1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАНЬ

* 1. Дослідження предметної області

Англійська мова дуже важлива, все більше батьків починають віддавати своїх дітей на курси іноземних мов ще до того, як вони підуть до школи. Дійсно, чим раніше діти починають вчитися, тим краще їм це вдається. У підлітковому віці це дозволяє розуміти світові тренди та спілкуватися з однолітками з інших країн, або вступити в омріяний вищий навчальний заклад.

Створення вебсервісу для вивчення основ англійської мови, спрямований саме на учнів початкової школи є важливою ініціативою. Він може бути використаний для:

* покращення якості освіти шляхом заохочення до вивчення іноземних мов в початкових класах;
* підвищення середнього рівня знань школяра в Україні;
* розвиток навичок читання та письма;
* використання ігрових елементів для стимулювання інтересу учнів та покращити процес навчання;
* підготовка до здачі зовнішнього незалежного оцінювання або тестів отримання сертифікатів яких, підтверджує володіння мовою на певному рівні;
* вебсервіс може служити додатковим ресурсом для вчителів, які можуть використовувати його для підтримки навчального процесу в класі.

Саме тому, створення вебсервісу з вивчення англійської мови є важливим етапом для того, щоб показати батькам та їх дітям, які нові можливості відкривають знання іноземних мов.

* 1. Обґрунтування вибору теми

У сучасному глобалізованому світі англійська мова стала необхідним інструментом для спілкування, навчання, розвитку кар’єри та культурного розуміння. Англійська мова вибрана зважаючи на свою всесвітню популярність та важливість. На рисунку 1.1, можна побачити статистику в якій 51% українців зазначили, що мають деякі знання з англійської мови. Однак, як зазначають соціологи, детальніший аналіз показує, що тільки 23% з них можуть читати, писати та спілкуватися цією мовою на побутовому і навіть на професійному рівнях.



Рисунок 1.1 —Опитування, проведене Київським міжнародним інститутом соціології

Дивлячись в майбутнє, створення веб-системи для спрощення та покращення навчального процесу вивчення англійської мови є важливим при подальшому пошуку високооплачуваної роботи. Такими можуть бути:

* керівник компанії, менеджер проєкту;
* журналіст;
* копірайтер;
* менеджер з продажу;
* маркетолог;
* програміст;
* економіст.

Рівень зарплати прямо пропорційно залежить від рівня володіння англійською. Приклад дослідження від Work.ua наведено на рисунку 1.2.



Рисунок 1.2 — Дослідження Work.ua щодо залежності заробітної плати від рівня англійської мови

Підсумовуючи все вище сказане, англійська мова важлива для вивчення з початкової школи. Вона впливає на подальший кар’єрний ріст та можливості які відкриватимуться по всьому світу.

* 1. Аналіз наявних рішень

Під час проектування технічного завдання слід проаналізувати існуючі веб-системи для навчання англійської мови. Дуже важливо виділити наявний функціонал та відсутній, щоб пізніше мати можливість оптимально спроектувати технічне завдання.

Першим рішенням проаналізуймо Duolingo. Це електронна платформа вивчення мови, що має платний та безплатний види акаунтів. Дуже відомий на весь світ своїм стилем та зеленою совою, яка виступає широковідомим символом, через що і популярність велика.

Перевагами є те, що більшість матеріалу платформи в безплатному доступі, проте. Платна версія пропонує більший функціонал з вивчення англійської мови. До прикладу, можливість відпрацьовувати питання або рівні, в яких допустилася помилка при проходженні. Дизайн та анімація сайту спроектований в ігровому стилі для привертання уваги користувача, що є дуже хорошим стимулятором до навчання дітям. Основна фішка в дизайні платформи це персонажі, які протягом проходження рівнів запрограмовані відтворювати різний текст для симуляції реальної розмови, що продемонстровано на рисунку 1.3.

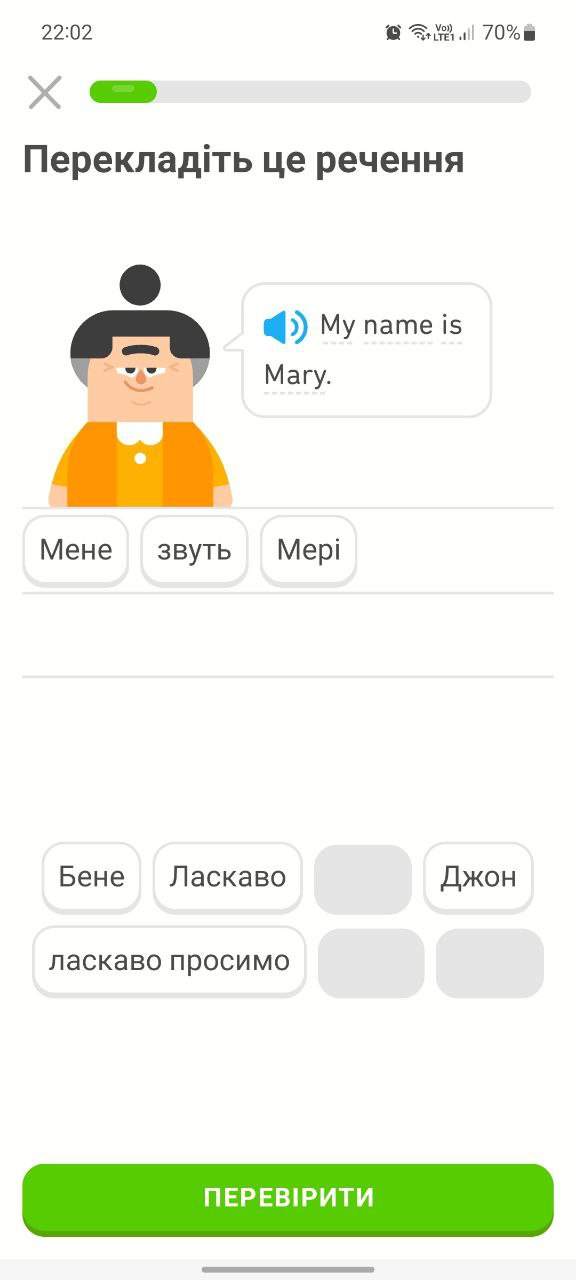


Рисунок 1.3 — Демонстрація проходження рівня в Duolingo

Наявна система ліг, яка дозволяє змагатись з іншими шляхом накопичення очок, які зачисляються після проходження рівнів. Чим більше очків – тим більший шанс перейти в наступну лігу. Варто відзначити широкий вибір мов для вивчення.

