foreach ($_POST['result_arr'] as $value) {
                    $answer = $db->query("SELECT `answer` FROM `answers` WHERE `test_num` = '" . $value['test_num'] . "' && `test_num_part` = '" . $value['test_num_part'] . "'");
                    if ($answer->num_rows != 0) {
                        $answer = $answer->fetch_array(MYSQLI_ASSOC);

                        if ($answer['answer'] == $value['answer']) {
                            $good_answers++;
                        }
                    }
                }

                $check_duplicate = $db->query("SELECT `id` FROM `history` WHERE `user_id` = '" . $u['id'] . "' && `topic_id` = '" . $_POST['result_arr'][0]['test_num'] . "'");
                if ($check_duplicate->num_rows != 0) {
                    $db->query("UPDATE `history` SET `good_answers` = '" . $good_answers . "' WHERE `topic_id` = '" . $_POST['result_arr'][0]['test_num'] . "' && `user_id` = '" . $u['id'] . "'");
                } else {
                    $db->query("INSERT INTO `history` (`topic_id`, `user_id`, `good_answers`) VALUES ('" . $_POST['result_arr'][0]['test_num'] . "', '" . $u['id'] . "', '" . $good_answers . "')");
                }

                echo 1;
            } else {
                echo 0;
            }
        } else {
            echo 0;
        }
    } else {
        echo 0;
    }
} else {
    echo 0;
}
}
}

```

Рисунок 3.8 – фрагмент кода збереження результатів.

3.3 База даних

Для зберігання потрібних даних потрібна база даних.

Вона була реалізована за допомогою СУБД MySQL та додатку для адміністрування phpMyAdmin.

Щоб створити базу даних нам потрібно використати РМА, в даному випадку використовується phpMyAdmin. На рисунку 3.9 зображений процес створення бази даних.

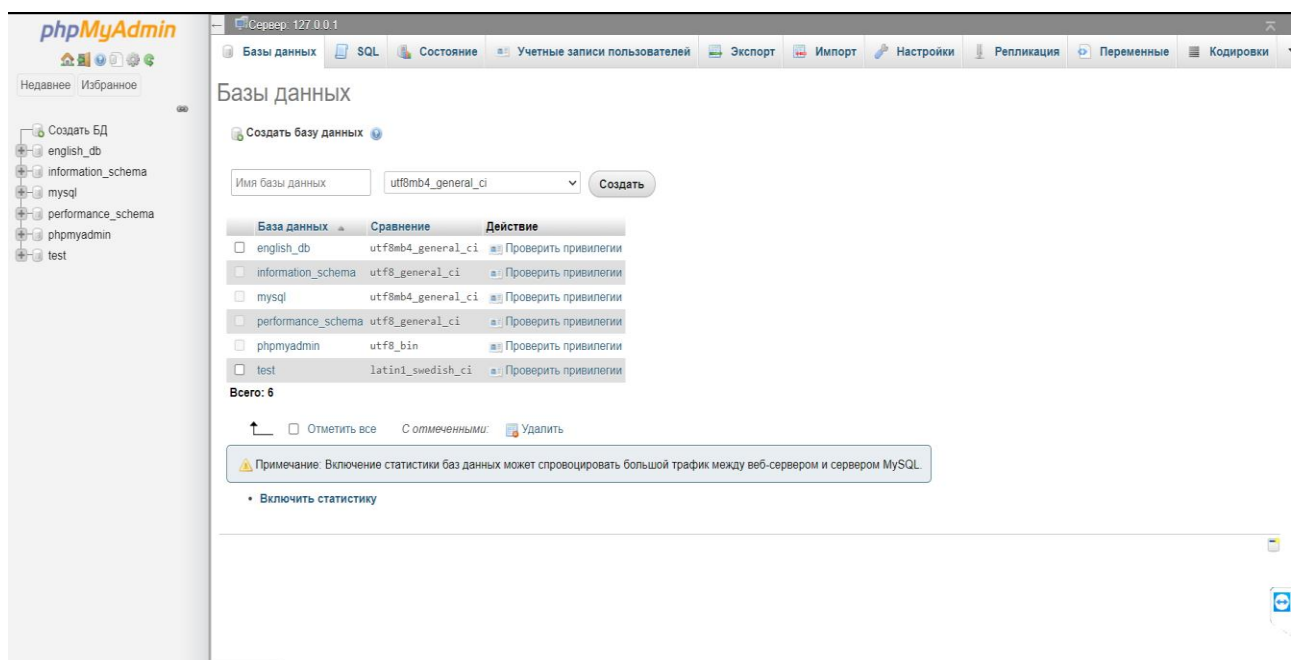


Рисунок 3.9 – Створення бази даних.

Далі потрібно було створити необхідні таблиці, відповідно функціоналу та логіці роботи додатку. На рисунку 3.10 зображено процес створення таблиць в реляційній базі даних.

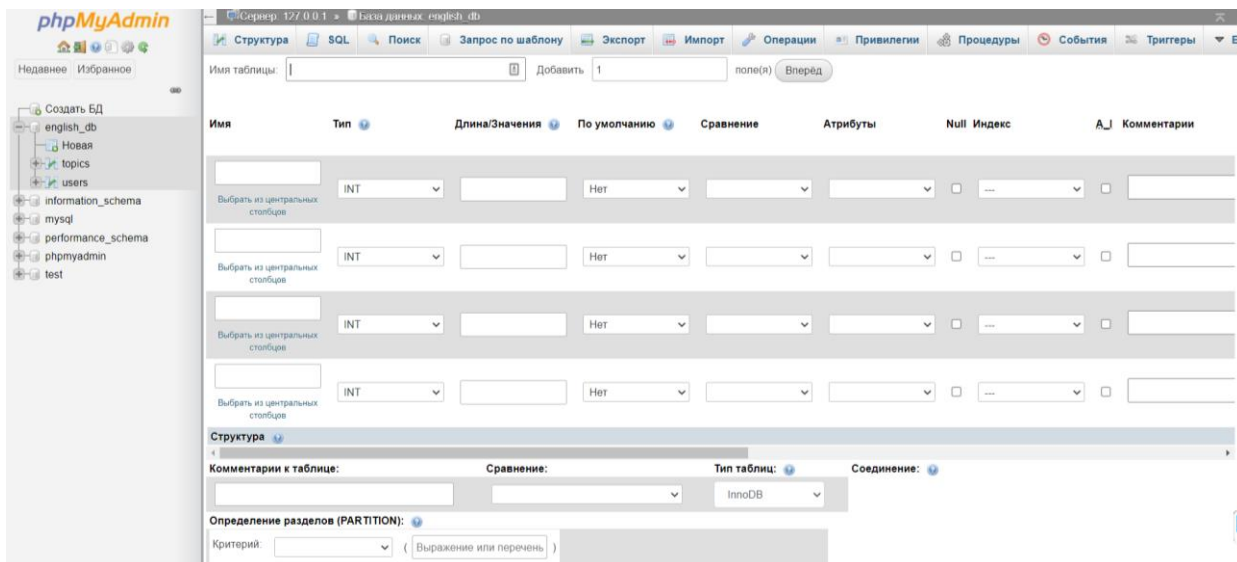


Рисунок 3.10 – процес створення таблиці в базі даних.

Для додатку необхідно було створити чотири таблиці. На рисунку 3.11 зображено загальний вигляд бази даних, після того як всі необхідні таблиці були створені.

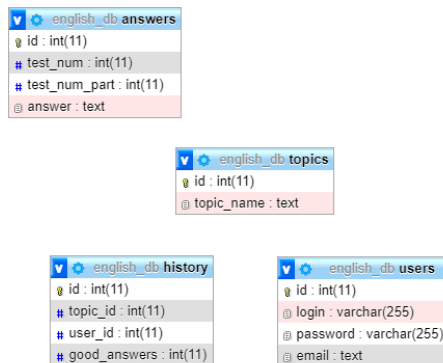


Рисунок 3.11 – фінальний вигляд бази даних.

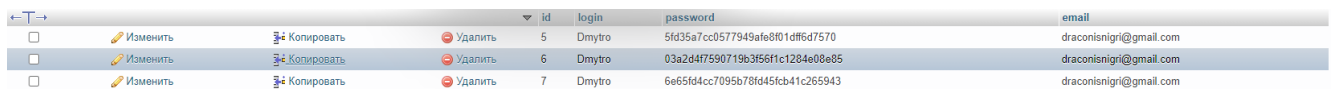
Схематичне зображення першої бази даних. Перша база даних необхідна для реєстрації для авторизації користувачів.

Таблиця 7, зображення першої реляційної бази даних.

login	user_password	id	email
Dmytro	Pass2588/	38514	dmytro@gmail.com
DmytroD	Dcfg256!	78235	dmytrod@gmail.com
Victor	Tehefbdu8585	74285	victor@gmail.com

В колонці «login» зберігається логін користувача, «password» – пароль, «id» – «id» даних в таблиці, «email» – зберігає електронну адресу користувача.

На рисунку 3.12 зображено вигляд першої бази даних.



