

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(освітньо-кваліфікаційний рівень)

на тему: **Розробка веб-застосунку «Englex» для вивчення
англійської мови**

Виконав: студент IV курсу, групи СНС-42

напряму підготовки (спеціальності):

122 Комп'ютерні науки

(шифр і назва напряму підготовки, спеціальності)

	<u>Свищ Т. Б.</u> (підпис)	<u>Свищ Т. Б.</u> (прізвище та ініціали)
Керівник	<u>Гром'як Р. С.</u> (підпис)	<u>Гром'як Р. С.</u> (прізвище та ініціали)
Нормоконтроль	<u>Шимчук Г.В.</u> (підпис)	<u>Шимчук Г.В.</u> (прізвище та ініціали)
Завідувач кафедри	<u>Боднарчук І.О.</u> (підпис)	<u>Боднарчук І.О.</u> (прізвище та ініціали)
Рецензент	<u>Осухівська Г.М.</u> (підпис)	<u>Осухівська Г.М.</u> (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« » 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Свищ Тетяні Богданівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-застосунку "Englex" для вивчення англійської мови

Керівник роботи Гром'як Роман Сильвестрович, к.ф.-м.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «16» березня 2022 року № 4/7-161

2. Термін подання студентом завершеної роботи 23 червня 2022р.

3. Вихідні дані до роботи Технічне завдання на розробку програмного забезпечення, мови програмування: PHP, JavaScript, MySQL, HTML

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз технічного завдання. 1.1 Найменування та область застосування. 1.2

Призначення розробки. 1.3 Вимоги до функціоналу сайту. 1.4 Вимоги до надійності

1.5 Умови до експлуатації 1.6 Вимоги до складу і параметрів технічних засобів 1.7 Вимоги

до інформаційної і програмної сумісності 1.8 Вимоги до програмної документації

1.9 Стадії та етапи розробки 1.10 Порядок контролю та прийому 1.11 Вимоги до

технічного забезпечення 1.12 Порядок тестування 2 Розробка технічного та робочого проекту

2.1 Постановка задачі. 2.2 Концептуальна модель системи 2.3 Розробка системи реєстрації та

авторизації користувача 2.4 Використання бази даних MySQL 2.5 Створення системи

коментування за допомогою сервісу Disqus 2.6 Розробка дизайну сайту 2.7 Створення

профілю

користувача 2.8 Створення системи тестування 2.9 Створення адміністративної частини

2.10 Інструкція з інсталяції та налаштування веб-серверів 2.11 Інструкція по встановленню та

налаштуванню розробленого проекту 2.12 Інструкція з розміщення сайту в Інтернеті.

3. Безпека життєдіяльності, основи охорони праці. 3.1 Вимоги до нормативно-правових

документів з охорони праці, до мікроклімату та іонного складу повітря на робочому місці

користувача ПК. 3.2 Дія електричного струму на організм людини 3.3 Організація діяльності

оператора за дисплеєм. Висновки. Перелік джерел. Додатки.

5. Перелік графічного матеріалу. (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання _____ 24 січня 2022 р. _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	24.01.2022	Виконано
2.	Підбір джерел про методики вивчення англійської мови за допомогою веб-ресурсів	04.01.2022-30.01.2022	Виконано
3.	Збір та узагальнення інформації по проекту	31.01.2022-06.02.2022	Виконано
4.	Розробка технічного та робочого проекту	07.02.2022-13.02.2022	Виконано
5.	Оформлення розділу «Аналіз технічного завдання»	14.02.2022-06.03.2022	Виконано
6.	Оформлення розділу «Розробка технічного та робочого проекту»	07.03.2022-03.04.2022	Виконано
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	04.04.2022-17.04.2022	Виконано
8.	Виконання завдання до підрозділу «Основи охорони праці»	18.04.2022-01.05.2022	Виконано
9.	Оформлення кваліфікаційної роботи	02.05.2022-15.05.2022	Виконано
10.	Нормоконтроль	16.05.2022-22.05.2022	Виконано
11.	Перевірка на плагіат	09.06.2022	Виконано
12.	Попередній захист кваліфікаційної роботи	10.06.2022	Виконано
13.	Захист кваліфікаційної роботи	23.06.2022	

Студент

_____ (підпис)

Свищ Т.Б.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Гром'як Р.С.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Розробка веб-застосунку “Englex” для вивчення англійської мови // Кваліфікаційна робота освітнього рівня «Бакалавр» // Свищ Тетяна Богданівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних наук, група СНс-42 // Тернопіль, 2022 // С. 69, рис. – 30 , табл. – 2 , кресл. – , додат. – 15, бібліогр. – 2 .

Ключові слова: розробка, сайт, веб-ресурс, навчання, англійська, інтерфейс, документація.

Тема кваліфікаційної роботи: Розробка сайту для вивчення англійської мови «Englex».

Метою даної роботи є розробка веб-застосунку для вивчення або покращення знань англійської мови за допомогою різних навчальних засобів.

В першій частині описується аналітичний огляд існуючих рішень та аналіз технічного завдання.

У другому розділі міститься матеріал по вибору тестової платформи для створюваної програми, огляд структури та вмісту програми, а також описані інструкції з інсталяції та налаштування веб-серверів та супутнього програмного забезпечення.

ANNOTATION

Development of web application "Englex" for learning English // Qualification work of the educational level "Bachelor" // Svyschch Tetiana Bogdanivna // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science Sciences, group SNs-42 // Ternopil, 2022 // C. 69, fig. - 30, table. - 2, chair. -, add. -15, bibliogr. -.

Keywords: development, site, web resource, training, English, interface, documentation.

Theme of qualification work: Development of web application "Englex" for learning English.

The purpose of this work is to develop a web application for learning or improving English language skills through various learning tools.

The first part describes the analytical review of existing solutions and analysis of the technical task.

The second section contains material on choosing a test platform for the program you are creating, an overview of the structure and content of the program, as well as instructions for installing and configuring web servers and related software.

ПЕРЕЛІК СКОРОЧЕНЬ, ОДИНИЦЬ ВИМІРЮВАНЬ І ТЕРМІНІВ

MySQL - вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних.

HTTP - прикладний протокол передачі даних у мережі

Хостинг - послуга надавання дискового простору, підключення до мережі та інших ресурсів для розміщення фізичної інформації на сервері,

UI – інтерфейс користувача

API – прикладний програмний інтерфейс

ID - унікальна ознака об'єкта, що дозволяє відрізнити його від інших об'єктів, тобто ідентифікувати.

ПЗ – програмне забезпечення.

Хешування - перетворення вхідного масиву даних довільної довжини у вихідний бітовий рядок фіксованої довжини.

Сервер - це комп'ютер, пристрій або програма, яка призначена для управління мережевими ресурсами

ЗМІСТ

ВСТУП.....
РОЗДІЛ 1. АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....
1.1 Найменування та область застосування
1.2 Призначення розробки
1.3 Вимоги до функціоналу сайту.....
1.4 Вимоги до надійності
1.5 Умови до експлуатації.....
1.6 Вимоги до складу і параметрів технічних засобів.....
1.7 Вимоги до інформаційної і програмної сумісності.....
1.8 Вимоги до програмної документації.....
1.9 Стадії та етапи розробки
1.10 Порядок контролю та прийому
1.11 Вимоги до технічного забезпечення
1.12 Порядок тестування сайту
РОЗДІЛ 2. РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ
2.1 Постановка задачі
2.2 Концептуальна модель системи
2.3 Розробка системи реєстрації та авторизації користувача.....
2.4 Використання бази даних MySQL
2.5 Створення системи коментування за допомогою сервісу Disqus
2.6 Розробка дизайну сайту.....
2.7 Створення профілю користувача
2.8 Створення системи тестування
2.9 Розробка адміністративної частини
2.10 Переваги мови програмування PHP при розробці сайту
2.11 Інструкція з інсталяції та налаштування веб-серверів та супутнього програмного забезпечення.....
2.12 Інструкція по встановленню та налаштуванню розробленого проекту
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ
3.1 Вимоги до нормативно-правових документів з охорони праці до

мікроклімату та іонного складу повітря на робочому місці користувача ПК.....	
3.2 Дія електричного струму на організм людини	
3.3 Організація діяльності оператора за дисплеєм	
ВИСНОВКИ	
ПЕРЕЛІК ДЖЕРЕЛ.....	
ДОДАТКИ	

ВСТУП

Немає жодної галузі чи предмета, де англійська мова не була б важливою. У світі понад 350 мільйонів носіїв цієї мови. Англійська мова – це мова бізнесу, торгівлі, спілкування, взаємодії, візуалізації тощо.

Це найбільш розмовна мова світу. Ми можемо знайти будь-кого, хто говорить англійською в будь-якому кутку світу. Велика кількість дат, доступних в Інтернеті, яка є джерелом великої кількості інформації та знань саме цією мовою. Якщо ми хочемо отримати ці знання та інформацію, ми повинні знати англійську мову.

В більш ніж 75-ти регіонах земної кулі ця лінгвістична система є офіційною або має особливий статус. Знаючи англійську можна подорожувати світом і не боятися мовного бар'єру, а також відкрити для себе нові кар'єрні можливості.

Причини для вивчення мов різні. Однак люди сходяться на думці, що всі мови призначені для полегшення процесу комунікації.

Англійська мова важлива для доступу до світових медіа та розваг. Супутникові канали по всьому світу транслюють новини та перегляди англійською мовою. Ігри та спортивні змагання транслюються в прямому ефірі, а їхні коментарі також транслюються англійською. Ця мова поширена в подорожах і розвагах, в комп'ютерних технологіях, в медичних і фармацевтичних областях і т.д.

В інтернеті є безліч ресурсів, які допомагають людям покращити свій рівень англійської мови, отже кожен може вибрати саме той, який вважає для себе найбільш ефективним.

РОЗДІЛ 1. АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Найменування та область застосування

Найменування – розробка веб-застосунку для вивчення англійської мови «Englex».

Область застосування програми – даний веб-додаток призначений для вивчення англійської мови користувачами з різними рівнями володіння мовою.

1.2 Призначення розробки

Експлуатаційне призначення – клієнт-серверний web-додаток для вивчення або покращення знань англійської мови за допомогою різних навчальних засобів.

Функціональне призначення – сайт повинен бути легкий в експлуатації, зрозумілий для користувачів та багатофункціональний.

1.3 Вимоги до функціоналу сайту

Загальні вимоги до Web-сайту:

- здійснення операцій з базою даних через графічний модуль сайту;
- наявність системи тестування для перевірки знань;
- навігація по сайту забезпечується за допомогою текстових посилань на його розділи;
- наявність профілю користувача;
- наявність адміністративної панелі;
- наявність форми реєстрації та авторизації.

Система повинна забезпечувати можливість виконання наступних функцій:

- Вивід статей на англійській мові.
- Вивід обраної категорії англійських слів для їх вивчення.
- Збереження реєстраційних даних користувача.
- Введення та редагування інформації про користувача.
- Збереження результатів тесту в базі даних.

- Вивід інформації про користувача в його особистому профілі.

Вимоги до структури сайту.

Структура Web-сайту, що розробляється, повинна передбачати подальший розвиток за допомогою додавання нових розділів, а також можливість їх зміни, впровадження додаткових блоків і модулів.

Вимоги до навігації сайту.

Графічний інтерфейс користувача повинен мати можливість забезпечувати наочне, інтуїтивно зрозуміле уявлення структури розміщеної на ньому інформації, забезпечувати швидкий перехід до вибраного розділу сайту.

Меню навігації має бути зрозумілим для користувача, а саме: гіперпосилання на сторінки повинні мати зрозумілі заголовки, щоб користувач міг без перешкод перейти до потрібної йому сторінки сайту. Для здійснення переходу між сторінками повинен бути реалізований блок, що складається зі списку гіперпосилань.

1.4 Вимоги до надійності

Надійне функціонування сайту повинне забезпечуватись наявністю блоку перевірки правильності введених даних. Це збереже від помилок, що виникають при виконанні рутинної роботи користувачем.

1.5 Умови до експлуатації

Сайт не потребує спеціальних навиків, окрім базових навиків користувача персонального комп'ютера.

1.6 Вимоги до складу і параметрів технічних засобів

В склад технічних засобів повинні входити:

- монітор (діагоналю не менше 15");
- клавіатура;
- мишка;

- персональний комп'ютер із такими мінімальними характеристиками:
- процесор – Intel Pentium 4;
- оперативна пам'ять - 1 ГБ;
- будь-який веб-браузер.

1.7 Вимоги до інформаційної і програмної сумісності

Вихідні коди сайту повинні бути реалізовані мовами програмування PHP та JavaScript, мовою розмітки HTML та мовою стилю сторінок CSS. В якості середовища розробки сайту повинне використовуватись середовище Atom або будь-який інший текстовий редактор. Зв'язок з сервером буде встановлюватися за допомогою програмного забезпечення MAMP, доступного для операційних систем Windows версії не нижче Windows 7 та MacOS мінімальної версії MacOSX.

1.8 Вимоги до програмної документації

По закінченню розробки сайту потрібно підготувати таку документацію:

- загальні відомості про можливості сайту;
- опис програми - інформація про логічну структуру та функціонування програми;
- технічне завдання - документ, в якому викладаються призначення і область застосування програми, вимоги до програмного виробу, стадії і терміни розробки, види випробувань;
- пояснювальна записка - обґрунтування прийнятих і застосованих технічних і техніко-економічних рішень, схеми та опис алгоритмів, загальний опис роботи програмного виробу.