Недоліками платформи виступає неможливість на момент аналізу вибір до вивчення інших іноземних мов крім англійської, якщо користувач хоче використовувати інтерфейс та пояснення виключно українською мовою. Це дуже ускладнює процес навчання.

Наступна проаналізована платформа — Kahoot!. Це навчальна платформа в ігровому стилі, яка широко використовується для швидких та цікавих тестувань в навчальних закладах по всьому світу.

Переваг тут чимало, одна з таких — наявність каталогу ігор-вікторин, кожна з яких містить питання з декількома варіантами відповіді. Платформа абсолютно безплатна та не має функціоналу, який розблоковується тільки після придбання платної підписки. Перевагою серед своїх конкурентів, платформа отримала через швидкість проведення тестування. Це відбувається так: користувач, в пошуковій системі пише Kahoot!, заходить на сайт і перше що він бачить це поле, де потрібно ввести код, щоб потрапити на тестування. Це продемонстровано на рисунку 1.4.

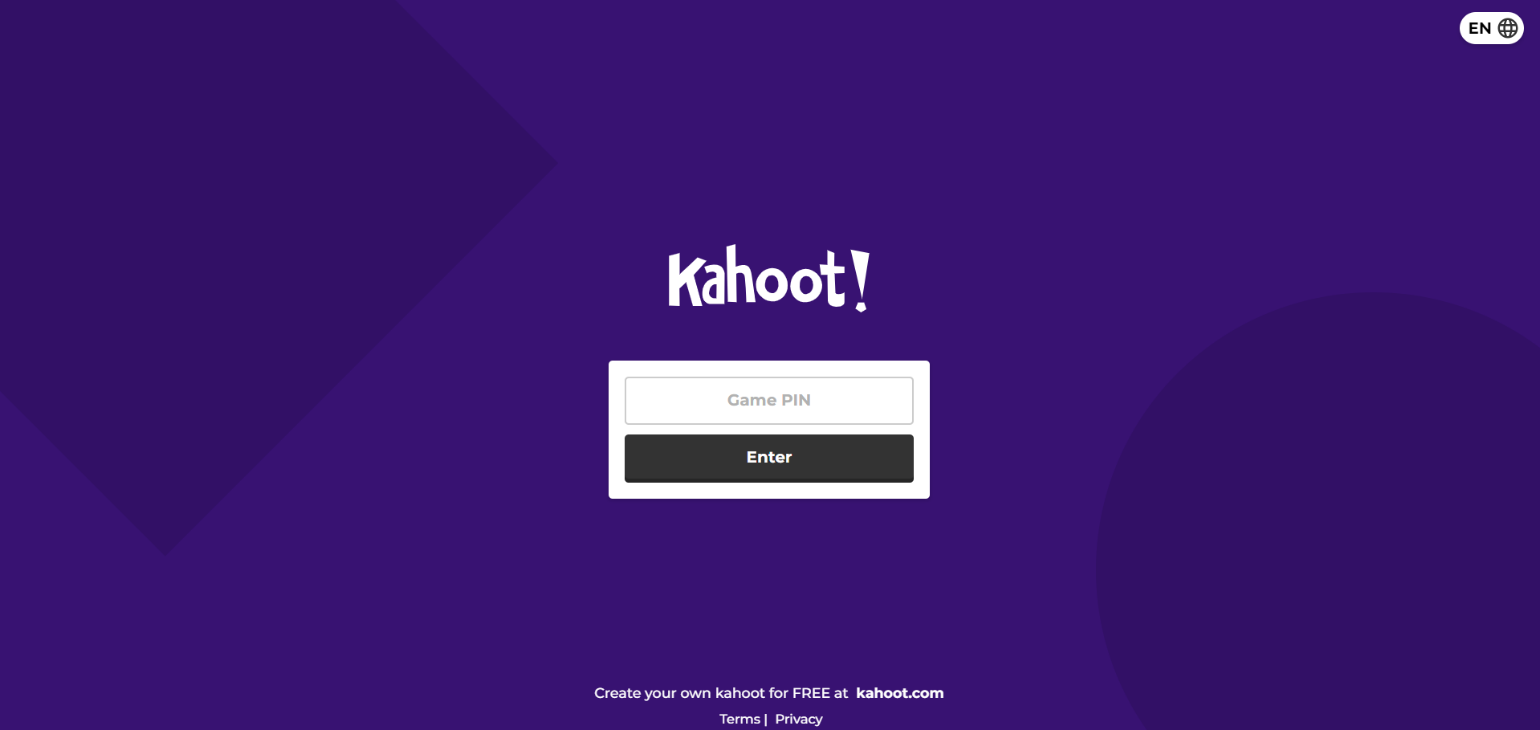


Рисунок 1.4 — Вигляд сторінки входу до тестування на платформі Kahoot!

У випадку, коли опитування проводиться серед дітей або незареєстрованих користувачів — це зручно, не потрібно реєструватись для отримання доступу. Потрібно лише ввести код та вписати ім’я для аутентифікації гравця.

Недоліком є обмежений функціонал під час створення тестувань, можливо створити тест тільки з вибором однієї правильної відповіді (рисунок 1.5).