	id	login	password	email
<input type="checkbox"/>	5	Dmytro	5fd35a7cc0577949afe8f01dff6d7570	draconisnigr@gmail.com
<input type="checkbox"/>	6	Dmytro	03a2d4ff7590719b3f56f1c1284e08e95	draconisnigr@gmail.com
<input type="checkbox"/>	7	Dmytro	6e65f84cc7095b78fd45fcb41c265943	draconisnigr@gmail.com

Рисунок 3.12 – вигляд першої бази даних.

Друга база даних зберігає відповіді на завдання.

Таблиця 8, зображення другої реляційної бази даних.

id	topic_id	user_id	good_answers
1	1	3	11
2	3	2	2
3	2	2	0

В колонці «id» – «id» даних в таблиці, «good_answers» – бали за проходження завдання, «topic_id» – номер завдання, «user_id» – «id» користувача.

На рисунку 3.13 показаний вигляд другої бази даних..



	id	topic_id	user_id	good_answers
<input type="checkbox"/>	1	1	3	11
<input type="checkbox"/>	2	3	3	2
<input type="checkbox"/>	3	2	3	5
<input type="checkbox"/>	4	1	5	2
<input type="checkbox"/>	5	3	7	0

Рисунок 3.13 – вигляд другої бази даних.

Третя база даних зберігає лише список завдань.

Таблиця 9, зображення третьої реляційної бази даних.

id	topic_name
1	Завдання 1
2	Завдання 2
3	Завдання 3

В колонці «id» – «id» даних в таблиці, «topic_name» – назви завдань.

На рисунку 3.13 зображено вигляд третьої бази даних.

	id	topic_name
<input type="checkbox"/>	1	Перше завдання
<input type="checkbox"/>	2	Друге завдання
<input type="checkbox"/>	3	Третє завдання

Рисунок 3.13 – вигляд третьої бази даних.

Схематичне зображення четвертої бази даних. Четверта база даних необхідна для зберігання даних завдань.

Таблиця 10, зображення першої реляційної бази даних.

id	test_num	test_num_part	task	answer
1	1	1	(task)	had
2	1	2	(task)	was
3	1	3	(task)	went

В колонці «id» – «id» даних в таблиці, «test_num» – номер групи завдань, «test_num_part» – номер конкретного завдання в групі, «answer» – відповіді до завдань, «task» – зберігає в собі завдання.

3.4 Клієнська частина

Була виконана за допомогою мови програмування JavaScript та її фреймворку React.JS.

Зв'язок клієнтської частини з сервером та базою даних відбувається за допомогою надсилання HTTP запитів. Дизайн виконаний у мінімалістичному стилі, що робить його ненав'язливим та водночас функціональним.

HTTP — захищений протокол передачі даних, що використовується в комп'ютерних мережах.

Розробку цієї частини додатку було розпочато з верстки необхідних сторінок. На рисунку 3.14 зображено «head» створений для верстки сайту.

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="Keywords" content="українська, англійська, навчання, мова, дистанційна, UkrEngLearnig, тести">
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>UkrEngLearnig</title>
  <link rel="stylesheet" href="../css/style.css">
  <link rel="stylesheet" href="../css/splide.min.css">
  <script src="https://cdn.jsdelivr.net/npm/@splidejs/splide@latest/dist/js/splide.min.js"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
</head>
```