1.9 Стадії та етапи розробки

Стадії розробки сайту [12]:

- Визначення цілей. На цій стадії проводиться підготовка до створення сайту та визначається мета розробки даного проекту.

- Формування технічного завдання. Потрібно створити план, по якому буде розроблятися сайт.
- Створення макету дизайну сайту. Одним з найважливіших етапів створення сайту є розробка зовнішньої оболонки, яка має бути максимально привабливою для користувача.
- Верстка сайту. За допомогою мови розмітки HTML потрібно перевести готовий дизайн в робочий проект.
- Проектування. Потрібно визначити спосіб взаємодії сайту з базою даних SQL. Реалізувати проект за допомогою мов програмування, наповнити сайт контентом та об'єднати всі файли для створення цілісного програмного продукту.
- Тестування. Метою даного етапу є виявлення помилок, які необхідно усунути.

1.10 Порядок контролю та прийому

Прийом розробленого програмного забезпечення повинен відбуватися на об'єкті Замовника в терміни, які зазначені в індивідуальному завданні. Для прийому роботи Виконавець повинен представити:

- діючий сайт, який повністю відповідає даному технічному завданню;
- вихідний програмний код, записаний разом із програмою на оптичний носій інформації.

Прийом програмного забезпечення повинен відбуватися перед комісією з двох чоловік (один з яких – Замовник) у такій послідовності:

- доповідь Виконавця про виконану роботу;
- демонстрація Виконавцем роботи програми;
- контрольні випробовування роботи програми;
- відповіді на запитання і зауваження комісії.

1.11 Вимоги до технічного забезпечення

Для коректної роботи програмного продукту необхідно мати будь-який пристрій, через який можна здійснювати підключення до мережі Інтернет.

Даний веб-сайт здатний працювати на бюджетних пристроях із невеликою потужністю.

Мінімальні вимоги до програмного та апаратного забезпечення для сервера:

- Операційна система: Windows 7, Mac OS X або Linux (Debian, SuSe, RedHat, Gentoo і ін.).
- Веб-сервер: Apache.
- Мови програмування:
 - Java Script.
 - PHP >= 5.3 (dom, xsl, xml, сесія, gd, iconv, eaccelerator).
 - СУБД: MySQL >= 5.1.

1.12 Порядок тестування сайту

Тестування безпеки

Тестування безпеки виконується для перевірки системи захисту даних і підтримки функціоналу.

Можна імітувати атаку шкідливого джерела, щоб оцінити рівень безпеки сайту (тестування проникнення).

Інший тип тестування безпеки, тестування уразливості, дозволяє оцінити загальний обсяг залучених ризиків.

Деякі перевірки для тестування безпеки:

- забезпечити неможливість несанкціонованого доступу до захищених сторінок;
- автоматичне припинення перевірки сеансів після тривалого простою користувача;
- тестування функцій безпеки SSL;
- всі спроби злому, повідомлення про помилки і т. П. Повинні реєструватися та зберігатися в окремому файлі для подальшого аналізу;
- перевірка роботи captcha з допомогою автоматичних скриптів;
- переконатися, що файли з обмеженим доступом не завантажуються без відповідного дозволу;

- переконатися, що при введенні неправильного пароля або імені користувача немає можливості входу в систему.

Функціональне тестування сайту.

Функціональне тестування спрямоване на те, щоб кожна функція веб-сайту працювала відповідно до вимог специфікації. Тестування функціональності веб-сайту показує «Що робить система».

Тестування посилань.

Потрібно перевірити:

- вихідні посилання;
- коректність внутрішніх посилань;
- посилання, які використовуються для відправки електронної пошти адмінам сайту;

- чи є сторінки, на які не вказані посилання;
- відсутність непрацюючих посилань;
- тестування форм для всіх сторінок;
- форми використовуються для інтерактивного спілкування з клієнтами. Отже, необхідно перевірити наступні моменти:

- дійсність вхідних даних;
- допустимі значення для поля даних;
- неприпустимі вхідні значення для поля даних;
- параметри форм, в яких можливе видалення або будь-яка інша модифікація даних.

HTML / CSS валідація.

- синтаксичні помилки HTML;
- переконатися, що сайт доступний для пошукових машин;
- переконатися, що веб-сторінка має точну карту сайту в форматі XML HTML.

Тестування UI (інтерфейсу користувача).

Тестування інтерфейсу користувача (UI) виконується для перевірки відповідності графічного призначеного для користувача інтерфейсу вашого сайту специфікаціям.

Ось деякі перевірки для тестування інтерфейсу веб-сайту:

- відповідність стандартам графічних інтерфейсів;
- оцінка елементів дизайну: макет, кольори, шрифти, розміри шрифтів, ярлики, текстові поля, форматування тексту, титри, кнопки, списки, значки, посилання;
- тестування з різними дозволами екрану;
- тестування локалізованих версій: точність перекладу, перевірка довжини імен елементів інтерфейсу і т. П.;
- тестування графічного інтерфейсу користувача на цільових пристроях: смартфони і планшети.

Usability тестування сайту (перевірки зручності).

Usability тестування призначене для оцінки веб-сторінки з точки зору кінцевого користувача. Це допомагає визначити відповідність продукту очікуванням користувачів, виявляє проблемні місця в інтерфейсі.

Навігаційне тестування сайту містить наступні перевірки:

- всі сторінки сайту зрозумілі і прості у використанні;
- кнопки, форми і поля зручні для використання;
- доступ до головного меню здійснюється з усіх сторінок;
- чек-лист тестування контенту;
- відсутні граматичні, орфографічні помилки;
- контент повинен бути інформативним, зрозумілим, структурованим і логічно пов'язаним;
- інструкції ясні і містять правильну інформацію.

РОЗДІЛ 2. РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ

2.1 Постановка задачі

Поставлені завдання для виконання проекту:

- виконати огляд і порівняльний аналіз сучасних методів та інструментів для розробки Web-сайтів;
- вибрати найбільш оптимальний варіант розробки сайтів;
- розробити модель сайту, його структуру;
- розробка Web-сайту з використанням сучасних програмних засобів розробки;
- розрахувати економічну ефективність від впровадження Web-сайту.

2.2 Концептуальна модель системи

На концептуальній моделі зображено, що завдяки API (програмний інтерфейс програми) [11], ми отримуємо потрібні нам дані та відображаєм їх завдяки нашому графічному модулю, який можна відстежити в Всесвітній мережі Інтернет.

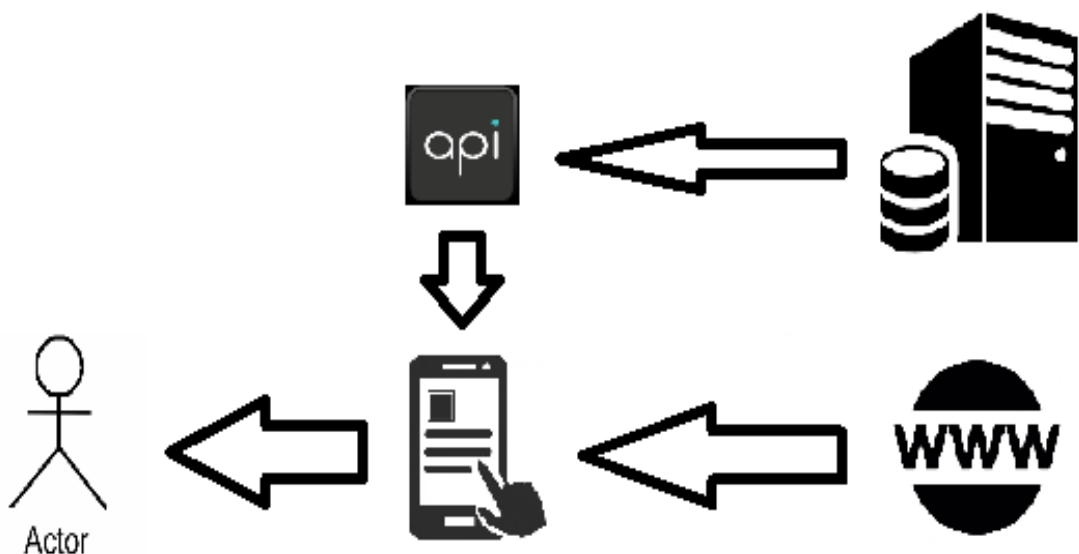


Рисунок 2.1 - Концептуальна модель системи

2.3 Розробка системи реєстрації та авторизації користувача

Розробка даної системи складається із ряду алгоритмів, які виконуються у визначеній послідовності.

Спочатку потрібно створити дві форми за допомогою мови розмітки HTML з методом POST, що дозволяє відправляти інформацію в базу даних для подальшої обробки.

Для зручного переходу з однієї форми на іншу, на верхній панелі є дві кнопки: “Реєстрація” та “Авторизація”, які показують відповідний блок. Даний код написаний на мові програмування Java Script за допомогою методів show() та hide() (див. лістинг 2.1).;

Лістинг 2.1 – Блок реєстрації та авторизації

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"
"></script>
<script>
    $(document).ready(function () {
        $("#s").hide();
        $("#aut").click(function () {
            $("#f").hide();
            $("#s").show();
        });
    });
    $(document).ready(function () {
        $("#reg").click(function () {
            $("#s").hide();
            $("#f").show();
        });
    });
</script>
```

Форма реєстрації буде містити кнопку “Зареєструватися” та такі поля (див.рисунок 2.3):

- Login – логін.
- Name – ім’я.
- Password – пароль.

В тезі <form> ми вказуємо посилання на файл Check.php, в якому підключаємо базу даних та прописуємо код для внесення реєстраційної інформації користувача в таблицю (див. лістинг 2.2).

Лістинг 2.2 – Форма реєстрації

```
<h1>Registration</h1>
<form action="php/Check.php" method="post">
    <input type="text" class="form-control" name="login"
id="login" placeholder="Enter your login"> <br>
    <input type="text" class="form-control" name="name" id="name"
placeholder="Enter your name"><br>
    <input type="password" class="form-control" name="pass"
id="pass" placeholder="Enter your password"><br>
    <button class="bt" class="btn btn-success" type="submit">
Register</button> </form> </div>
```

Надсилаємо запит для збереження даних на веб-сервері за допомогою методу `$_POST` (див. лістинг 2.3).

Лістинг 2.3 – Запит для збереження даних в СУБД

```
$login = filter_var(trim($_POST['login']), FILTER_SANITIZE_STRING);
$name = filter_var(trim($_POST['name']), FILTER_SANITIZE_STRING);
$pass = filter_var(trim($_POST['pass']), FILTER_SANITIZE_STRING);
```

Оскільки логін кожного користувача повинен бути унікальним, при реєстрації надсилаємо запит в базу даних на наявність введеного логіна, якщо такий вже існує, виводиться повідомлення про помилку (див. лістинг 2.4).

Лістинг 2.4 – Запит для перевірки логіна

```
$result = $mysql->query("SELECT * FROM `users` WHERE `login` =
'$login'"); $user = $result->fetch_assoc();
if(count($user) > 0) {
    echo "Логін існує"; exit(); } else {
    $mysql->query("INSERT INTO `users` (`login`, `pass`, `name`)
VALUES ('$login', '$pass', '$name')");}
```

Окрему увагу слід приділити паролю. Збереження паролю в базі даних в початковому вигляді є досить небезпечним, оскільки завжди є загроза проникнення в базу злоумисниками. Для уникнення такої ситуації використовується хешування, яке дає змогу зберігати пароль в базі даних зовсім в іншому вигляді, а при авторизації на сервері перевіряється відповідність паролю до хешу. Наприклад, замість паролю user12, який ввів користувач, в базі даних буде відображатися набір символів, в якому зашифрований даний пароль (див.рисунок 2.2). Таким чином персональні дані користувача будуть захищені від злому (див. лістинг 2.5).

id	login	pass	name
8	User12	06726a6b241dbda130fed9947ba22d70	User12

Рисунок 2.2 - Хеш паролю в БД

Лістинг 2.5 - хешування паролю

```
$pass = md5($pass."ghhr56btn");
//Підключення бази даних
$mysql = new mysqli('localhost', 'root', 'root', 'Register');
if ($mysql->connect_errno) {
    echo "Не удалось подключиться к MySQL: (" . $mysql->connect_errno . ") " . $mysql->connect_error;
}
//Введення даних в таблицю
$mysql->query("INSERT INTO `users` (`login`, `pass`, `name`)
VALUES('$login', '$pass', '$name')");
```

Для коректної роботи сайту, після завершення роботи з MySQL, обов'язково потрібно прописати код для закриття бази даних.