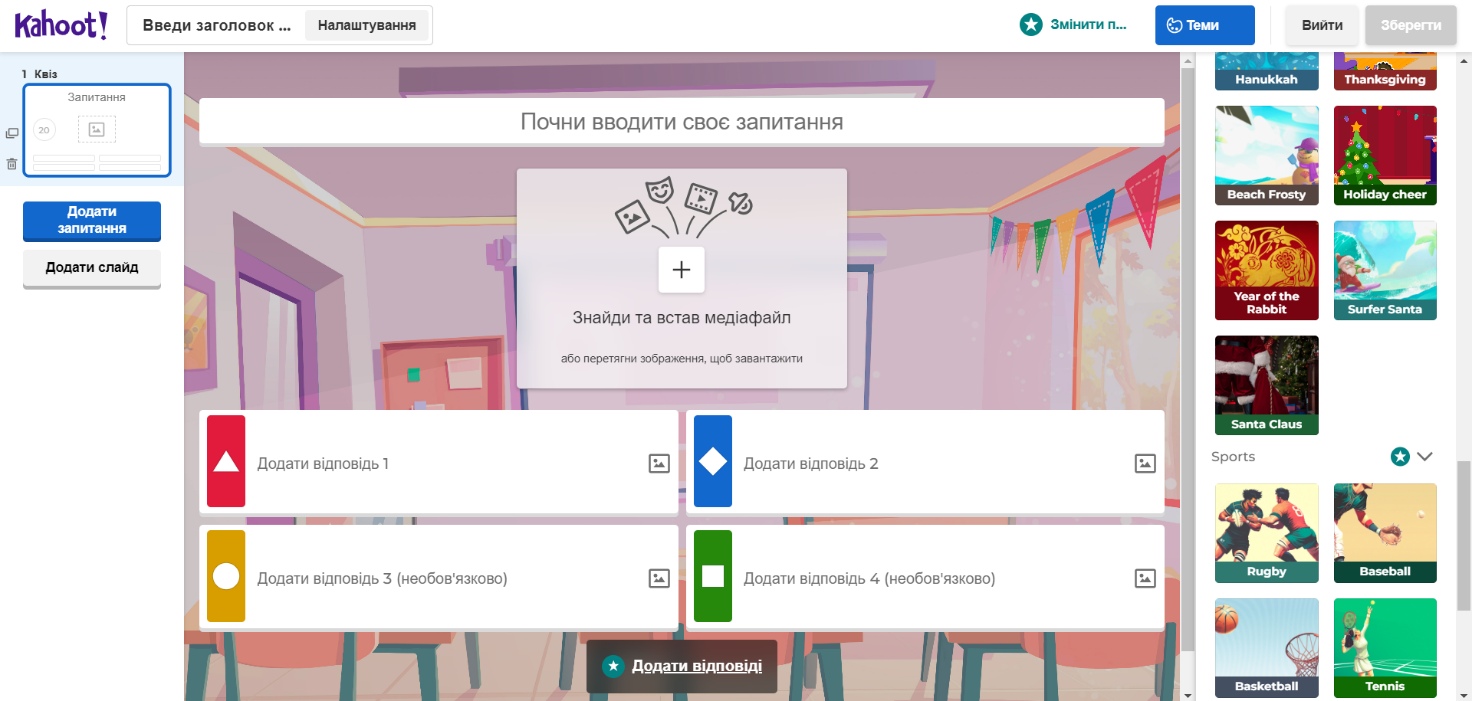


Рисунок 1.5 — Інтерфейс створення тестування на платформі Kahoot!

Платформу не можна назвати націленою на вивчення саме іноземних мов, вона спеціалізується на полегшені проведення опитувань в класі та робить дуже зручною систему під’єднань до тесту на аутентифікації гравців.

Третьою проаналізованою платформою буде Quizlet. Це сервіс, який як і попередній, спеціалізується на покращені та спрощені проведення тестувань. Сервіс дозволяє легко запам’ятати будь-яку інформацію.

Перевагою перед попередніми веб-системами є ширший функціонал при створенні чи проходженні тестувань. Доступне вивчення використовуючи флеш-картки, співвідношення, з вибором однієї правильної відповіді з текстовим чи аудіо умовою, з вибором декількох правильних відповідей та вибір істини чи хибної відповіді. Це все можна отримати оплативши платну підписку на сайті. Після чого в будь-який момент можна віднайти потрібне тестування та пройти його ще раз. Великим плюсом є можливість створення опитування на основі флеш-карток. Це працює наступним чином: користувач створює або знаходить потрібні картки, вибирає функцію “Створити тест”, та на основі кількості інформації формує на вибір ту кількість на форму питань, яку захоче. Це може бути суцільний вибір однієї правильної відповіді чи різноманітне тестування. Це продемонстровано на рисунку 1.6.

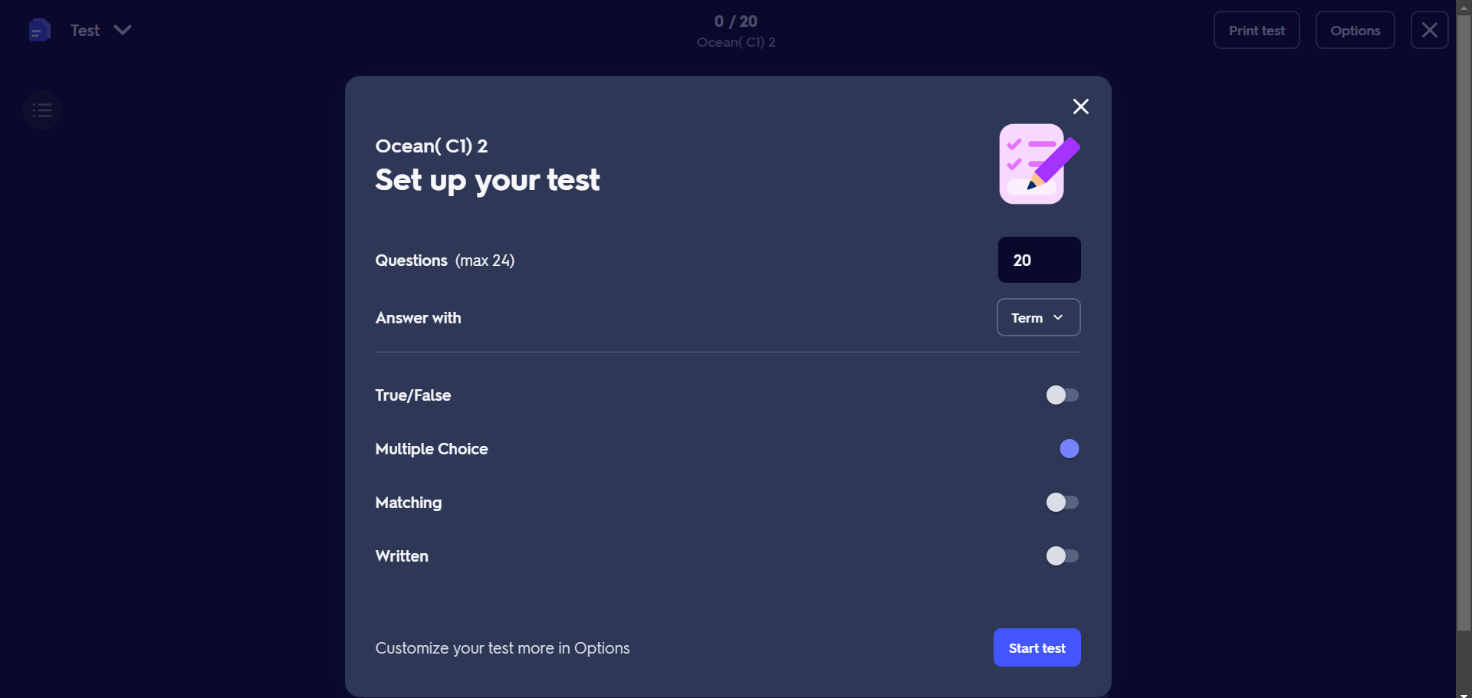


Рисунок 1.6 — Процес створення користувацького тесту на платформі Quizlet

Як альтернатива простим паперовим опитуванням цей сайт є прекрасною варіантом для внесення різноманіття в навчальний процес, що особливо важливо при навчанні з учнями початкових класів.