Рисунок 3.14 – фрагмент коду «head»

Перше, що потрібно зробити користувачу це зареєструватися (Додаток А).

Перевірка правильності введення даних та їх валідація проходить саме на клієнтській стороні.

На рисунку 3.15 зображено інтерфейс поля авторизації.

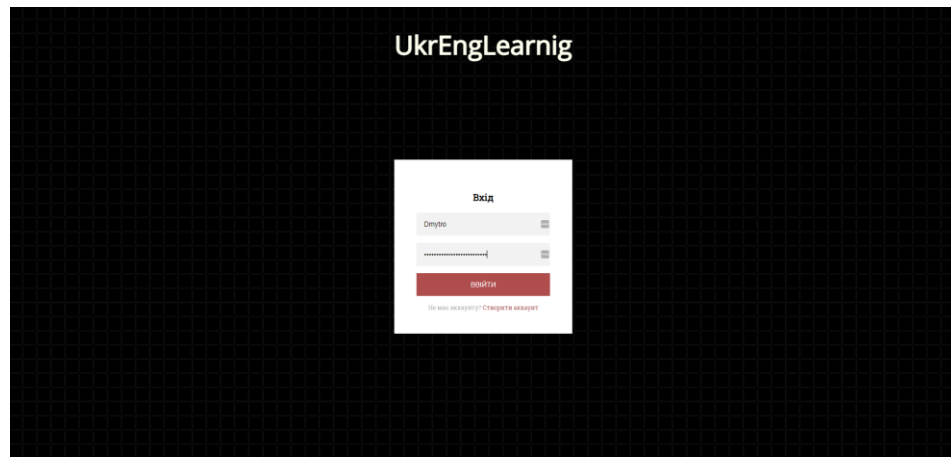


Рисунок 3.15 – інтерфейс поля авторизації

Після того, як користувач зареєструвався та авторизувався, він може починати проходити завдання.

Користувачам доступні завдання трьох типів:

1. Переклад;
2. Тест;
3. Вставити в готове речення слово або словосполучення.

Всі завдання виконані на React.JS, та обчислення проходить на стороні клієнта, з сервера користувач отримує лише завдання та відповіді до них. Це дозволяє мінімізувати навантаження на інтернет трафік користувача.