```
$mysql->close();
```

The image shows a registration form with a teal and blue geometric background. At the top, there are two buttons: "Реєстрація" (Registration) and "Авторизація" (Authorization). The main heading is "Реєстрація". Below it are three input fields: "Enter your login", "Enter your name", and "Enter your password". At the bottom, there is a "Зареєструватись" (Register) button and a link "LOG IN AS A GUEST".

Рисунок 2.3 - Форма реєстрації

Якщо за допомогою форми реєстрації ми вносили інформацію в базу даних, то в формі авторизації ми будемо надсилати запит на її перевірку (див. лістинг 2.6). Якщо такий користувач є в базі, його буде авторизовано. Ім'я даного користувача буде виведено в верхній панелі сайту (див.рисунок 2.4). Для того, щоб вийти з профілю, потрібно натиснути на посилання «LogOut».



Рисунок 2.4 - Головне меню сайту для авторизованого користувача

Форма авторизації буде містити поля для вводу логіна та паролю та кнопку “Авторизуватися” (див.рисунок 2.5).

Лістинг 2.6 - Форма авторизації

```
<h1>Authorization</h1>
<form action="php/Auth.php" method="post">
  <input type="text" class="form-control" name="login" id="login"
placeholder="Enter your login"><br>
  <input type="password" class="form-control" name="pass" id="pass"
placeholder="Enter your password"><br>
  <button class="bt" class="btn btn-success" type="submit">Log
In</button> </form>
```

Для надсилання запиту на перевірку введених даних використовується файл Auth.php (див. лістинг 2.7).

Лістинг 2.7 – Запит на перевірку даних в СУБД