Недоліками є доволі не дешева ціна за платну підписку на платформі. Також варто звернути увагу на те, що сервіс не спрямований на вивченні саме англійської мови, це може бути будь-що інше. На сайті немає функції разового проходження чи аутентифікації користувача, що ускладнює процес встановлення правдивості тестування.

* 1. Аналіз вимог та постановка завдання

Обміркувавши всі переваги та недоліки платформ які допомагають вивчати англійську мову, можна виділити основні вимоги до веб-системи.

Перш за все, дизайн веб-системи повинен бути в ігровому стилі, що дозволить привернути увагу школярів та зробити процес навчання цікавим. Лиш тільки так можна досягнути більших результатів. Кольори та загальне оформлення не повинно викликати відрази або бажання якнайшвидше залишити платформу.

Для вчителів повинна бути можливість простого та швидкого створення тестувань з різними типами питань. Це може бути як з вибором однієї або декількох правильних відповідей так і встановлення співвідношення. Для учнів ж можливість пройти тестування. Після цього система повинна видати персональний результат. На сторінці особистого профілю реалізувати можливість перегляду всіх пройдених раніше тестувань та їх результати. Таким чином вся необхідна інформація буде в зручному місці та доступ до якої немає часових обмежень.

На окремій сторінці створити список з всіх існуючих тестувань, які були створені раніше. Це зроблено для розширення можливостей навчання в індивідуальних цілях.

Реєстрація та вхід користувача в веб-системі повинен реалізовуватися зручно, щоб батькам, вчителям чи дітям не було важко пройти процес реєстрації. Обов’язково повинне використовуватись шифрування паролів задля зберігання конфіденційної інформації кожного користувача.

Навігація по сайту не повинна складатись з багатьох елементів. Це може погано вплинути на сприйняття школярів. Все повинно бути максимально просто та доступно. На рисунку 1.7 продемонстровано варіанти використання веб-системи.



Рисунок 1.7 — Варіанти використання веб-системи

Кожен користувач має свої особливі можливості. Весь функціонал який доступний учневі:

* проходження тестування;
* отримання результату;
* реєстрація/вхід;
* перегляд особистого профілю;
* перегляд всіх тестувань.

Функціонал вчителя дозволяє створювати тестування, що є основним для даного користувача. Далі перераховано доступний функціонал:

* проходження тестування;
* отримання результату;
* реєстрація/вхід;
* перегляд особистого профілю;
* перегляд всіх тестувань;
* створення тестувань.

Адаптація під мобільні додатки теж важливий крок який не можливо не оминути. Для ефективного використання веб-системи на будь-яких пристроях це обов’язково для реалізації.

Підсумовуючи все вище сказане, веб-система повинна мати функціонал створення, проходження та отримання результатів тестувань, мати доступ до особистого профілю. Для адміністраторів додатково доступ до перегляду або редагуванню бази даних.

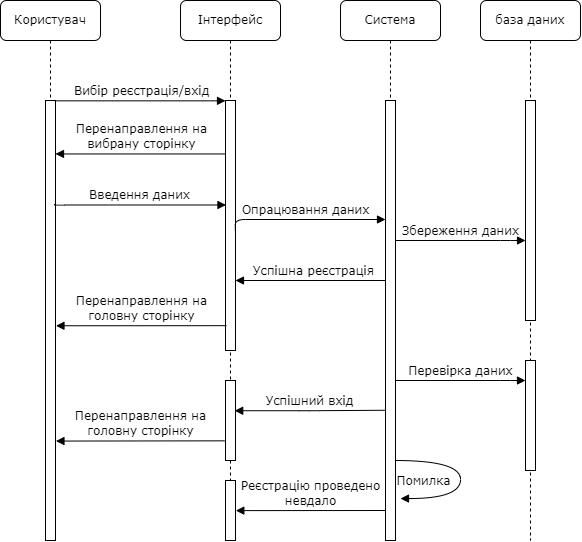
2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Формалізація вимог

Формалізація функціональних вимог базується на технічних потребах кінцевого продукту. Такими є:

* захист інформації. Базовий рівень захисту інформації, який буде реалізований шляхом обмеження функціональних можливостей окремої групи користувачів;
* можливість реєстрації/входу в систему. Доступ до можливостей системи та інформації надаватиметься залежно від типу користувача. Інформації необхідної для реєстрації та входу потрібно використовувати мінімально, зважаючи на те, що система розроблятиметься в більшості для школярів молодших класів, хоча під чітким керуванням вчителя та допомоги зі сторони батьків це не спричинятиме великих проблем. На рисунку 2.1 продемонстровано діаграму послідовності реєстрації/входу користувача;

Рисунок 2.1 — Діаграма послідовності реєстрації/входу в систему



* доступ до перегляду особистого профілю. На даній сторінці розташовуватиметься особиста інформація, з метою надання користувачу можливості ідентифікувати акаунт або можливі помилки введених даних. Також розміщення інформації про пройдені тестування;
* доступ до перегляду навчального матеріалу. Можливість переглядати теоретичну інформацію, створену вчителями, з метою пояснення та вивчення матеріалу. Може проводитись як на уроці, так і самостійно вдома;
* можливість створювати/проходити розділи тестування. Тестування, спрямоване на підтвердження учнем засвоєння інформації. Можливість окремою групою користувачів створювати тестування. Можливість проходження надаватиметься всім групам користувачів;
* доступ до навігаційної панелі системи. Основним способом навігації по системі буде саме навігаційна панель. Метою буде створення мінімальної кількості розділів навігація для полегшення процесу знаходження необхідної інформації;
* перегляд/редагування даних бази даних. Можливість реалізовуватиметься виключно для адміністраторів системи. Надаватиметься обмеженій групі користувачів з метою забезпечення безпеки цілісності даних.