Мета завдання «Переклад» полягає в тому, щоб перекласти текст який наведений в завданні на іншу мову.

На рисунку 3.16 показаний приклад завдання «Переклад».

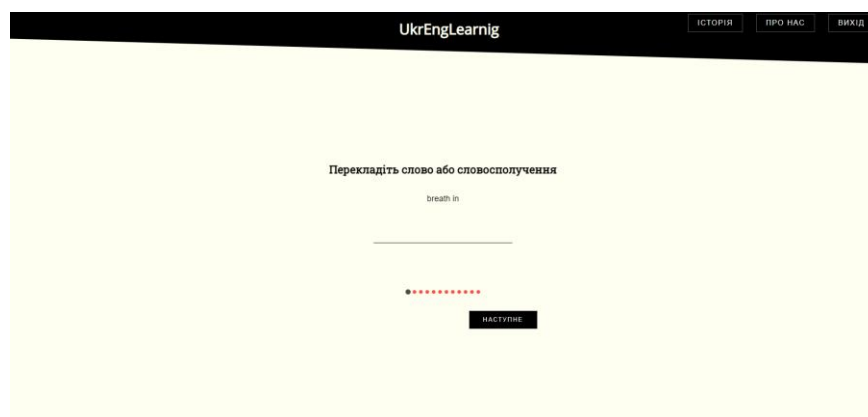


Рисунок 3.16 – приклад завдання «Переклад»

Під час виконання завдання «Тест» користувачу потрібно вибрати правильну відповідь з одного, або декількох варіантів.

На рисунку 3.17 показаний приклад завдання «Тест».



Рисунок 3.17 – приклад завдання «Тест»

Завдання «Вставити слово, або словосполучення» полягає в тому, що користувач повинен підібрати вдалий варіант слова, або словосполучення до конкретного речення.

На рисунку 3.18 показаний приклад завдання «Вставити слово, або словосполучення».

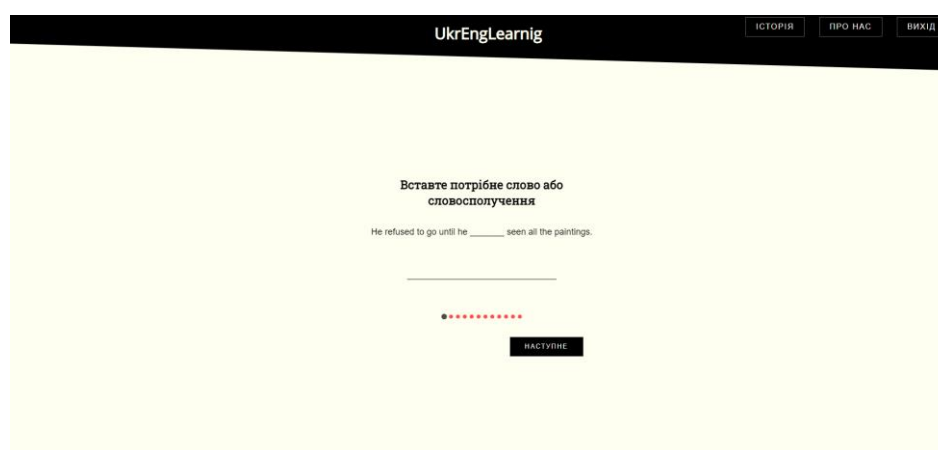


Рисунок 3.18 – приклад завдання «Вставити слово, або словосполучення»

За один раз завантажується лише одне питання, це дозволяє завантажувати з сервера питання по одному. Це допомагає знизити будь-які затримки при завантаженні завдання, завдяки тому, що зменшується розмір JSON-відповіді від сервера.

Перед тим як почати проходження завдання, користувач має можливість ознайомитися з теоретичним матеріалом. На рисунку 3.19 зображено інтерфейс вкладки з теорією.



Рисунок 3.19 – приклад інтерфейсу створення завдання.

Після проходження завдань користувач отримує можливість переглянути свій результат. Після перегляду результат записується в базу даних (Додаток Б). Тому користувач може переглянути свій результат в будь-який момент на відповідній вкладці (Історія).

Також за допомогою спеціальної вкладки користувач може прочитати загальну інформацію про платформу, на вкладці під назвою «Про нас».

Після завершення проходження завдань, користувач має змогу покинути свій аккаунт.

3.5 Тестування

3.5.1 Інтерфейсу користувача

Тестування інтерфейсу користувача – це процес тестування інтерфейсу на графічні та графічно – функціональні баги.

Додаток повністю пройшов тестування інтерфейсу користувача, в ньому повністю відсутні всі баги цього типу особливо при адаптації до екрану в різних пристроїв.

Також були знайдені та виправлені:

1. Помилки у функціональності за допомогою інтерфейсу;
2. Втрата даних які передаються через інтерфейс;
3. Помилки в інтерфейсі.