```
//Auth.php <?php
$login = filter_var(trim($_POST['login']), FILTER_SANITIZE_STRING);
$pass = filter_var(trim($_POST['pass']), FILTER_SANITIZE_STRING);
$pass = md5($pass."ghhr56btn"); $mysql = new mysqli('localhost',
'root', 'root', 'Register'); if ($mysql->connect_errno) {
  echo "Не вдалося підключитись до MySQL: (" . $mysql->connect_errno . ") " . $mysql->connect_error; }
$result = $mysql->query("SELECT * FROM `users` WHERE `login` =
'$login' AND `pass` = '$pass'");
$user = $result->fetch_assoc();
if(count($user) == 0) { echo "Користувач не знайдений"; exit(); }
//Встановлення часу сесії на 2 години setcookie('user',
$user['name'], time() + 7200, "/");
$mysql->close(); header('Location: /');
```

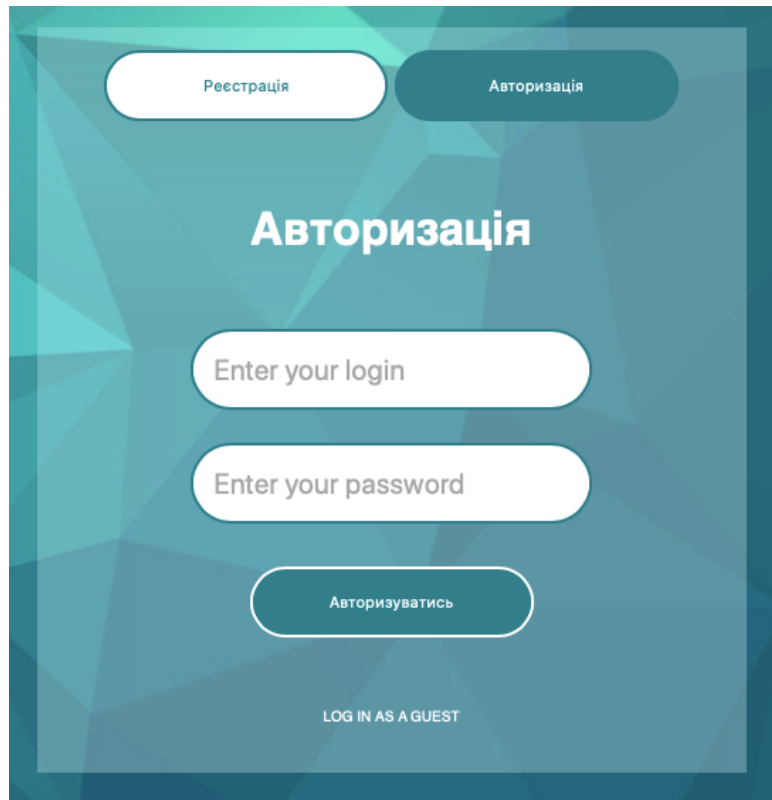


Рисунок 2.5 - Форма авторизації

Посилання “LOG IN AS A GUEST” дозволяє увійти на сайт без реєстрації. Для неавторизованого користувача головне меню буде містити поле для авторизації замість відображення імені (див.рисунок 2.6).



Рисунок 2.6 - Головне меню сайту для неавторизованого користувача

2.4 Використання бази даних MySQL

Окрім системи реєстрації та авторизації користувача, база даних MySQL використовується також для вивчення англійських слів. При натисненні на пункт «Words» на головному меню сайту, відкриється сторінка, де користувач може обрати бажаний рівень складності та теми. (див.рисунок 2.7).

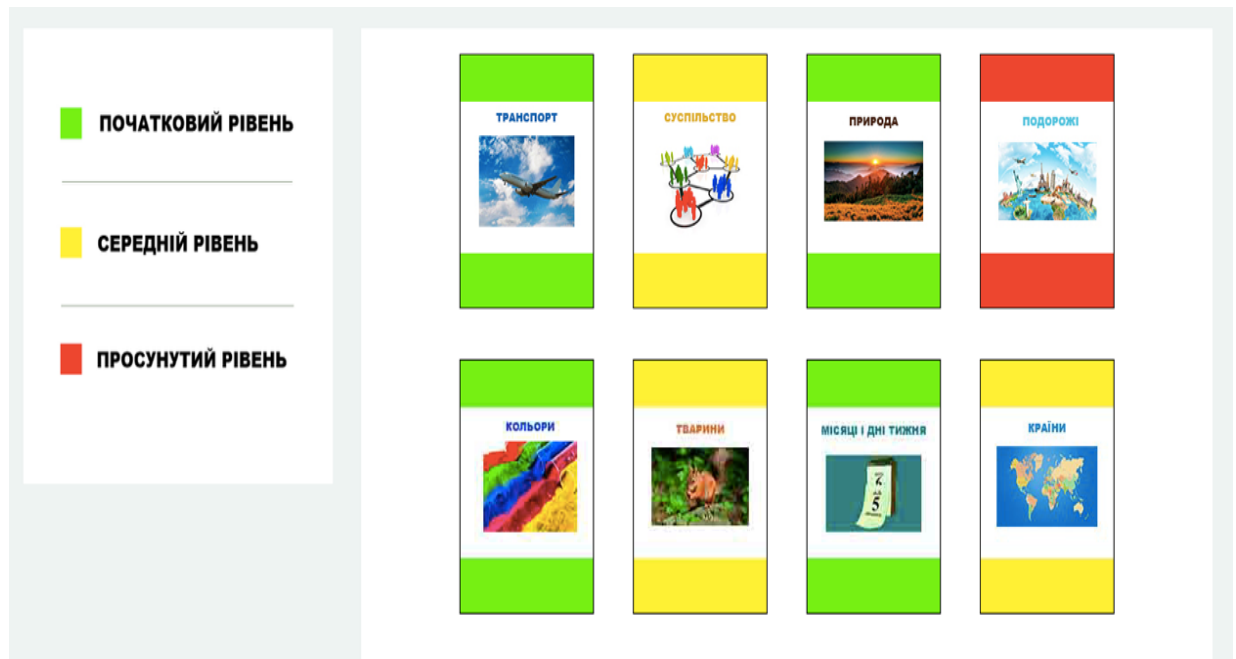


Рисунок 2.7 - Інтерфейс сторінки сайту

В базі даних «Words» створено таблицю, яка містить всі слова (див.рисунок 2.8).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id	int(11)		Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
<input type="checkbox"/>	2	Eng	varchar(30)	utf8_general_ci	Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/>	3	Ukr	varchar(30)	utf8_general_ci	Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/>	4	Theme	varchar(30)	utf8_general_ci	Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/>	5	Level	varchar(2)	utf8_general_ci	Ні	Немає			Змінити Знищити Більше

Рисунок 2.8 - Структура таблиці БД

Таблиця містить такі поля:

- id – унікальний номер елемента, присвоюємо йому автоінкрементацію, для автоматичного введення порядкового номера;
- eng – слово на англійській мові;
- ukr – переклад слова на українську мову;
- theme – тема, до якої відноситься дане слово;
- level – вказує на рівень складності слова.

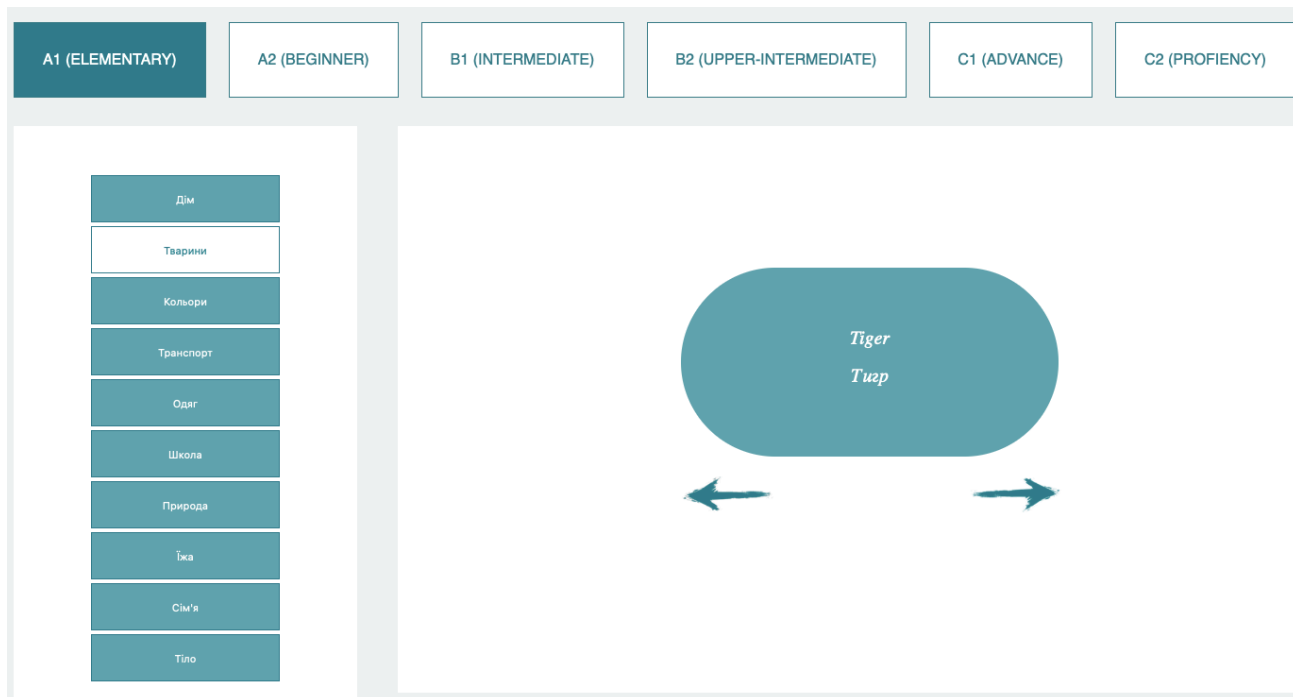


Рисунок 2.9 - Вивід результату

Після натиснення на одну з карток отримали такий результат (див.рисунок 2.9).

2.5 Створення системи коментування за допомогою сервісу Disqus

Для того щоб мати змогу залишати відгуки чи коментарі на сайті потрібна система коментування. На даному сайті для цього існує окремий пункт меню «English Chat», де користувачі можуть покращити свій рівень англійської завдяки живому спілкуванню з іншими людьми. Найкращим варіантом серед всіх пропозицій в інтернеті є сервіс Disqus. Він має ряд переваг в порівнянні з аналогічними сервісами:

- зручний інтерфейс (див. рисунок 2.10);
- надійність;
- можливість відповідати на чужі коментарі;
- редагування коментарів;
- підтримка багатьох мов;
- можливість поділитися коментарем в соціальних мережах Twitter та Facebook;
- підрахунок кількості коментарів;

- сортування коментарів за різними параметрами;
- зручне відстеження та керування коментарями на сторінці модератора на сайті Disqus;
- можливість позначити повідомлення як спам;
- блокування користувача при необхідності;
- можливість надсилати фотографії та змінити стиль шрифту;
- можливість надіслати повідомлення, яке користувачі зможуть прочитати тільки при наведенні на нього курсором миші. Іноді ця функція є дуже корисною, наприклад, щоб уникнути спойлерів (див.рисунок 2.12).

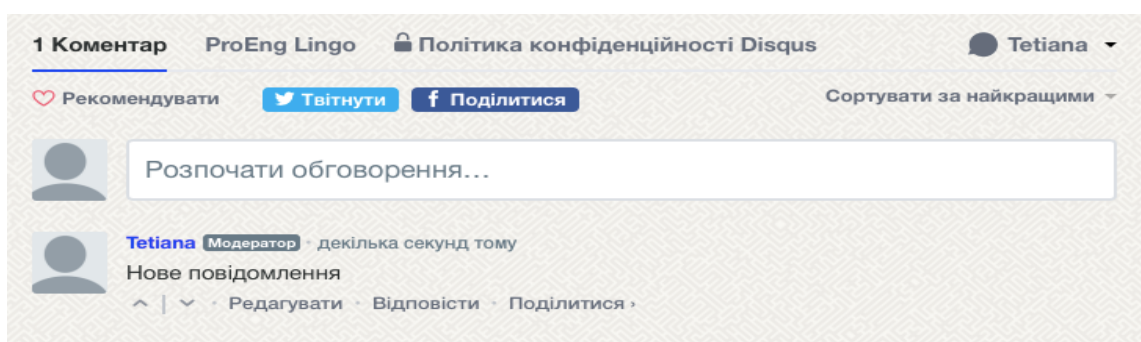


Рисунок 2.11 - Інтерфейс системи

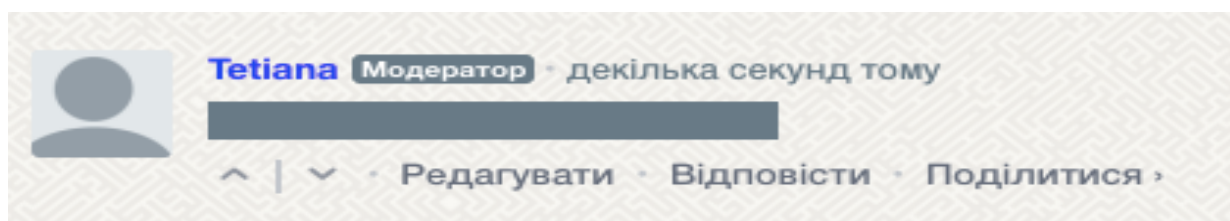


Рисунок 2.12 - Приховане повідомлення

Для того щоб мати змогу писати коментарі потрібно увійти за допомогою однієї з таких систем: Google, Facebook, Twitter, або ж зареєструватися на сайті Disqus (див. рисунок 2.13).

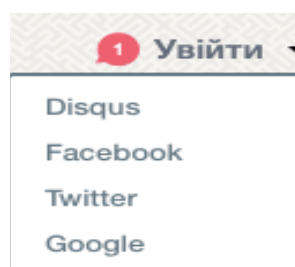


Рисунок 2.13 - Вхід в систему

Даний сервіс використовується на більшості сучасних сайтів, оскільки він здатен виконувати необхідні функції як для коментарів чи відгуків, так і для чатування.

2.6 Розробка дизайну сайту

Головна сторінка будь-якого сайту повинна максимально інформативно і в стислому обсязі відображати необхідну користувачу інформацію про сайт (див рисунок 2.15).



Рисунок 2.14 - Футер сайту

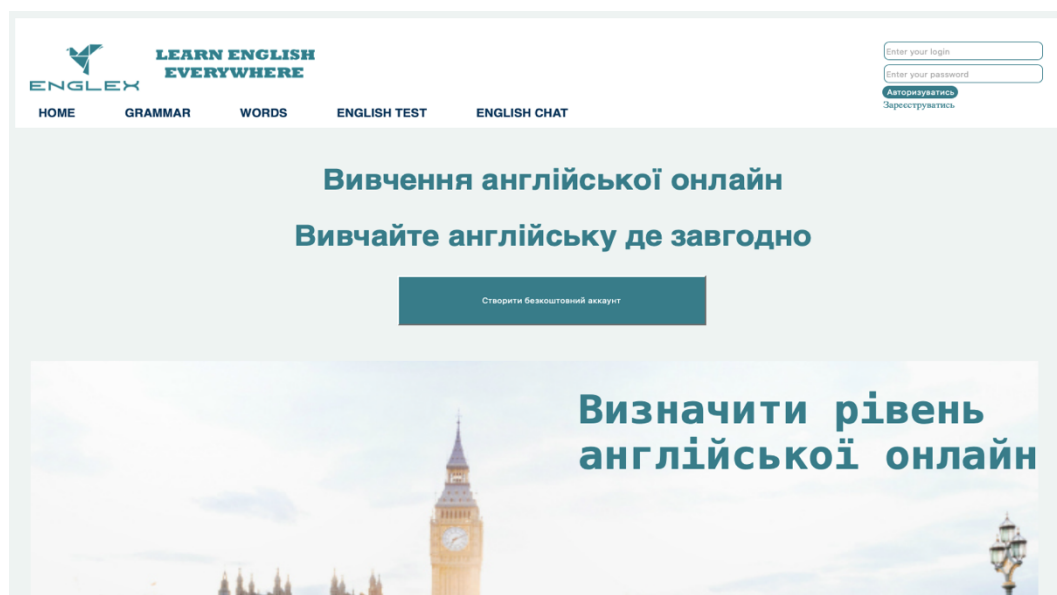


Рисунок 2.15 - Головна сторінка сайту

На головній сторінці необхідно помістити логотип сайту, головне меню сайту (для навігації по його структурі), форму аутентифікації (входу зареєстрованих користувачів), реєстраційне посилання (реєстрація нових

клієнтів), список пропозицій та футер (див.рисунок 2.14).

2.7 Створення профілю користувача

Для зручного перегляду та редагування інформації створено профіль користувача (див. рисунок 2.16), де можна змінити пароль, ввести додаткову інформацію про себе або побачити свій рівень англійської.

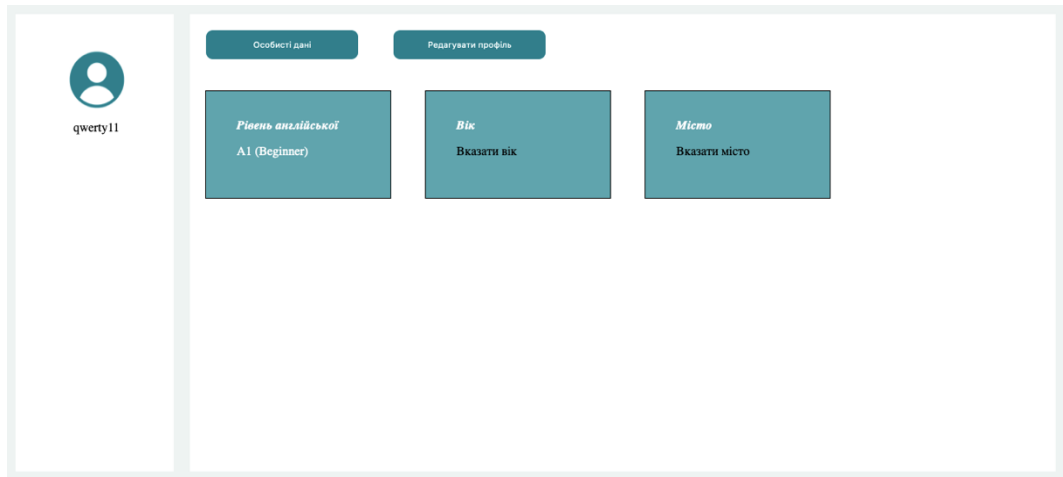


Рисунок 2.16 - Перегляд інформації

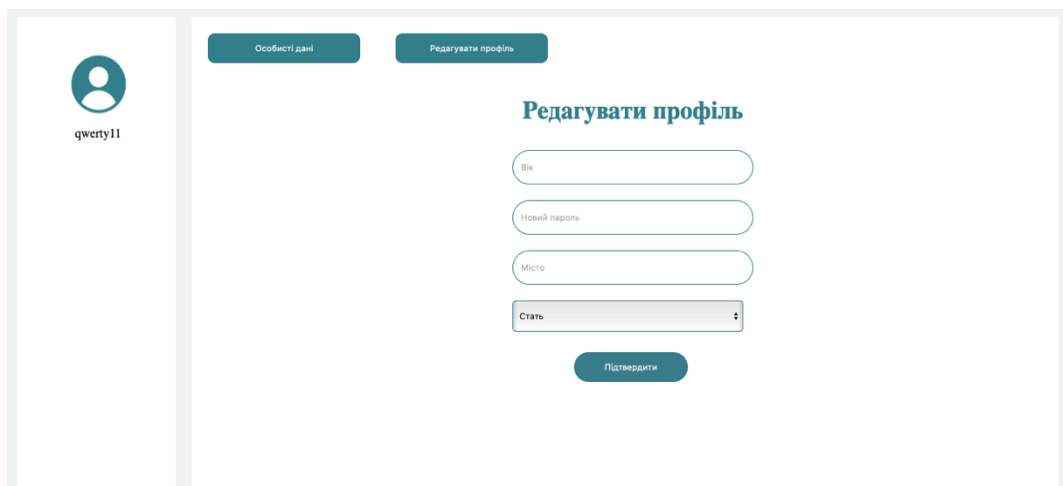



Рисунок 2.17 - Редагування інформації

За допомогою запитів MySQL можна як внести або оновити інформацію в базі даних, так і відобразити її на сторінці профілю.

2.8 Створення системи тестування

Для перевірки отриманих знань використовується система тестування. В даному випадку тест використовується для визначення рівня англійської мови (див. рисунок 2.18). Для написання коду використовується мова програмування JavaScript, а за допомогою PHP результати тесту відправляються на сервер та зберігаються в базі даних [14].



The image shows a screenshot of a web-based English level test interface. The title is "Тест на рівень англійської". It contains four multiple-choice questions:

1. What does your husband do?
 He is feeding the dog.
 He is a doctor.
 Yes, he does.
 Yes, he is.
2. Yesterday I a bird.
 saw
 sawed
 see
 seed
3. Знайдіть неправильне дієслово
 to play
 to smile
 to laugh
 to see
4. Знайдіть помилку в трьох формах дієслова
 teach – taught – taught
 catch – caught – caught
 bring – braught – braught
 seek – sought – sought

Рисунок 2.18 - Тест на рівень англійської

Після проходження тесту користувач бачить свій рівень, який буде відображатися також і в особистому профілі користувача.

2.9 Розробка адміністративної частини

Для доступу до панелі адміністратора потрібно перейти за посиланням, яке знаходиться на сторінці авторизації та реєстрації (див. рисунок 2.19), та увійти за допомогою логіна та паролю, які зберігаються в таблиці "Admin" (див. рисунок 2.20).

Рисунок 2.19 - Форма входу для адміністратора

	id	login	password
<input type="checkbox"/> ✎ Редагувати 📄 Копіювати 🗑 Видалити	1	admin	admin2458

Рисунок 2.20 - Дані адміністратора в БД

Рисунок 2.21 - Панель адміністратора

При успішному вході, адміністратор потрапляє на сторінку, де може додати нові слова для вивчення (див. рисунок 2.21).

2.10 Переваги мови програмування PHP при розробці сайту

PHP – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів [13].

В області веб-програмування, зокрема серверної частини, PHP - один з

популярних сценарних мов (разом з JSP, Perl і мовами, використовуваними в ASP.NET).

Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів і додаткових модулів для розробки веб-додатків. Основні з них:

- автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;
- взаємодія з великою кількістю різних систем управління базами даних через додаткові модулі (MySQL, SQLite, Oracle, Microsoft SQL Server, та інші);
- автоматизована відправка HTTP-заголовків;
- робота з HTTP-авторизацією;
- робота з cookies і сесіями.

В даний час PHP використовується сотнями тисяч розробників. До найбільших сайтів, які використовують PHP, відносяться Facebook, Wikipedia та ін.

2.11 Інструкція з інсталяції та налаштування веб-серверів та супутнього програмного забезпечення

Для розробки сайту “Englex” використовується локальний сервер MAMP. Це безкоштовний і простий в установці додаток, який доступний на операційних системах MacOS або Windows. MAMP розшифровується як MacOS + Apache + MySQL + PHP [10]. Встановити додаток можна з офіційного сайту (див. рисунок 2.22).

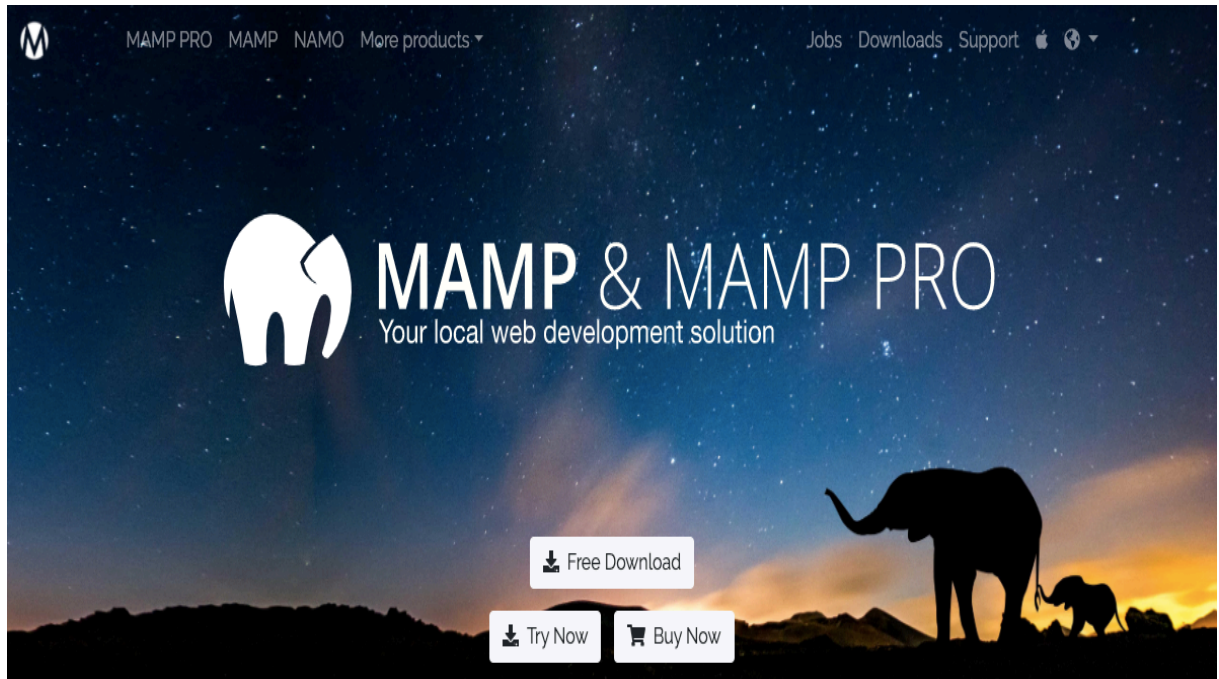


Рисунок 2.22 - Головна сторінка сайту

Далі потрібно перейти в розділ Downloads і вибрати власну операційну систему, скачування пакета почнеться відразу після переходу за посиланням.



Рисунок 2.23 - Вікно завантаження

Після того як скачування завершиться, потрібно запустити завантажений файл щоб розпочати процес установки.

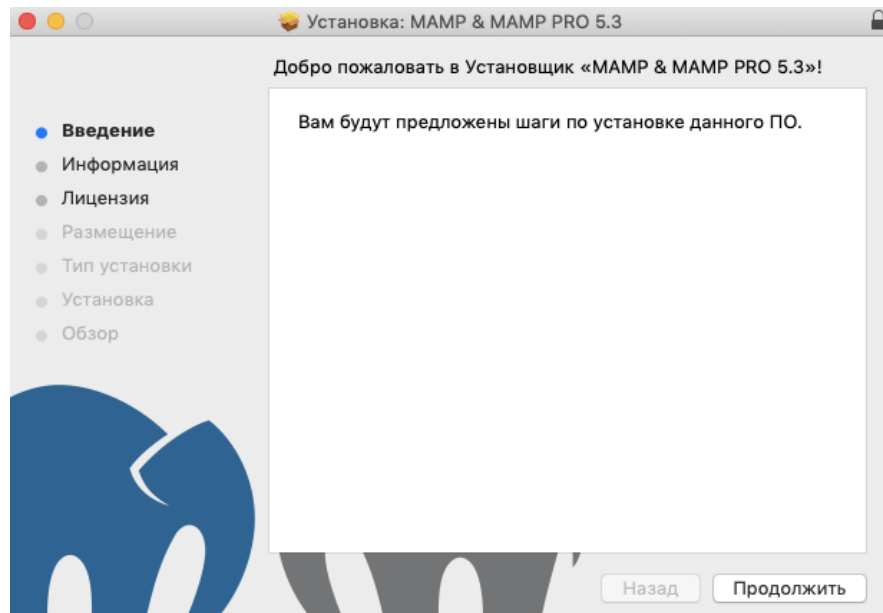


Рисунок 2.24 - Програма установки

В процесі установки все стандартно: потрібно підтвердити ознайомлення з ліцензією і кілька разів натиснути кнопку "Продовжити". Але є один крок, де потрібно вибрати версію MAMP: PRO або стандартна. Справа в тому, що версія PRO платна, але зазвичай в більшості випадків достатньо безкоштовної версії.

Отже, на етапі, де буде запропоновано вибрати тип установки, потрібно натиснути на кнопку "Налаштувати" (див. рисунок 2.25).

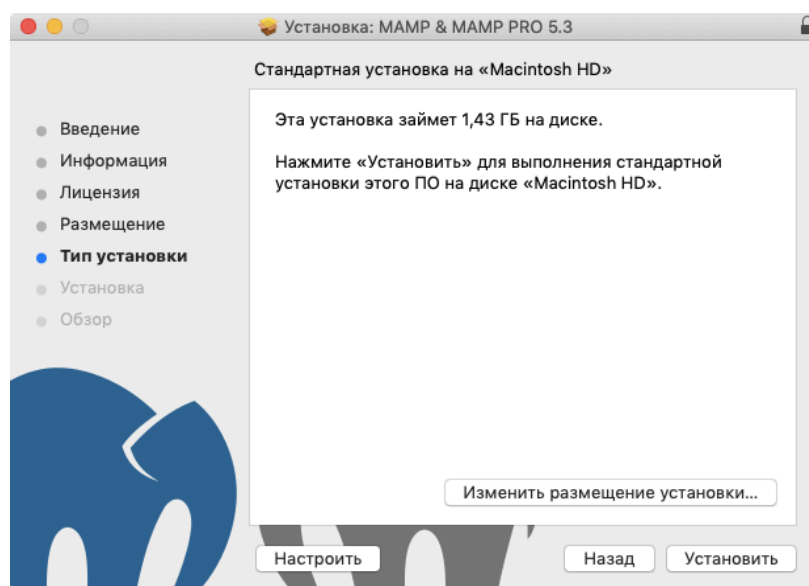


Рисунок 2.25 - Налаштування типу установки

У вікні, що з'явилося, потрібно зняти галочку з пакета MAMP PRO (див. рисунок 2.26).

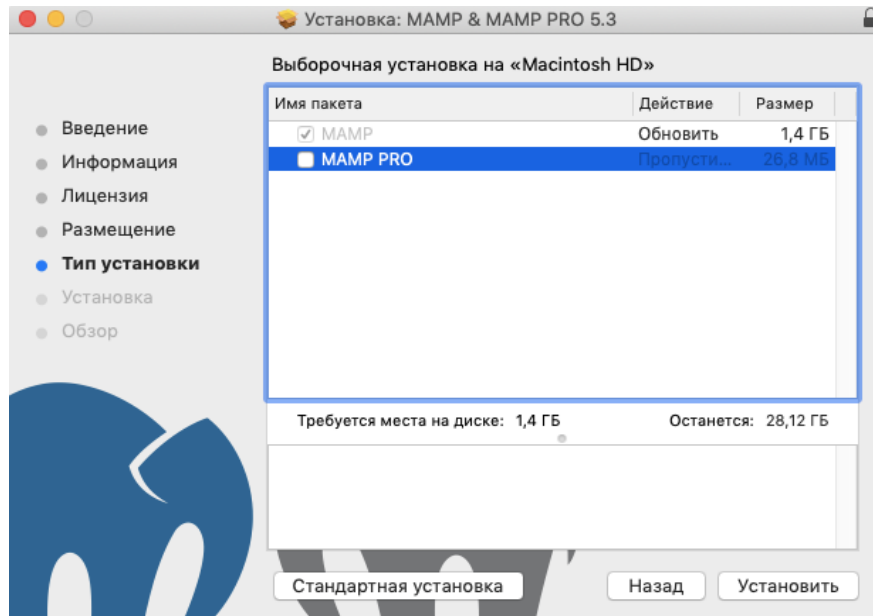


Рисунок 2.26 - Вибір програми яку необхідно встановити

Далі потрібно натиснути кнопку "Встановити", це запустить програму установки, яка скопіює потрібні файли в директорію "Програми".

Для того, щоб запустити MAMP потрібно клікнути на іконці MAMP. У вікні, нас цікавлять лише дві кнопки: "Open WebStart page" і "Start servers" (див. рисунок 2.27).

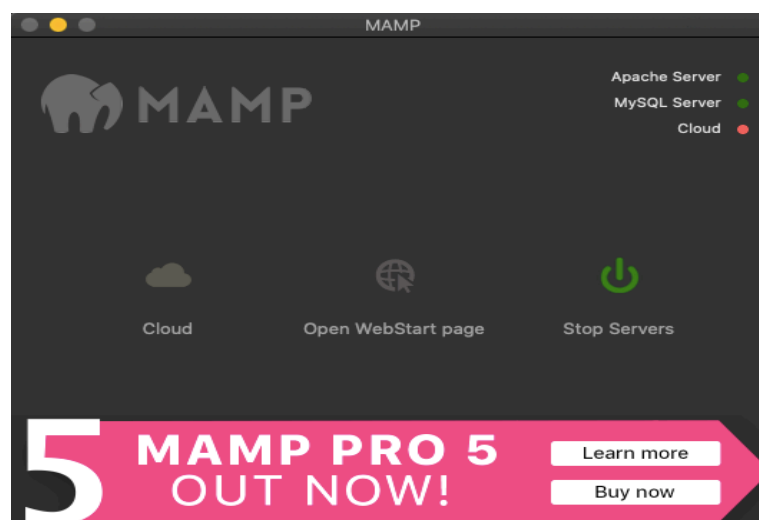


Рисунок 2.27 - Головне вікно програми

Перша відкриває допоміжну сторінку, на якій можна подивитися інформацію по конфігурації PHP на вашому локальному сервері, а також перейти на сторінку управління базами даних PhpMyAdmin (яка теж входить в стандартну

установку MAMP).

Друга кнопка запускається і зупиняє роботу сервера. Зазвичай сервер Apache автоматично запускається при запуску програми MAMP і відключається при закритті програми.

Після запуску сервера, PHP файли будуть доступні за адресою <http://localhost:8888>.

2.12 Інструкція по встановленню та налаштуванню розробленого проекту

Для перегляду сайту спочатку потрібно натиснути на кнопку "Open WebStart page", після чого відкриється вікно в браузері, де потрібно перейти по посиланню «MY WEBSITE» (див. рисунок 2.28).



Рисунок 2.28 Головне вікно

Після цього відкриється вікно з папкою проекту, де зберігаються всі файли.

Для доступу до бази даних на головній сторінці потрібно знайти посилання на PhpMyAdmin (див. рисунок 2.29).

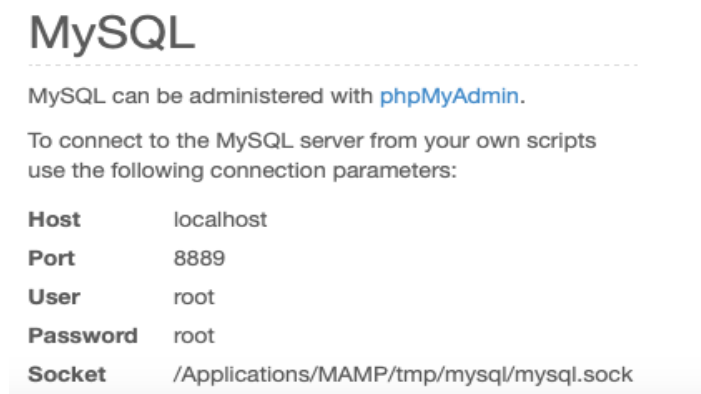


Рисунок 2.29 Посилання на PhpMyAdmin

Після переходу по даному посиланню з'явиться головна сторінка PhpMyAdmin, де можна здійснювати операції з базами даних (див. рисунок 2.30).

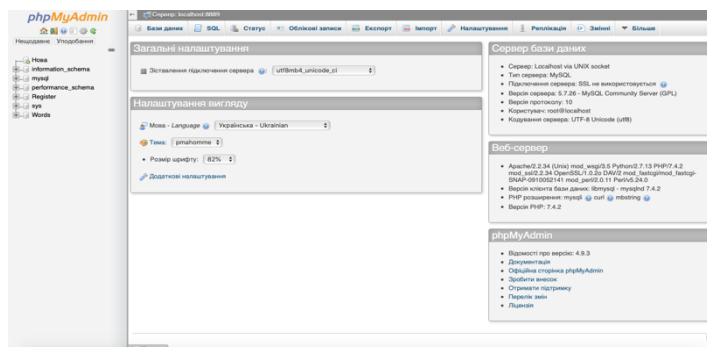


Рисунок 2.30 Головна сторінка PhpMyAdmin

Після налаштування веб-сервера все готове до роботи з сайтом.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Вимоги до нормативно-правових документів з охорони праці, до мікроклімату та іонного складу повітря на робочому місці користувача ПК

При розробці сайту “Englex” враховано вимоги охорони праці зокрема, Закон України «Про охорону праці» [1], а також інші підзаконні нормативно-правові акти. Безпека праці під час експлуатації електронно-обчислювальних машин регламентована ДНАОП 0.00-1.31-99 [2].