Нефункціональні вимоги однаково важливі як і функціональні. Попередній аналіз показує необхідність чітко сформувати пункти, такі як:

* простий та привабливий дизайн системи. Яскрава колірна гама, простий дизайн необхідний для мотивації учнів брати участь у процесі навчанні використовуючи систему для отримання інформації;
* зручна та інтуїтивно зрозуміла структура системи. Очікуючи те, що учні молодших класів ще не мають достатнього досвіду для самостійного пошуку тої чи іншої інформації оптимізація та спрощення структури системи необхідно реалізувати;
* мобільна адаптація. Один з найважливіших пунктів, зважаючи на те, що під час уроків і вдома учні не матимуть можливості придбати або використовувати персональні комп’ютери чи ноутбуки під час користування системою.

Формалізація вимог важливе на етапі проєктування системи. Наступні кроки залежать від правильно попередньо виконаних етапів. Потрібно врахувати всі можливі проблеми, потреби які виникатимуть під час реалізації або кінцевого користування продуктом.

2.2 Проєктування структури

В даній системі важливо виділити лише найнеобхідніші елементи та розташувати їх в інтуїтивно зрозумілих модулях. Варто зауважити, що всі сторінки крім реєстрації та входу матимуть доступ до компонентів навігації системою, тому далі це не прописано. Отже, структура спроєктована наступним чином:

Сторінка реєстрації. Перше що побачить новий користувач — саме цю сторінку. Система не даватиме доступ до матеріалів без акаунту. Компоненти які будуть використовуватимуться:

* поле для вводу даних;
* кнопка підтвердження реєстрації;
* кнопка переходу на сторінку для входу.

Сторінка входу. Якщо користувач вже має створений акаунт, за допомогою даної сторінки можна увійти в систему, після чого буде відображена сторінка з навчальними матеріалами. На цій сторінці розміщуватимуться:

* поле для вводу даних;
* кнопка підтвердження входу;
* кнопка переходу на сторінку для реєстрації.

Особистий профіль. Відображатиме інформацію про користувача. Також, в іншій частині сторінки розміщуватиметься історія пройдених тестувань, для зручності відстеження. Компоненти будуть наступні:

* блок з особистою інформацією;
* блок з пройденими тестуваннями;
* кнопка виходу з профілю;

Навчальні матеріали. Основна сторінка, на якій буде відображено всі доступні теми для вивчення. Припускається, що керувати процесом навчання буде вчитель, за власним розсудом, тому система не надаватиме можливості створити стрічку з тем чи чогось подібного. Відображення буде в вигляду списку з переліком всіх тем на вибір. Компонентами на даній сторінці буде:

* список тем;
* кнопка переходу сторінку відповідної теми.

Сторінка вибраної теми. Сторінка, яка відображатиме всю додану раніше навчальну інформацію для вивчення. Також, окремо розташовуватиметься кнопка для проходження тестування, лише за умови, що користувач який створював тему додав та відредагував тестування. Компонентами саме тут буде лише кнопка повернення на попередню сторінку, тобто, навчальні матеріали та кнопка яка переадресує користувача на іншу сторінку для початку проходження тестування.

Сторінка проходження тестування вибраної теми. Сторінка, яка відкриватиметься для проходження користувачам. Як було сказано в попередньому пункті — доступ можна отримати натиснувши відповідну кнопку на сторінці теми. Складатися сторінка буде з:

* блок з текстовою інформацією питання;
* кнопки переходу до наступного/попереднього запитання;
* поле для введення відповіді;
* кнопка завершення тестування та отримання результату.

Саме таким чином виглядає структура даної веб-системи. Візуально діяльність можна переглянути на рисунку 2.2.

Дуже чітко та зрозуміло продемонстровано діяльність користувачів, компонентів та зв’язки між сторінками та процесами. Таким чином, детальний опис структури веб-системи та її компонентів забезпечує чітке уявлення про дії користувача та взаємозв’язки між різними функціональними сторінками та компонентами, що сприяє ефективному впровадженню та розвитку системи.

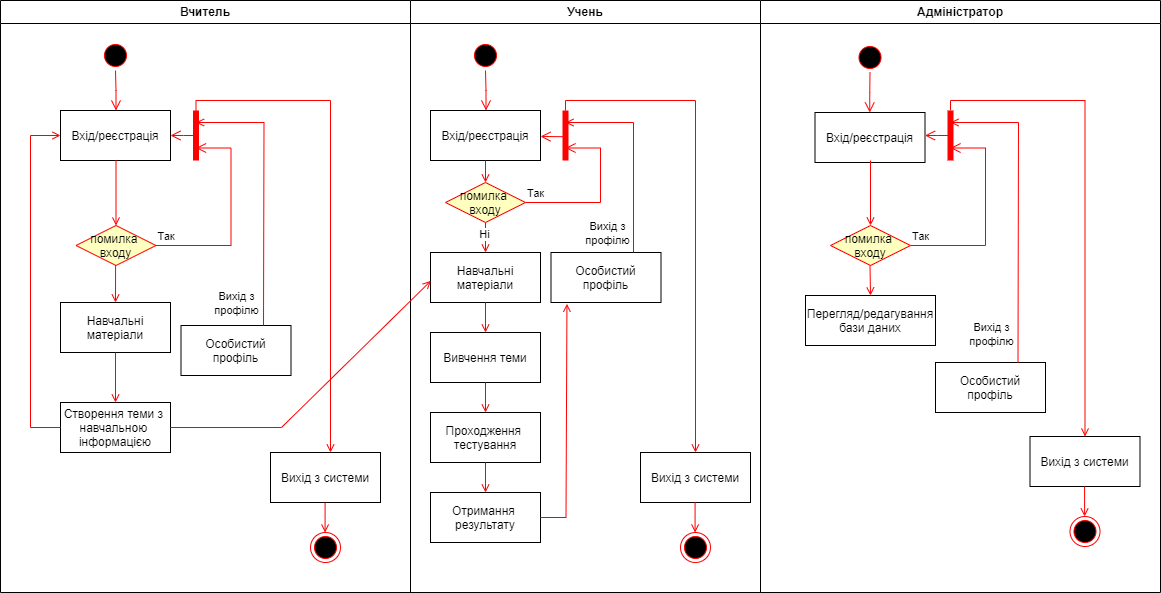


Рисунок 2.2 — Діаграма діяльності веб-системи

2.3 Аналіз технологій реалізацій

Технологій для вирішення схожих проблем існує безліч, але деякі з них виділяються серед всіх. Деякі своєю простотою в використанні, деякі в легкості вивчення, якісь великою кількістю додаткових бібліотек. Проте, не варто використовувати будь що, окремі технології потребують спеціальних знань та можуть застосовуватись лише для обмеженого списку завдань.

Для даної системи потрібно простота та ефективність. Ще одним показником при виборі засобу реалізації інтерфейсу буде можливість використовувати компоненти для мобільної адаптації.