На рисунку 3.20 показано тестування адаптації веб додатка.

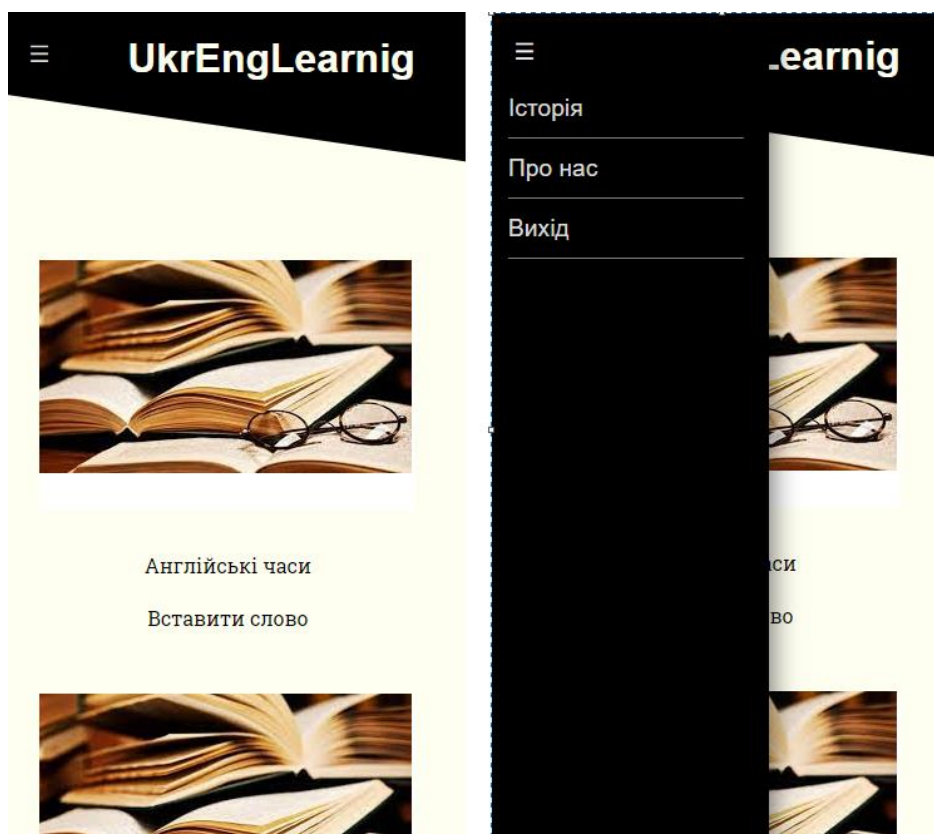


Рисунок 3.20 – вигляд мобільної версії додатку

3.5.2 Функціональне тестування

Функціональні тести повинні охоплювати всі реалізовані функції додатку. Функціональні тести створюються за зовнішніми специфікаціями функцій та проектною інформацією.

Була проведена ідентифікація множини функціональних вимог

Були виявлені та усунуті всі невідповідності між реальною поведінкою функцій та очікуваною поведінкою.

Підчас цього виду тестування було виявлено наступні баги:

1. Невірне повернення «user_id» з бази даних;
2. Невірно зараховані відповіді в тестах;
3. Помилка при введенні логіну та паролю, навіть якщо вони були введені правильно;
4. Невірний підрахунок балів, лічильник нараховував більше балів ніж повинен був дозволяти ліміт.

Висновки до третього розділу

В третьому розділі була розглянута тема: повноцінної розробки веб додатка. По завершенню цього розділу, була закінчена розробка та тестування додатку.

База даних додатку була розроблена на MySQL та phpMyAdmin.

Серверна частина додатку була розроблена на мові програмування PHP.

Користувацька частина додатку була розроблена на HTML, CSS, JavaScript та його фреймворку React.JS.

ВИСНОВКИ

У ході роботи над випускною кваліфікаційною роботою було виконано усі поставлені завдання.

Був проведений аналіз усіх переваг та недоліків аналогів, були висунуті вимоги, щодо розробки веб додатка.

Опираючись на вимоги , був складений план розробки, який вмщував в собі весь необхідний функціонал, який би відрізняв його від найближчих аналогів.

У процесі аналізу для виконання необхідних завдань були вибрані наступні технології: HTML, CSS, JavaScript, ReactJS, PHP, phpMyAdmin, MySQL, та середовище розробки Visual Code.