Робочі місця працівників обладнані персональними комп'ютерами та відповідають вимогам «Правил охорони праці під час експлуатації електронно-обчислювальних машин» [1]. Правила поширюються на всіх суб'єктів господарювання незалежно від форм власності, які у своїй діяльності здійснюють роботу, пов'язану з персональними комп'ютерами, у тому числі на тих, які мають робочі місця, обладнані персональними комп'ютерами і периферійними пристроями.

Вимоги щодо рівня іонізації повітря на робочих місцях користувачів ВДТ ПК встановлені “Санітарно-гігієнічними нормами допустимих рівней іонізації повітря виробничих і громадських будівель” №2152-80, ДСан ПіН 3.3.2.007-98, ДНАОП 0.00-1.31-99 [2] і представлені в таблиці 1.

Таблиця 3.1 - Рівні іонізації повітря на робочих місцях користувачів ЕОМ та ПЕОМ

Відеотермінали ЕОМ	Кількість іонів в 1см ³ повітря		
	мінімальна	оптимальна	Максимально допустима
Загального користування	400	1500-3000	50000
Персонального користування	600	3000-5000	50000

Для забезпечення нормованого мікроклімату та рівня іонізації повітря на робочих місцях користувачів рекомендується застосовувати припливно-витяжну вентиляцію чи систему кондиціонування повітря, прилади зволоження та/або установки генерації негативних іонів (аероіонізатори).

Знизити деіонізацію повітря в зоні дихання користувача ПК дозволяє також встановлений на монітор захисний заземлений екран.

Сучасні модифіковані аероіонізатори дозволяють одночасно очищувати і іонізувати повітря негативними іонами.

Вимоги до вмісту шкідливих речовин у повітрі робочої зони, відповідно до ДСН 3.3.6.042-99:

- концентрація озону – не більше 0,1 мг/м³;
- вміст оксидів азоту – не більше 5 мг/м³;
- вміст пилу – не більше 4 мг/м³.

Разом з тим недопустима вологість повітря більше 75%. На робочих місцях користувачів ПК параметри мікроклімату мають відповідати вимогам ДСН 3.3.6.042-99 “Санітарні норми мікроклімату виробничих приміщень [4], ДСан Пін 3.3.2-007-98 [3], ДНАОП 0.00-1.31-99 [2]. Нормативні параметри мікроклімату для приміщень з ПЕОМ зазначені в таблиці 2.

Таблиця 3.2 - Норми мікроклімату для приміщень з ВДТ ЕОМ та ПЕОМ

Пора року	Категорія робіт згідно з ГОСТ 12.1.005-88	Температура повітря, °С	Відносна вологість повітря, %	Швидкість руху повітря м/с
		Оптимальна	Оптимальна	Оптимальна
Холодна	Легка - 1а	22-24	40-60	0,1
	Легка - 1 б	21-23	40-60	0,1
Тепла	Легка - 1а	23-25	40-60	0,1
	Легка - 1 б	22-24	40-60	0,2

До категорії 1а належать роботи, що виконуються сидячи і не потребують фізичного напруження, при яких витрата енергії дорівнює 105-140Вт (90-120 ккал/год.). [4].

До категорії 1б належать роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням, при яких витрата енергії дорівнює 141-175 Вт (121-150 ккал/год.). [4].

3.2 Дія електричного струму на організм людини

Електричний струм – це впорядкований рух електричних зарядів. Сила струму в ділянці кола прямо пропорційна різниці потенціалів, тобто напрузі на кінцях ділянки і обернено пропорційна опору ділянки кола.

Доторкнувшись до провідника, що знаходиться під напругою, людина включає себе в електричний ланцюг, якщо він погано ізолюваний від землі або відразу дотикається до об'єкту з іншим значенням потенціалу. У цьому випадку через тіло людини проходить електричний струм.

Електричний струм порушує нормальні біоструми, спричиняє їх параліч. Має місце місцеве пошкодження тканин тіла та вплив на нервову систему. Під впливом електричного струму відбувається пробиття шкіряного покриву і різке зниження електричного опору тіла. Небезпечною може видатись низька напруга, а вплив високої може завершитися сприятливо. У промисловості найчастіші випадки ураження електричним струмом напругою 127, 220 та 380 В. Електричний струм може проходити через тіло людини в тому випадку, якщо людина включилася у замкнене електричне коло (паралельне вмикання) або замкнула це коло (послідовне вмикання).

Виділяють наступні основні причини уражень електрострумом:

- дотик до струмопровідних або неструмопровідних частин, що опинились під напругою;
- користування несправним електрообладнанням, електроінструментом, вимірювальними приладами, лампами і побутовими електроприладами.

Рефлекторна реакція організму на дію електричного струму зі сторони центральної і периферійної нервової систем призводить до порушення нормального ритму роботи серця, що викликає фібриляцію серця, внаслідок чого кровообіг припиняється.

Є два види ураження електричним струмом: місцеві електричні травми та електричний удар. Подекуди мають місце обидва види одночасно.

Місцеві електричні травми - це травми, викликані дією електричного струму та електричної дуги: електричні опіки, електричні знаки, металізація шкіри, електроофтальмія та механічні пошкодження.

Електричний удар - це збудження живих тканин організму електричним струмом, що проходить через нього. Воно супроводжується судомними скороченнями м'язів, у тому числі м'язів серця і легенів, внаслідок чого може статися порушення або припинення діяльності важливих систем організму людини, кровообігу і дихання, а також викликає клінічну смерть або електричний шок.

Тяжкість наслідків дії на людину електричного струму визначається величиною, родом і частотою струму, електричним опором тіла людини, тривалістю дії, напрямком проходження струму через тіло, індивідуальними властивостями людини, схемою доторкання її до ланцюга струму та умовами середовища.

Навколишнє середовище значною мірою визначає наслідок дії електричного струму на людину. Опір перегрітого організму знижується, тому з підвищенням температури тяжкість ураження струмом підвищується. Небезпечність ураження електричним струмом зростає з підвищенням вологості і забрудненості повітря.

3.3 Організація діяльності оператора за дисплеєм

Правила поширюються на умови й організацію праці при роботі з візуальними дисплейними терміналами (ВДТ) усіх типів вітчизняного та зарубіжного виробництва на основі електронно-променевих трубок (ЕПТ), що використовуються в електронно-обчислювальних машинах (ЕОМ) колективного використання та персональних ЕОМ (ПЕОМ). [3]

При розміщенні робочих столів з персональними комп'ютерами слід дотримувати:

- відстань між бічними поверхнями персональних комп'ютерів - 1,2 м;
- відстань від тильної поверхні одного персонального комп'ютера до екрана іншого - 2,5 м

За потреби особливої концентрації уваги під час виконання робіт суміжні робочі місця операторів необхідно відділяти одне від одного перегородками висотою 1,5 – 2 м.

Конструкція робочого місця користувача персонального комп'ютера забезпечує підтримання оптимальної робочої пози офісного працівника. Конструкція робочого столу відповідає сучасним вимогам ергономіки і забезпечує оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого столу регулюється в межах 680-800 мм, а ширина і глибина – забезпечує можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600-1400 мм, глибина – 800-1000 мм).

Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600 мм, завширшки не менше ніж 500 мм, завглибшки (на рівні колін) не менше ніж 450 мм, на рівні простягнутої ноги не менше ніж 650 мм. Відстань від спинки до переднього краю сидіння має регулюватися в межах 260-400 мм. Для зниження статичного напруження м'язів верхніх кінцівок слід використовувати стаціонарні або змінні підлокітники завдовжки не менше ніж 250 мм, завширшки 50-70 мм, що регулюються за висотою над сидінням у межах 230-260 мм і відстанню між підлокітниками в межах 350-500 мм.

Робочі місця слід розташовувати відносно світових прорізів так, щоб природне світло падало переважно з лівого боку. Монітор має розташовуватися на оптимальній відстані від очей користувача, що становить 600-700 мм, але не ближче ніж за 600 мм з урахуванням розміру літерно-цифрових знаків і символів. Розташування екрана монітору має забезпечувати зручність зорового спостереження у вертикальній площині під кутом $+30^\circ$ до нормальної лінії погляду працівника. Клавіатуру слід розташовувати на поверхні столу на відстані 100-300 мм від краю, звернутого до працюючого. У конструкції клавіатури має передбачатися опорний пристрій (виготовлений із матеріалу з високим коефіцієнтом тертя, що перешкоджає мимовільному її зсуву), який дає змогу змінювати кут нахилу поверхні клавіатури у межах $5-15^\circ$. Висота середнього рядка клавіш має не перевищувати 30 мм.

При організації робочих місць операторів електромережу штепсельних розеток для живлення персональних комп'ютерів, периферійних пристроїв і у центрі приміщення прокладене у каналах або під знімною підлогою в металевих трубах або гнучких металевих рукавах.

ВИСНОВКИ

Головною метою кваліфікаційної роботи була розробка сайту для вивчення англійської мови “Englex”.

Проаналізувавши предметну область теми кваліфікаційної роботи, було продумано структуру системи, діаграму варіантів використання для її реалізації.

Проведено поетапний розподіл завдань для виконання та оформлення кваліфікаційної роботи, такі як: збір та опрацювання матеріалу та формування технічного завдання. Вивчено порядок виконання проекту та вимоги до його оформлення. Було висвітлено і обґрунтовано необхідність розробки сайту для вивчення англійської мови «Englex».

У процесі виконання проекту було проведено роботу з базою даних MySQL та використано її для обробки та відображення інформації.

Розроблено програмне забезпечення графічного модуля сайту для вивчення іноземної мови. Дану веб-систему можна застосовувати для отримання інформації з бази даних в зручному графічному вигляді.

Розроблено програмну частину сайту за допомогою мов програмування PHP та JavaScript.

Здійснено тестування сайту за всіма необхідними параметрами.

Дослідивши вищеповисаний об'єм робіт можна стверджувати, що поставлені задачі були успішно виконані.

ПЕРЕЛІК ДЖЕРЕЛ

1. Закон України «Про охорону праці» від 21.11.2002р.
2. ДНАОП 0.00-1.31-99 «Правила охорони праці під час експлуатації електронно-обчислюваних машин», від 10.02.1999 року N 21.
3. ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислюваних машин», від 10.12.1998 р. № 7.
4. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень», від 01 грудня 1999 р. № 42.
5. «Наказ про затвердження вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», від 14.02.2018р. № 207.

6. Основи охорони праці: Підручник. 21ге видання, доповнене та перероблене. / К. Н. Ткачук, М. О. Халімовський, В. В. Зацарний, Д. В. Зеркалов, Р. В. Сабарно, О. І. Полукаров, В. С. Коз'яков, Л. О. Мітюк. За ред. К. Н. Ткачука і М. О. Халімовського. — К.: Основа, 2006 — 448 с.
7. Створення веб-сайтів з нуля [Електронний ресурс] : [Веб-сайт]. — Електронні дані. — Режим доступу: <https://www.hostinger.com.ua/rukovodstva/kak-sozdat-sait/>.
8. Artificial Intelligence [Електронний ресурс] : [Веб-сайт]. — Електронні дані. — Режим доступу: https://www.sas.com/en_nz/insights/analytics/what-is-artificial-intelligence.html.
9. Introduction To K-Means Clustering [Електронний ресурс] : [Веб-сайт]. — Електронні дані. — Режим доступу: <https://www.datascience.com/blog/k-means-clustering>. — Назва з екрану.
10. MAMP & MAMP PRO - local web development solution [Електронний ресурс] : [Веб-сайт]. — Електронні дані. — Режим доступу: <https://www.mamp.info>.
11. Apache Commons Math 3.6 API [Електронний ресурс] : [Веб-сайт]. — Електронні дані. — Режим доступу: <https://commons.apache.org/proper/commons-math/javadocs/api-3.6/overview-summary.html>.
12. Етапи створення (розробки) web-сайту. Процес створення сайтів [Електронний ресурс] : [Веб-сайт]. — Електронні дані. — Режим доступу: <https://itstatti.in.ua/stvorennya-sajtiv/115-etapi-stvorennya-rozrobki-web-sajtu-protses-stvorennya-sajtiv.html>
13. PHP+MySQL [Електронний ресурс] : [Веб-сайт]. — Електронні дані. — Режим доступу: <https://astwellsoft.com/uk/blog/tehnology/php-mysql.html>
14. Мова програмування JavaScript [Електронний ресурс] : [Веб-сайт]. — Електронні дані. — Режим доступу: <https://what.com.ua/mova-programyvannia-javascript/>

ДОДАТКИ

Лістинг файлу «Comments.html»