HTML — це мова, яка використовується для створення вебсторінок: вона керує їх структурою та вмістом. Гіпертекст відноситься до текстової інформації, яка пов’язана з іншими текстами за допомогою посилань, сплітаючи цю взаємопов’язану павутину сторінок.



Рисунок 2.3 — Логотип HTML5

Мова розмітки допомагає браузеру розпізнавати текстову інформацію та подавати її читачеві в зручному вигляді. Існують інші мови гіпертекстової розмітки, але переважна більшість сторінок в Інтернеті написані в HTML.

CSS (абревіатура від Cascading Style Sheets, що в перекладі означає каскадні таблиці стилів) - це спеціальна мова (мова стилів), за допомогою якої описують вигляду документів (як і де відображати елементи вебсторінки), написаних мовами розмітки даних. Найчастіше CSS використовується для документів, котрі розмічені мовою HTML, XHTML та XML.



Рисунок 2.4 — Логотип CSS

Одна з головних переваг використання CSS - це можливість розділити зміст сторінки від її оформлення. Таке розділення дозволило покращити сприйняття та доступність змісту, забезпечити більшу гнучкість та контроль за відображенням змісту в різних умовах, зробити зміст більш структурованим та простим, прибрати повторення та ін. Власне це ж і була основна мета створення цієї технології.

Bootstrap – це відкритий та безкоштовний HTML, CSS та JS фреймворк для швидкої верстки адаптивних дизайнів сайтів та WEB-додатків.



Рисунок 2.5 — Логотип BootStrap5

Bootstrap є дуже популярним серед розробників. Це пов'язано з тим, що він дозволяє верстати сайти в кілька разів швидше, ніж на чистому CSS і JavaScript. Він є добре спроектованим фронтенд фреймворком, який досить просто можна налаштувати під себе за допомогою редагування Sass змінних та використання міксинів. Навіть початківець   (без глибоких знань і достатньої практики) зможе швидко створювати досить якісні макети.

Фреймворк Bootstrap являти собою набір CSS та JavaScript файлів. Щоб його використовувати ці файли, необхідно просто підключити їх до сторінки або включити аналогічний функціонал з CDN (Content Delivery Network).

Перевагами Bootstrap є швидкість розробки та гнучкість. Фреймворк надає багато можливостей для налаштування дизайну і поведінки елементів вебсторінки та дозволяє швидко створювати стилізовані і адаптивні вебсторінки за рахунок готових компонентів і шаблонів. Також, Bootstrap має відкритий вихідний код, який доступний на Github. Він має ліцензію MIT. Це означає, що його можна безкоштовно використовувати як у відкритих, так і комерційних застосунках.

Недоліками ж будуть: залежність від фреймворка та збільшення розміру файлів. Використання Bootstrap може призвести до того, що веб-сайт буде схожий на інші сайти, що також використовують цей фреймворк. Також, включення всіх необхідних файлів Bootstrap може призвести до збільшення обсягу сторінки і збільшення часу завантаження.

Node.js — це однопоточне кросплатформове середовище виконання з відкритим вихідним кодом і бібліотека, яка використовується для запуску вебдодатків, написаних на JavaScript, поза браузером клієнта.



Рисунок 2.6 — Логотип Node.js

Простіше кажучи, Node.js — це програмне середовище, яке дозволяє запускати програми, написані мовою Javascript, поза браузером. Історично програми, написані на Javascript, на відміну від інших мов програмування, можна було запустити лише у браузерах, які мали спеціальний вбудований движок виконання коду цієї мови. Поза браузером Javascript, можна сказати, не працював.

Переваги Node.js серед інших технологій є масштабованість та використання JavaScript. Велика кількість розробників вже володіє знаннями JavaScript, що полегшує вивчення Node.js. Також надається можливість створювати швидкі і масштабовані веб-додатки завдяки своїй асинхронності і подієвому принципу.

Недоліками в основному є однопотоковість та велике споживання пам'яті. Деякі застосунки на Node.js можуть споживати більше пам'яті через асинхронний стиль програмування, а оскільки він однопотоковий то він може виявитися неефективним для обробки великої кількості одночасних запитів.

2.4 Проєктування інтерфейсу

Зважаючи на досвід проаналізованих раніше наявних рішень та програмних забезпечень які використовувалися чи використовуються зараз в навчальних цілях, мінімізувати кількість кроків для виконання тої чи іншої дії є в пріоритеті.

Почнемо з проєктування інтерфейсу сторінок для входу (рисунок 2.7) та реєстрації (рисунок 2.8). Оскільки користувач не зацікавлений залишатися довго на цьому етапі, розташування функціонала повинне бути разом. Обов’язково на кожній потрібно розмістити кнопки які переадресовують користувача на іншу сторінку, тобто з реєстрації на вхід і навпаки.