Результатом діяльності, є розробка україномовного веб – додатку для вивчення англійської мови.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1) Частка учнів в школах України, що навчаються українською [Електронний ресурс] – Режим доступу:
http://statistika.in.ua/mova2021/ukrainska_v_shkolah
- 2) Сайт компанії Busuu [Електронний ресурс] – Режим доступу:
<https://www.busuu.com/>
- 3) Сайт компанії Duolingo [Електронний ресурс] – 2016 – Режим доступу:
<https://uk.duolingo.com/>
- 4) Lingualeo [Електронний ресурс] – Режим доступу:
<https://lingualeo.com/uk>
- 5) Whats is JavaScript [Електронний ресурс]: – Режим доступу:
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- 6) About JavaScript [Електронний ресурс] – Режим доступу:
https://flexberry.github.io/en/gbt_javascript.html
- 7) Tutorial into React. [Електронний ресурс] / - Режим доступу:
<https://reactjs.org/tutorial/tutorial.html>
- 8) About Angular [Електронний ресурс] /- Режим доступу:
<https://angular.io/docs>
- 9) Vue.js API - [Електронний ресурс] /- Режим доступу:
<https://vuejs.org/v2/api/>
- 10) What is MySQL [Електронний ресурс] – Режим доступу:
<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- 11) About SQLite [Електронний ресурс]:- Режим доступу:
<https://www.sqlite.org/about.html>.
- 12) What is MongoDB? Introduction, Architecture, Features & Example [Електронний ресурс] – Режим доступу: <https://www.guru99.com/what-is-mongodb.html>

13) MongoDB Documentation [Электронный ресурс]: Manual – Режим доступа: <https://docs.mongodb.com/manual>

14) PhpMyAdmin [Электронный ресурс]: API – Режим доступа: <https://www.phpmyadmin.net/>

15) HTML: HyperText Markup Language [Электронный ресурс] /– Режим доступа: <https://developer.mozilla.org/ru/docs/Web/HTML>

16) What is CSS? [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics

17) What is PHP? [Электронный ресурс] /– Режим доступа: <https://www.php.net/manual/en/intro-what-is.php>

18) PHP Tutorial [Электронный ресурс] – Режим доступа: <https://www.w3schools.com/php/DEFAULT.asp>

19) What is VisualCode? [Электронный ресурс] /– Режим доступа: <https://code.visualstudio.com/docs>

ДОДАТОК А

Код, за допомогою якого дані користувача записуються в базу даних.

```
<?php
    include_once "bd.php";

    if (isset($_POST['login']) && isset($_POST['password']) && isset($_POST['email'])) {
        if ($_POST['login'] != '') {
            if ($_POST['password'] != '') {
                if ($_POST['email'] != '') {
                    $pass = md5($_POST['password']);

                    if ($db->query("INSERT INTO `users` (`id`, `login`, `password`, `email`) VALUES (
                        'NULL',
                        '". $_POST['login'] ."',
                        '$pass ."',
                        '". $_POST['email'] ."'
                    ))) {
                        echo '1';
                    }
                } else {
                    echo '0';
                }
            } else {
                echo '0';
            }
        } else {
            echo '0';
        }
    } else {
        echo '0';
    }
}
```

Код за допомогою якого дані користувача отримуються та відбувається авторизація.

```
<?php
    include_once "bd.php";

    if (isset($_POST['login']) && isset($_POST['password'])) {
        if ($_POST['login'] != '') {
            if ($_POST['password'] != '') {
                $pass = md5($_POST['password']);
            }
        }
    }
}
```

```
        $log = $db-
>query("SELECT `id` FROM `users` WHERE `login` = '". $_POST['login'] ."' && `password` = '". $pass ."'");
        if ($log->num_rows != 0) {
            echo '1';
        } else {
            echo '0';
        }
    } else {
        echo '0';
    }
} else {
    echo '0';
}
}
?>
```

ДОДАТОК Б

Код за допомогою, якого відбувається обробка результатів.

```

<?php
    include_once "bd.php";

    if (isset($_POST['login']) && isset($_POST['password']) && isset($_POST['result_a
rr'])) {
        if ($_POST['login'] != '') {
            if ($_POST['password'] != '') {
                $pass = md5($_POST['password']);

                $u = $db-
>query("SELECT `id` FROM `users` WHERE `login` = '". $_POST['login'] ."' && `password
` = '". $pass ."'");
                if ($u->num_rows != 0) {
                    $u = $u->fetch_array(MYSQLI_ASSOC);

                    $good_answers = 0;

                    foreach ($_POST['result_arr'] as $value) {

                        $answer = $db-
>query("SELECT `answer` FROM `answers` WHERE `test_num` = '". $value['test_num'] ."'
&& `test_num_part` = '". $value['test_num_part'] ."'");
                        if ($answer->num_rows != 0) {
                            $answer = $answer->fetch_array(MYSQLI_ASSOC);

                            if ($answer['answer'] == $value['answer']) {
                                $good_answers++;
                            }
                        }
                    }

                    $check_duplicate = $db-
>query("SELECT `id` FROM `history` WHERE `user_id` = '". $u['id'] ."' && `topic_id` =
'". $_POST['result_arr'][0]['test_num'] ."'");
                    if ($check_duplicate->num_rows != 0) {
                        $db-
>query("UPDATE `history` SET `good_answers` = '". $good_answers ."' WHERE `topic_id`
= '". $_POST['result_arr'][0]['test_num'] ."' && `user_id` = '". $u['id'] ."'");
                    } else {
                        $db-
>query("INSERT INTO `history` (`topic_id`, `user_id`, `good_answers`) VALUES ('. $_P
OST['result_arr'][0]['test_num'] .'", '". $u['id'] .'", '". $good_answers .'");
                    }

                    echo 1;
                } else {