```
<!DOCTYPE html>
<html lang = "en">
<head>
  <meta charset="utf-8">
  <script src="divs.js"></script>
  <!-- <link href="css/Pagination.css" rel="stylesheet" -->
  <link href="css/Menu.css" rel="stylesheet">
</head>
<body><style>
  body {
    background: url(img/11.png);
    background-size: 100%;}
  #disqus_thread {
    width: 50%;
    margin-left: 25%;
    margin-top: 15%; } </style>
<div id="disqus_thread"></div>
<script>
var disqus_config = function () {
this.page.url = "http://localhost:8888/Diplom/Comments.html";
this.page.identifier = "Comments.html";};
(function() {
var d = document, s = d.createElement('script');
s.src = 'https://proeng.disqus.com/embed.js';
s.setAttribute('data-timestamp', +new Date());
(d.head || d.body).appendChild(s);})(); </script>
<noscript>Please enable JavaScript to view the <a
href="https://disqus.com/?ref_noscript">comments powered by
Disqus.</a></noscript>
<script id="dsq-count-scr" src="//proeng.disqus.com/count.js"
async></script></body></html>
```

Лістинг файлу «Auth.php»

```
<?php

$login = filter_var(trim($_POST['login']), FILTER_SANITIZE_STRING);
$pass = filter_var(trim($_POST['pass']), FILTER_SANITIZE_STRING);

$pass = md5($pass."ghhr56btn");

$mysql = new mysqli('localhost', 'root', 'root', 'Register');
if ($mysql->connect_errno) {
    echo "Не вдалося підключитись до MySQL: (" . $mysql->connect_errno . ") " . $mysql->connect_error;
}

$result = $mysql->query("SELECT * FROM `users` WHERE `login` = '$login' AND `pass` = '$pass'");
$user = $result->fetch_assoc();
if(count($user) == 0) {
    echo "Користувач не знайдений";
    exit();
}

setcookie('user', $user['name'], time() + 7200, "/");

$mysql->close();
header('Location: /');

?>
```

Лістинг файлу «Form.php»

```

<!DOCTYPE html>

<html lang="ru">
<head>
<title>Тест</title>
<meta charset="utf-8">
<link href="css/Menu.css" rel="stylesheet">
<link href="css/Form.css" rel="stylesheet">