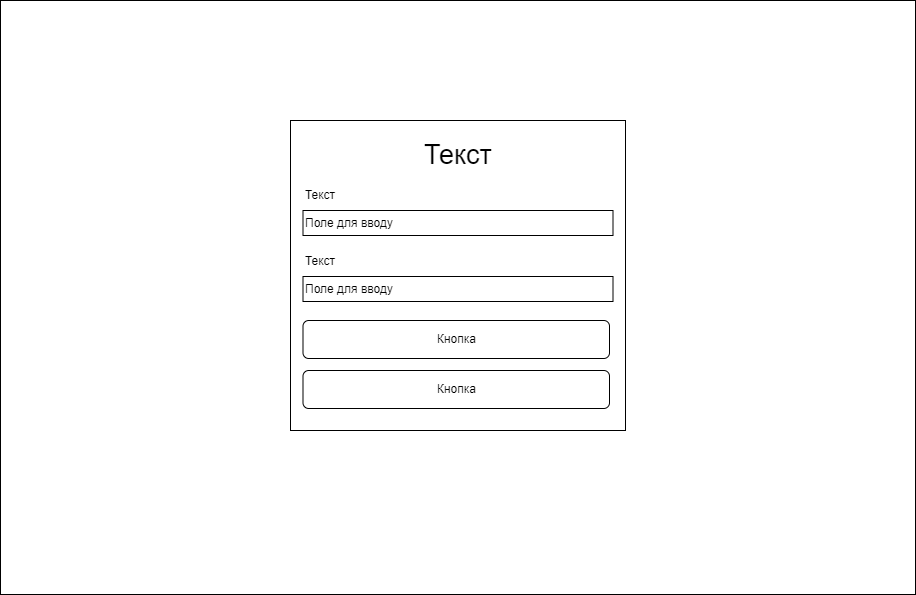


Рисунок 2.7 — Макет сторінки входу

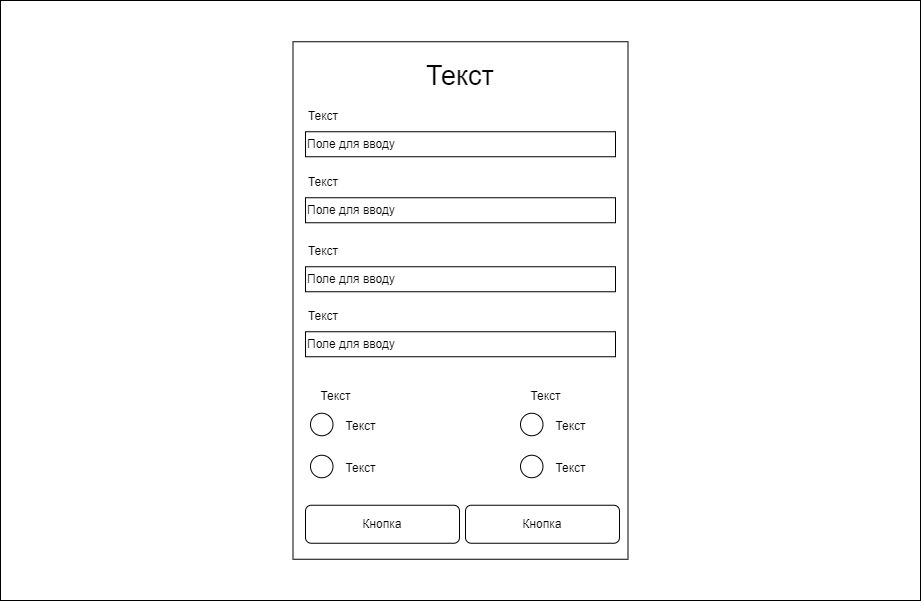


Рисунок 2.8 — Макет сторінки реєстрації

На всіх наступних сторінках буде доступним панель керування для переміщення між сторінками.

Особистий профіль, сторінка міститиме особисту інформацію користувача (ім’я, прізвище, логін), також групу до якої належить користувач, це може бути адміністратор або простий користувач. Попередньо визначено, що додаткова інформація про пройдені тестування як окремий блок. Виглядатиме це наступним чином (рисунок 2.9).

Сторінка навчальні матеріали повинна містити найнеобхіднішу інформацію. Розміщення елементів, таких як, навігаційна панель та розділи з темами, повинні спрощувати процес користування веб-системою. Макет виглядатиме так, як продемонстровано на рисунку 2.10.

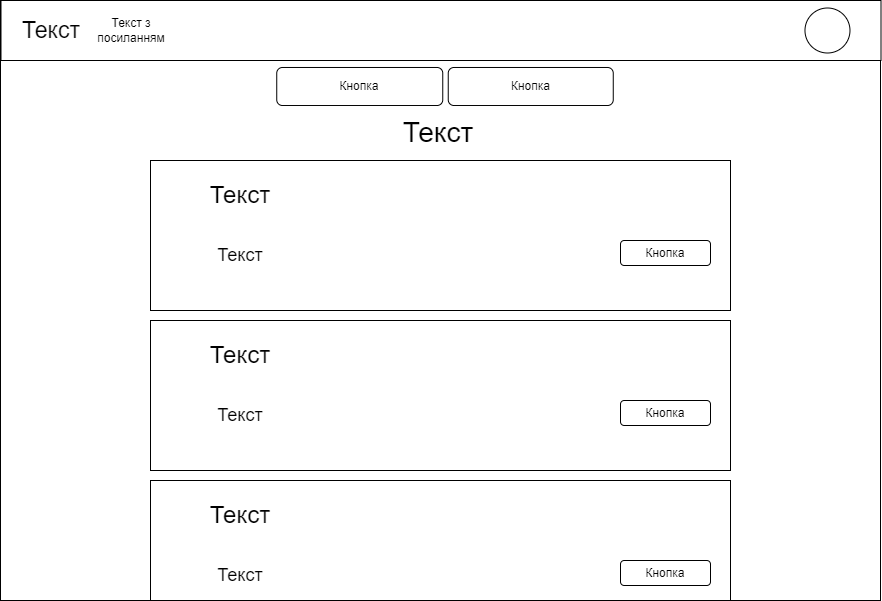


Рисунок 2.10 — Макет сторінки «перелік тем»

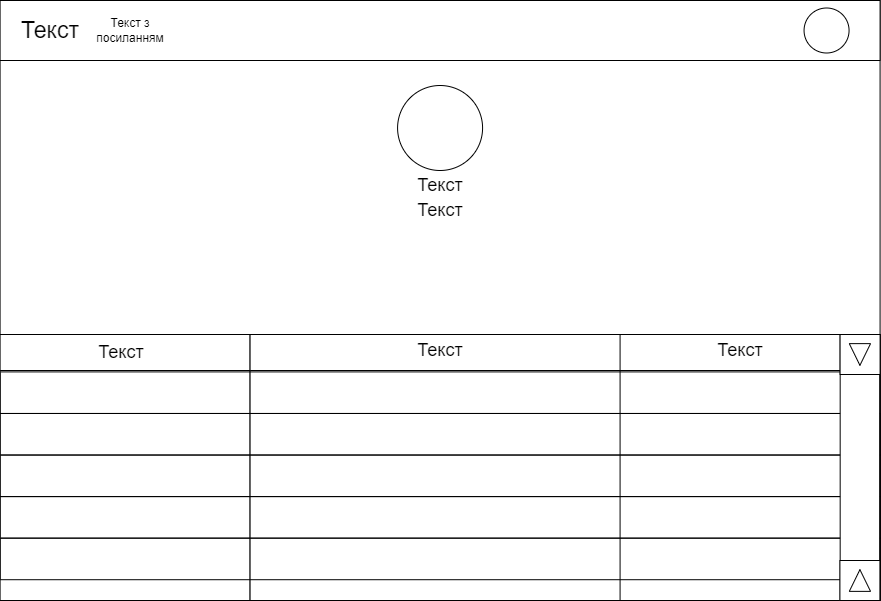


Рисунок 2.9 — Макет сторінки особистого профілю

Сторінка перегляду теми міститиме інформацію додану іншим користувачем. Кнопка повернення до попередньої сторінки, навігаційна панель відіграватимуть роль інструмента переміщення. Додаткова кнопка для перенаправлення на іншу сторінку — проходження тестування по вибраній тему, розміщуватиметься на початку. Це зроблено для тих випадків, коли користувач вже пройшов матеріал або бажає почати тестування одразу. Макет сторінки виглядатиме наступним чином (рисунок 2.11).