```

```
        echo '0';
    }
} else {
    echo '0';
}
} else {
    echo '0';
}
}
?>
```



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА УКРАЇНОМОВНОГО WEB-ДОДАТКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ ЗА ДОПОМОГОЮ REACT.JS

Виконав студент 4 курсу
Групи ПД-44
Двірний Дмитро Юрійович
Керівник роботи
Яскевич Владислав Олександрович

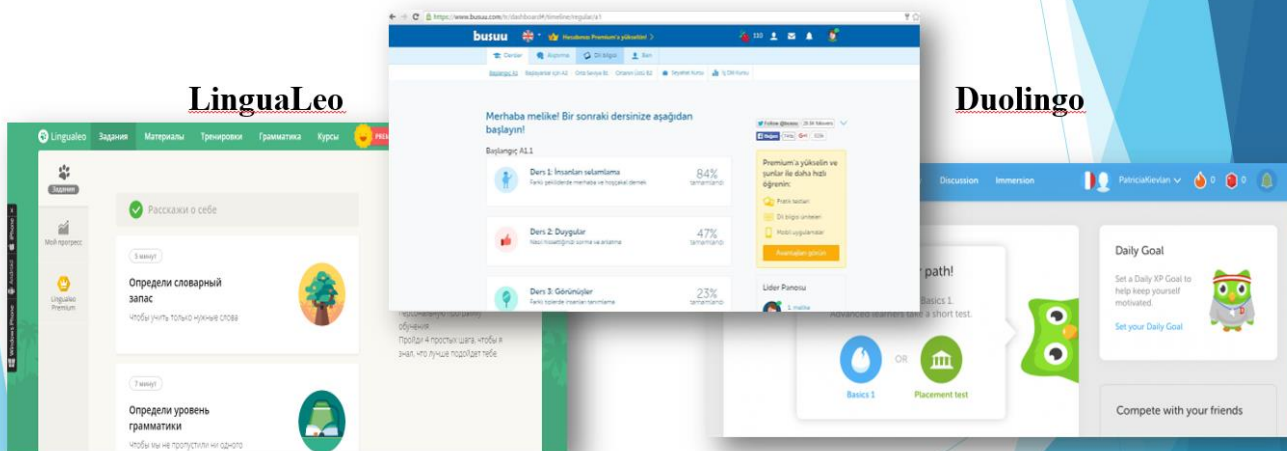
Київ – 2021

АНАЛОГИ

Busuu

LinguaLeo

Duolingo



ПОРІВНЯЛЬНИЙ АНАЛІЗ ПЕРЕВАГ ТА НЕДОЛІКІВ

Назва додатку	Переваги	Недоліки
Duolingo	<ul style="list-style-type: none"> Широкий вибір мов Велика база завдань 	<ul style="list-style-type: none"> Лише ігрова форма навчання Наявність платних завдань Завантаженість графічними зображеннями
Busuu	<ul style="list-style-type: none"> Приємний інтерфейс Декілька мов для вивчення Перевірена часом база завдань 	<ul style="list-style-type: none"> Відсутність підтримки української мови Наявність платних завдань
LinguaLeo	<ul style="list-style-type: none"> Повна локалізація українською Доступність у вигляді веб додатка та у вигляді додатка для Android і IOS. 	<ul style="list-style-type: none"> Перевантажений інтерфейс Наявність платних завдань Деякі матеріали доступні лише у формі відео

3

Порівня швидкості завантаження

The image displays two screenshots comparing the loading speed of two websites. On the left is the UkrEngLearnig website, which has a Lighthouse performance score of 100. On the right is the Duolingo website, which has a Lighthouse performance score of 30. The Duolingo screenshot also shows a warning message: "There were issues affecting the run of Lighthouse. The page loaded too slowly to finish within the time limit. Results may be incomplete." The Duolingo website interface is in Russian, with the text "Изучайте языки с нами. Всегда бесплатно." (Learn languages with us. Always free).

4

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи:** покращення процесу вивчення англійської мови за допомогою україномовного додатку на основі React.js;
- **Об'єкт дослідження:** вивчення англійської мови;
- **Предмет дослідження:** україномовна платформа для вивчення англійської, розроблена на основі React.js;

5

ТЕХНІЧНІ ЗАВДАННЯ

- 1. Проаналізувати можливі додатки-аналоги;
- 2. Описати створення україномовного веб-додатку для вивчення англійської;
- 3. Створити веб-додаток;
- 4. Протестувати створений веб-додаток.