<body>

<div>
  <?php
    if ($_COOKIE['user'] == ''):
  ?>
  <div>

<div class="form">
  <button class="change" class="active1"
id="reg">Реєстрація</button>
  <button class="change" class="active2"
id="aut">Авторизація</button>
  <div class="col" id="f">

    <h1>Реєстрація</h1>
    <form action="php/Check.php" method="post">
      <input type="text" class="form-control" name="login"
id="login" placeholder="Enter your login"><br>
      <input type="text" class="form-control" name="name"
id="name" placeholder="Enter your name"><br>
      <input type="password" class="form-control" name="pass"
id="pass" placeholder="Enter your password"><br>
      <button class="bt" class="btn btn-success"
type="submit">Зареєструватись</button>
    </form>
  </div>
  <div class="col" id="s">
    <h1>Авторизація</h1>
    <form action="php/Auth.php" method="post">
      <input type="text" class="form-control" name="login"
id="login" placeholder="Enter your login"><br>
      <input type="password" class="form-control" name="pass"
id="pass" placeholder="Enter your password"><br>
      <button class="bt" class="btn btn-success"
type="submit">Авторизуватись</button>
    </form>
  </div>
  <a id="gt" href="Test.php">LOG IN AS A GUEST</a>
  <?php endif; ?>
</div>
</div>

```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script>
    $(document).ready(function () {
        $("#s").hide();
        $("#aut").click(function () {
            $("#f").hide();
            $("#s").show();
        });
    });

    $(document).ready(function () {
        $("#reg").click(function () {
            $("#s").hide();
            $("#f").show();
        });
    });
    document.getElementById('reg').style.background='#0b7e8b';
    document.getElementById('reg').style.color='white';

    $('#reg').click(function() {
        document.getElementById('reg').style.background='#0b7e8b';
        document.getElementById('reg').style.color='white';
        document.getElementById('aut').style.background='white';
        document.getElementById('aut').style.color='#0b7e8b';
    })

    $('#aut').click(function() {
        document.getElementById('aut').style.background='#0b7e8b';
        document.getElementById('aut').style.color='white';
        document.getElementById('reg').style.background='white';
        document.getElementById('reg').style.color='#0b7e8b';
    })
</script>
</div>
</body></html>
```


Лістинг файлу «Check.php»

```
<?php

$login = filter_var(trim($_POST['login']), FILTER_SANITIZE_STRING);
$name = filter_var(trim($_POST['name']), FILTER_SANITIZE_STRING);
$pass = filter_var(trim($_POST['pass']), FILTER_SANITIZE_STRING);

if (mb_strlen($login) < 5 || mb_strlen($login) > 30) {
echo "Допустимий розмір логіна 6-30 символів!";
exit();

}
if (mb_strlen($name) < 3 || mb_strlen($name) > 30) {
echo "Допустимий розмір імені 3-30 символів!";
exit();
}
if (mb_strlen($pass) < 5 || mb_strlen($pass) > 30) {
echo "Допустимий розмір паролю 6-30 символів!";
exit();
}
$pass = md5($pass."ghhr56btn");
$mysql = new mysqli('localhost', 'root', 'root', 'Register');
if ($mysql->connect_errno) {
    echo "Не удалось подключиться к MySQL: (" . $mysql->connect_errno . ") " . $mysql->connect_error;
}
$mysql->query("INSERT INTO `users` (`login`, `pass`, `name`)
VALUES('$login', '$pass', '$name')");
$mysql->close();
header('Location: /');
?>
```

Лістинг файлу «Exit.html»

```
<?php  
setcookie('user', $user['name'], time() - 3600, "/");  
header('Location: /');  
?>
```

Лістинг файлу «Main.php»

```

<!DOCTYPE html>
<html lang = "en">
<head>
  <meta charset="utf-8">
  <link href="style.css" rel="stylesheet">

</head>
<body>
  <div id="top">
    <nav>
      <ul class="topmenu">

          <div id="logo"><li id="img"><a href=""></a></li></div>

        <?php
          if ($_COOKIE['user'] == ''):
            ?>
            <div class="coll" id="sss">
              <form action="php/Auth.php" method="post">
                <input type="text" class="form-controll" name="login"
id="login" placeholder="Enter your login"><br>
                <input type="password" class="form-controll" name="pass"
id="pass" placeholder="Enter your password"><br>
                <button class="btn" class="btn btn-success"
type="submit">Авторизуватись</button></br>
                <a href="Form.php" id="hhh">Зареєструватись</a>
              </form>
            </div>
            <?php else:
              ?>
              <li id="ff"><?php
                echo "
                <p id='p'><img
src='http://localhost:8888/Diplom/img/logo.png' width='50'
height='50'>" . "<br>" . $_COOKIE['user'] . "<br><a id='ex'
href='http://localhost:8888/Diplom/Exit.php'>LogOut</a></p>"
              ?></li>
            <?php endif; ?>

          </ul>
        </nav>

        <nav id="bot">
          <ul class="topmenu">
            <li><a href="Main.php">Home</a></li>
            <li><a href="Articles.php">Gramatic</a></li>
            <li><a href="Learn.php" class="down">Words</a>
          </li>

```

```

        <li><a href="Test.php" class="down">English Test</a>
        </li>
        <li><a href="Comments.html">English Chat</a></li>

    </ul>
</nav>
<div id="fl">
<h1 class="nn">Вивчення англійської онлайн</h1>
<h2 class="nn">Вивчайте англійську де завгодно</h2>
    <a href="Form.php"><button id="a1">Створити безкоштовний
акаунт</button></a>
</div>

<div id="test">
    <h1 id="m1">Визначити рівень </br> англійської онлайн</h1>
    <a href="Test.php"><button id="m2">Перейти до тесту</button></a>
</div>

<div id="wds">
    <h1 id="b1">Вчити слова</h1>
    <a href="Levels.php"><button id="b2">Перейти до
вивчення</button></a>
</div>

<div id="grm">
    <h1 id="d1">Вивчати граматику</h1>
    <a href="Articles.php"><button id="d2">Перейти до
вивчення</button></a>
</div>
    <footer id="fot">

        <div id="ddl">
            <a class="jj" id="e0" href="">©2020 ProEng Lingo</a></br>
            <a href="Profile.php"></a></br>
            <a class="jj" id="e1" href="">Home</a>
            <a class="jj" id="e2" href="">Grammatic</a>
            <a class="jj" id="e3" href="">Words</a>
            <a class="jj" id="e4" href="">English Level Test</a>
            <a class="jj" href="">English Chat</a>

        </div>

    </footer>
</body>
<html>

```

Лістинг файлу «Menu.php»

```

<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <!-- <link href="css/Menu.css" rel="stylesheet"> -->
  </head>
  <body>
    <div id="top">
      <nav>
        <ul class="topmenu">
          <div id="logo"><li id="img"><a href=""></a></li></div>
          <!-- <div id="sl"><strong>LEARN ENGLISH WITH
US</strong></div> -->
          <!-- <div id="sl"><li id="img"><a href=""></a></li></div> -->
          <?php
            if ($_COOKIE['user'] == ''):
              ?>
              <div id="form">
                <li><input class="r" type="button" value="Log In"
onClick='#'"></li>
                <li><input class="r" type="button" value="Sign Up"
onClick='#'"></li>
              </div>
              <?php else: ?>
              <li id="ff"><?php
                echo "<img src='http://localhost:8888/Diplom/img/logo.png'
width='50' height='50'" . "</br>" . $_COOKIE['user'] . "</br><img
id='arr' src='http://localhost:8888/Diplom/img/arr.png' width='50'
height='50'"
                ?></li>
              <li id="exit"><a
href="Applications/MAMP/htdocs/Diplom/Exit.php">Exit</a></li>
              <?php endif; ?>
            </ul>
          </nav>
          <style>
          .r{
            color:#00c6ff;
            padding: 10px 20px;
            width:100px;
            text-decoration:none;
            text-align:center;
            display: block;
            background: white;
            color: #387c89;
            margin-bottom: 10px;

```

```
background-position: 100% 0;
background-size: 200% 100%;
transition: all .25s ease-in;
font: 400 18px tahoma;
border: 1px solid #387c89;
}
.r:hover {
background: #387c89;
color: white;
}
#form {
float: right;
margin-right: 30px;
margin-bottom: 60px;
padding-top: 25px;
}
#logo, #ff {
display: inline-block; }
#exit {
display: none;
}
#ff {
float: right;
margin-top: 10px;
margin-right: 50px; }
#gg, #ff {
display: inline-block;
}
body {
background: url(img/mm.jpg); }
.topmenu {
backface-visibility: hidden;
background: rgba(255,255,255,.6);
}
.topmenu > li {
display: inline-block;
position: relative; }
.topmenu > li > a {
font-family: 'Exo 2', sans-serif;
height: 70px;
line-height: 100px;
padding: 0 30px;
font-weight: bold;
color: #003559;
text-transform: uppercase;
}
.down:after {
/* content: "\f107"; */
margin-left: 80px;
font-family: FontAwesome; }
.topmenu li a:hover { color: #E6855F; }
```

```

.submenu {
  background: white;
  border: 2px solid #003559;
  position: absolute;
  left: 0;
  visibility: hidden;
  opacity: 0;
  z-index: 5;
  width: 150px;
  transform: perspective(600px) rotateX(-90deg);
  transform-origin: 0% 0%;
  transition: .6s ease-in-out; }
.topmenu > li:hover .submenu{
  visibility: visible;
  opacity: 1;
  transform: perspective(600px) rotateX(0deg); }
.submenu li a {
  color: #7f7f7f;
  font-size: 13px;
  line-height: 20px;
  padding: 0 25px;
  font-family: 'Kurale', serif; }
nav ul {
  list-style: none;
  padding: 0; }
nav a {
  display: block;
  text-decoration: none;
  outline: none;
  transition: .4s ease-in-out; }
#sl, #logo, #form {
  display: inline-block;}
#logo {
  padding-top: 10px;}
#top {
  background-size: 100%;}
#bot {
  margin-bottom: 150px;
  margin-top: -16px;}
</style>
<nav id="bot">
  <ul class="topmenu">
    <li><a href="">Home</a></li>
    <li><a href="">Gramatic</a></li>
    <li><a href="" class="down">Words</a>
      <ul class="submenu">
        <li><a href="">Category</a></li>
        <li><a href="">Author</a></li>
        <li><a href="">Archive</a></li>
        <li><a href="">Tags</a></li>
      </ul>
    </li>
  </ul>

```

```
</li>
<li><a href="" class="down">Tests</a>
  <ul class="submenu">
    <li><a href="">Category</a></li>
    <li><a href="">Author</a></li>
    <li><a href="">Archive</a></li>
    <li><a href="">Tags</a></li>
  </ul>
</li>
<li><a href="Comments.html">English Chat</a></li>
</ul>
</nav>
</body>
</html>
```


Лістинг файлу «Update.php»

```
<?php
$name = $_POST['name'];
$old = $_POST['old'];
$new = $_POST['new'];
$age = $_POST['age'];
$nm = $_COOKIE['user'];
$mw = $_POST['sex'];
$city = $_POST['city'];

$mysql = new mysqli('localhost', 'root', 'root', 'Register');
if ($mysql->connect_errno) {
    echo "Не вдалось підключитись до MySQL: (" . $mysqli-
>connect_errno . ") " . $mysqli->connect_error;
}

if($age < 10 || $age > 99) {
    echo "Недопустимий вік";
} else {
    $mysql->query("UPDATE `users` SET `age`=' $age' WHERE
`name`=' $nm'");
}

$mysql->query("UPDATE `users` SET `mw`=' $mw' WHERE `name`=' $nm'");

if ($city != '')
$mysql->query("UPDATE `users` SET `city`=' $city' WHERE
`name`=' $nm'");
    $sql = "SELECT pass FROM users WHERE name = ' $nm'";
    $result = mysqli_query($mysql,$sql);
    $row = mysqli_fetch_array($result);

$new1 = md5($new."ghhr56btn");

if (strlen($new) > 5) {
$result = "UPDATE `users` SET `pass`=' $new1' WHERE `name`=' $nm'";

if ($mysql->query($result) == TRUE) {
    echo "Інформація оновлена успішно!</br>";
    echo "<a href='Profile.php'>Повернутися до профілю</a>";
} else {
    echo "Помилка " . $mysql->error;
}
}
exit();
}

$mysql->close();
header('Location: http://localhost:8888/Diplom/Profile.php');

?>
```

Лістинг файлу «Learn.php»

```

<!DOCTYPE html>
<html lang = "en">
<head>
  <meta charset="utf-8">
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <div id="top">
    <nav>
      <ul class="topmenu">
        <div id="logo"><li id="img"><a href="Main.php"></a></li></div>

        <?php
          if ($_COOKIE['user'] == ''):
            ?>
            <div class="coll" id="sss">
              <form action="php/Auth.php" method="post">
                <input type="text" class="form-controll" name="login"
id="login" placeholder="Enter your login"><br>
                <input type="password" class="form-controll" name="pass"
id="pass" placeholder="Enter your password"><br>
                <button class="btn" class="btn btn-success"
type="submit">Авторизуватись</button></br>
                <a href="Form.php" id="hhh">Зареєструватись</a>
              </form>
            </div>
            <?php else:
              ?>
              <li id="ff"><?php
                echo " <p id='p'><a href='Profile.php'><img
src='http://localhost:8888/Diplom/img/logo.png' width='50'
height='50'></a>" . $_COOKIE['user'] . "<br><a id='ex'
href='http://localhost:8888/Diplom/Exit.php'>LogOut</a></p>"
              ?></li>
            <?php endif; ?>

          </ul>
        </nav>
      </div>

      <nav id="bot">
        <ul class="topmenu">
          <li><a
href="/Applications/MAMP/htdocs/Diplom/Main.php">Home</a></li>
          <li><a href="">Gramatic</a></li>
          <li><a href="Levels.php" class="down">Words</a>
</li>
          <li><a href="" class="down">Tests</a>
</li>

```

```

    <li><a href="Comments.html">English Chat</a></li>

  </ul>
</nav>
<div class="lit">
  

</div>

<div id="block">
<div class="bl"><a href="a1_10.php"></a></div>
<div class="bl"><a href="a1_9.php"></a></div>
<div class="bl"><a href="a1_8.php"></a></div>
<div class="bl"></div></br>

<div class="bl"></div>
<div class="bl"></div>
<div class="bl"></div>
<div class="bl"></div></br>

<div class="bl"></div>
<div class="bl"></div>
<div class="bl"></div>
<div class="bl"></div></br>

<div class="bl"></div>
<div class="bl"></div>
<div class="bl"></div>
<div class="bl"></div></br>
</div>

<footer id="fot">

  <div id="dd1">
    <a class="jj" id="e0" href="">©2020 ProEng Lingo</a></br>
    </br>
    <a class="jj" id="e1" href="">Home</a>
    <a class="jj" id="e2" href="">Grammatic</a>
    <a class="jj" id="e3" href="">Words</a>
    <a class="jj" id="e4" href="">English Level Test</a>
    <a class="jj" href="">English Chat</a>

  </div>

</footer>

```

Лістинг файлу «Panel.php»

```
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Тест</title>
<meta charset="utf-8">

<link href="css/Menu.css" rel="stylesheet">
<link href="css/Form.css" rel="stylesheet">

<body>

<div id="top">
  <h1 id="cn"> АДМІНІСТРАТИВНА ПАНЕЛЬ</h1>

</div>
<div id="all">
  <h1>Додати слова</h1>

<form action="Wds.php" method="post">

  <input type="text" class="form-control" name="uk" id="uk"
placeholder="Введіть слово українською"></br>
  <input type="text" class="form-control" name="en" id="en"
placeholder="Введіть переклад"></br>

<select id="sel" name="sl">
  <option value="lvl">Оберіть складність</option>
  <option value="L">A1-A2</option>
  <option value="M">B1-B2</option>
  <option value="H">C1-C2</option>
</select>

<select id="sel" name="sel">
  <option value="lvl" disabled selected>Оберіть категорію</option>
  <option value="Animals">Тварини</option>
  <option value="Body">Тіло</option>
  <option value="Clothes">Одяг</option>
  <option value="Colors">Кольори</option>
  <option value="Food">Їжа та напої</option>
  <option value="House">Дім</option>
  <option value="Nature">Природа</option>
  <option value="School">Освіта</option>
  <option value="Transport">Транспорт</option>
  <option value="Countries">Країни</option>
  <option value="Internet">Інтернет</option>
  <option value="People">Люди</option>
  <option value="Travel">Подорожі</option>
  <option value="Weather">Погода</option>
  <option value="Entertainment">Розваги</option>
</select></br>
```

```
<button class="bt" id="b1" class="btn btn-success"
type="submit">Підтвердити</button>
</br><a id="ss"href="Main.php">Повернутися на сайт</a>

</form>
</div>
```

Лістинг файлу «style.css»

```
.art {
  margin: 26px;}
#hd {
  text-align: center;
  font-size: 30px;
  margin-right: 20px;
}
#ex {
  color: black;
  font-size: 15px;
}
#content {
  height: 650px;}
#sl, #logo, #form {
display: inline-block;
}
#logo {
padding-top: 10px;
}
#top {
background-size: 100%;
}
#bot {
margin-bottom: 50px;
margin-top: -16px;}
#rec {
  background: white;
  font-size: 20px;
  width: 27%;
  float: left;
  position: relative;
}
#gram {
  width: 67%;
  float: right;
  margin-left: 30px;
  background: white;
  margin-right: 20px;
  margin: 0 1.5%;
position: relative;
}
p, h1, h2, h3 {
  margin-left: 20px;
}
#d1, #d2, #d3, #d4 {
  display: inline-block;
}
#href:hover {
background: white;
color: #0b7e8b;
border: 2px solid #0b7e8b;
```

```
}
#fr {
text-decoration: none;
width: 98%;
color: black;
}
#href {
background: #0b7e8b;
color: white;
border: 2px solid white;
padding: 15px;
text-align: center;
width: 98%;
font-size: 15px;
}
#fr:hover {
color: white;
}
}
#ex {
padding-top: 5px;
color: blue;
font-family: serif;
text-decoration: none;
}
.r{
color:#00c6ff;
padding: 5px 10px;
width:100px;
text-decoration:none;
text-align:center;
display: block;
background: rgba(255,255,255,0.1);
color: #387c89;
margin-bottom: 10px;
background-position: 100% 0;
background-size: 200% 100%;
transition: all .25s ease-in;
font: 400 18px tahoma;
border: 1px solid #387c89;
}
.r:hover {
background: #387c89;
color: white;
}
#form {
float: right;
margin-right: 30px;
margin-bottom: 60px;
padding-top: 45px;
}
#logo, #ff {
```

```

display: inline-block;
}
#exit {
display: none;
}
#p {
color: #387c89;
}
#ff {
float: right;
margin-top: 20px;
margin-right: 50px;
}
#gg, #ff {
display: inline-block;
}
body {
background-size: 100%;
padding-left: 50px;
background: #eef3f2;
}
#new {
margin: 20px;
}
td {
padding: 20px;
}
.topmenu {
backface-visibility: hidden;
background: white;
margin-right: 20px;
}
.topmenu > li {
display: inline-block;
position: relative;
}
.topmenu > li > a {
font-family: 'Exo 2', sans-serif;
height: 30px;
line-height: 20px;
padding: 0 30px;
font-weight: bold;
color: #003559;
text-transform: uppercase;
}
.down:after {
margin-left: 80px;
font-family: FontAwesome;
}
.topmenu li a:hover { color:#387c89; }
nav ul {
list-style: none;

```



```
padding: 0;
}
nav a {
display: block;
text-decoration: none;
outline: none;
transition: .4s ease-in-out;
}
```

Лістинг файлу «Articles.php»

```

<!DOCTYPE html>

<html lang="ru">
<head>
<title>Тест</title>
<meta charset="utf-8">
<!-- <link rel="import" href="Menu.html"> -->
<!-- <link href="css/Menu.css" rel="stylesheet"> -->
<link href="style.css" rel="stylesheet">
<script type="text/javascript" src="/js/blockajax.js"></script>
</head>
<body>
<div id="top">
  <nav>
    <ul class="topmenu">
      <div id="logo"><li id="img"><a href="Main.php"></a></li></div>

      <?php
        if ($_COOKIE['user'] == ''):
          ?>
          <div id="form">
            <li><input class="r" type="button" value="Log In"
onClick="#"></li>
            <li><input class="r" type="button" value="Sign Up"
onClick="#"></li>
          </div>
          <?php else:
            ?>
            <li id="ff"><?php
              echo "<p id='p'><a href='Profile.php'><img
src='http://localhost:8888/Diplom/img/logo.png' width='50'
height='50'></a>" . $_COOKIE['user'] . "</br><a id='ex'
href='http://localhost:8888/Diplom/Exit.php'>LogOut</a></p>"
              ?></li>
            <?php endif; ?>

          </ul>
        </nav>
      </div>

<nav id="bot">
  <ul class="topmenu">
    <li><a
href="/Applications/MAMP/htdocs/Diplom/Main.php">Home</a></li>
    <li><a href="Articles.php">Gramatic</a></li>
    <li><a href="Levels.php" class="down">Words</a>
</li>
    <li><a href="Test.php" class="down">English Level Test</a>
</li>
    <li><a href="Comments.html">English Chat</a></li>
  </ul>

```

```
</nav>
```

```
<div id="content">
```

```
<div id="rec">
</br><p id="hd"><strong>Рекомендації</strong></p>
<p class="md" id="h1">Present Simple</p>
<p class="md" id="h2">Past Simple</p>
<p class="md" id="h3">To Be</p>
<p class="md" id="h4">Pronouns</p></br>
```

```
</div>
```

```
<div id="gram">
```

```
<div id="d1" class="art"></div>
<div id="d2" class="art"></div>
<div id="d3" class="art"></div></br>
<div id="d4" class="art"></div>
```

```
</div>
```

```
</div>
```

```
<footer id="fot">
```

```

<div class="col" id="ss">
  <h1>Авторизація</h1>
  <form action="php/Auth.php" method="post">
    <input type="text" class="form-control" name="login"
id="login" placeholder="Enter your login"><br>
    <input type="password" class="form-control" name="pass"
id="pass" placeholder="Enter your password"><br>
    <button class="bt" class="btn btn-success"
type="submit">Авторизуватись</button></br>
    <a href="" id="hh">Зареєструватись</a>
  </form>
</div>
```

```
<div id="kk">
```

```

<div id="dd1">
  <ul class="ul"><a class="jj" href="">GRAMMATIC</a>
  <li><a class="rr" href="">Present Simple</a></li>
  <li><a class="rr" href="">Past Simple</a></li>
  <li><a class="rr" href="">Pronouns</a></li>
```

```
    <li><a class="rr" href="">To Be</a></li>
  </ul>
</div>

<div id="dd3">
  <ul><a class="jj" href="">TESTS</a>
  <li><a class="rr" href="">English Level</a></li>
  <li><a class="rr" href="">Present Simple</a></li>
  <li><a class="rr" href="">Past Simple</a></li>
  <li><a class="rr" href="">Pronouns</a></li>
  <li><a class="rr" href="">To Be</a></li>
  </ul></div>
<div id="dd2">
<ul class="ul"><a class="jj" href="">WORDS</a></ul>
</div>

<div id="dd4">
  <ul><a class="jj" href="">ENGLISH CHAT</a></ul>
</div>
</div>
<div id="gg">
  
</div>

</footer>
</body>
</html>
```

Діаграма варіантів використання

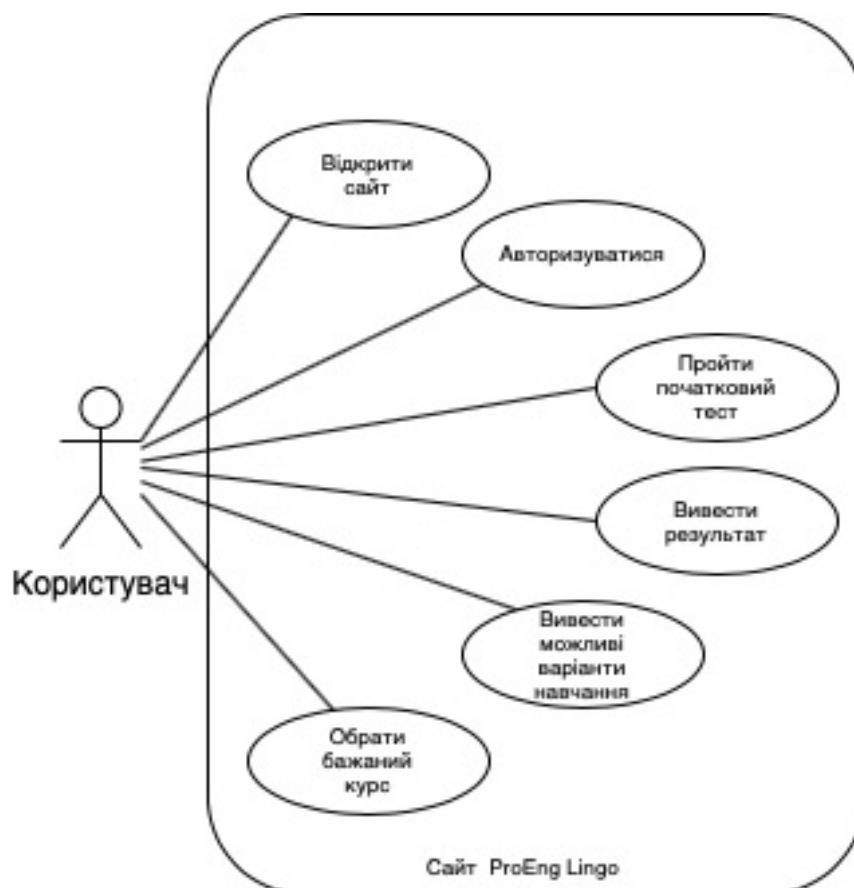


Схема таблиць баз даних

Words A1	
id	int(11)
Eng	varchar(30)
Ukr	varchar(30)
Theme	varchar(30)
Level	varchar(2)

Register Progress	
id	int(11)
name	varchar(30)
res	int(100)

Register users	
id	int(11) unsigned
login	varchar(30)
pass	varchar(32)
name	varchar(30)
age	int(3)
mw	varchar(10)
city	varchar(30)
level	varchar(3)

Register Admin	
id	int(1)
login	varchar(20)
password	varchar(20)

Діаграма послідовності дій



Структурна схема сайту