6

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Клієнтська частина додатку реалізована за допомогою:

- HTML,
- CSS,
- JavaScript,
- React.JS



Серверна частина додатку реалізована за допомогою:

- PHP,
- PHPMyAdmin,
- MySQL

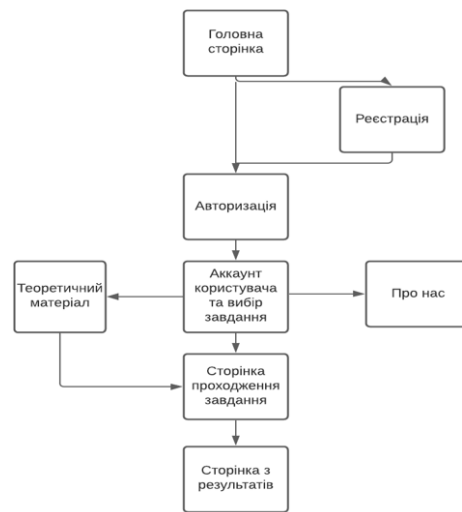
7

Архітектура веб-додатку



8

Схема роботи веб-додатку



9

9

Структура бази даних

english_db answers	
id	int(11)
test_num	int(11)
test_num_part	int(11)
answer	text

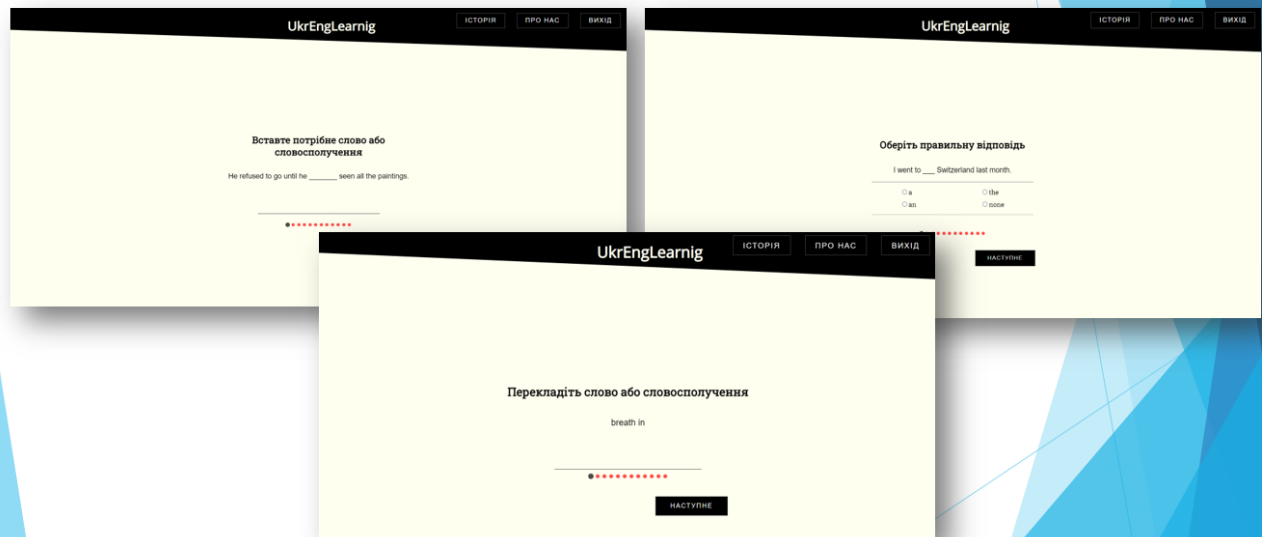
english_db topics	
id	int(11)
topic_name	text

english_db history	
id	int(11)
topic_id	int(11)
user_id	int(11)
good_answers	int(11)

english_db users	
id	int(11)
login	varchar(255)
password	varchar(255)
email	text

10

Приклад екрану виконання завдання



11

ВИСНОВКИ

У ході виконання дипломної роботи були виконані поставлені завдання:

- Проаналізовано додатки-аналоги;
- Розглянуто технології для розробки веб-додатка ;
- Описано алгоритм його створення;
- Розроблено веб-додаток;
- Проведено тестування розробленого додатку;

12

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

- Двірний, Д.Ю. // Розробка навчального веб-додатка на основі фреймворка скриптової мови програмування JavaScript - React.JS. Збірник тез. 12.02.2021, ДУТ, м. Київ — К.: ДУТ, 2021. — С. 109.
- Двірний Д.Ю.// Актуальність розробки навчального веб-додатка на основі фреймворка скриптової мови програмування JavaScript – React.JS/ Збірник тез. 15.03.2021, ДУТ, м. Київ — К.: ДУТ, 2021.

13

ДЯКУЮ ЗА УВАГУ!