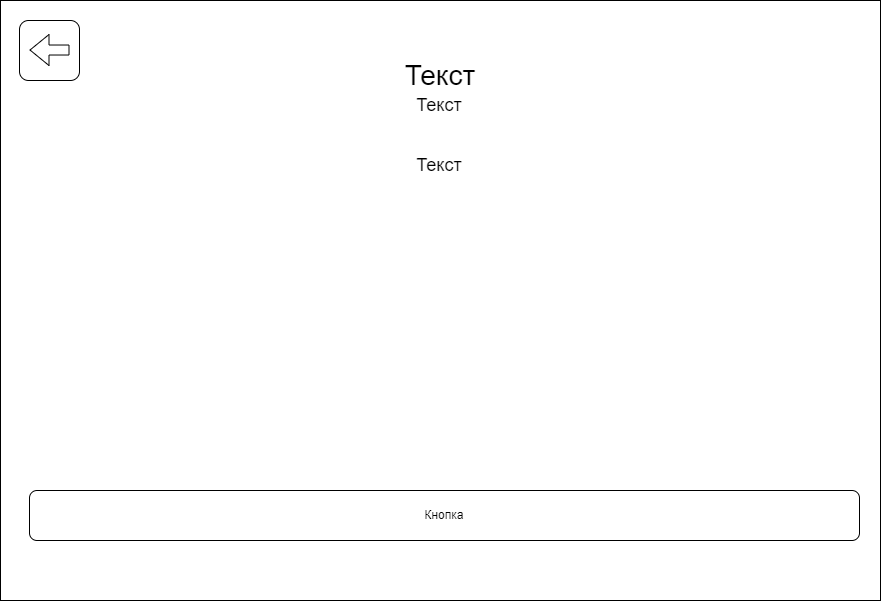


Рисунок 2.11 — Макет сторінки теми

Сторінка проходження тестів міститиме текст питання та функціональні кнопки для взаємодії. Також поле для вводу відповіді на відповідне питання. В будь який момент користувач матиме можливість натиснути кнопку для завершення тестування. Макет виглядатиме наступним чином (рисунок 2.12).

Сторінка створення тестувань та теми міститиме функціонал з використанням введення користувацького тексту. Елементи розміщуватимуться максимально компактно та виглядатимуть наступним чином (рисунок 2.13).



Рисунок 2.13 — Макет сторінки створення теми та тестування

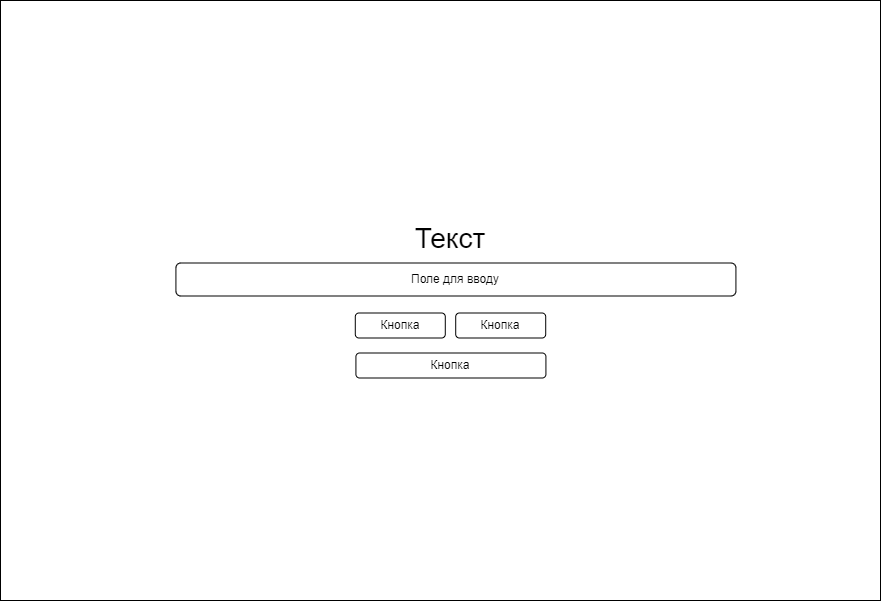


Рисунок 2.12 — Макет сторінки проходження тестування

Для зручності, варто спроєктувати сторінку, на якій відображатимуться всі теми, які створив певний користувач. Макет дещо схожий з переліком тем, проте матиме в переліку обмежену інформацію (рисунок 2.14).

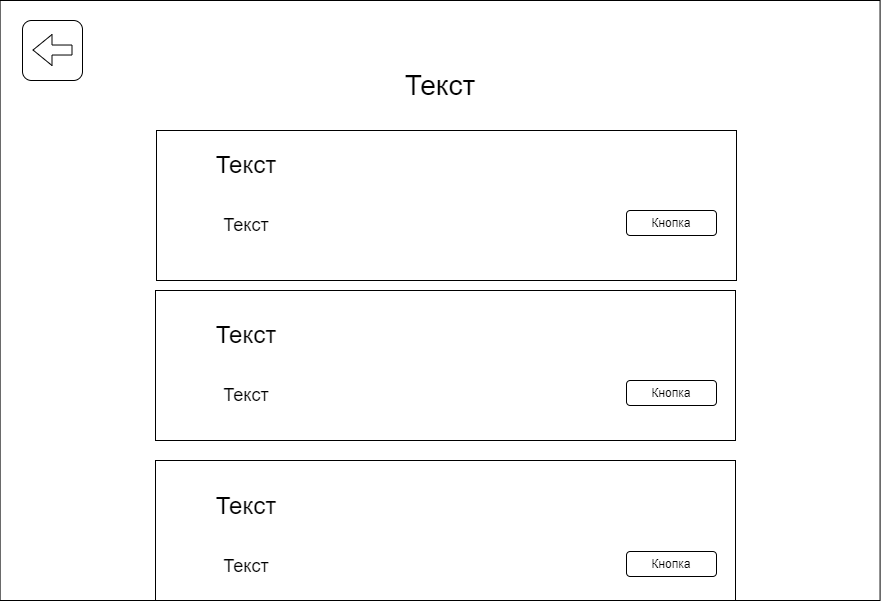


Рисунок 2.14 — Сторінка тем, створених користувачем

Для подальшої взаємодії з створеними темами, користувачу буде запропоновано перейти на наступну сторінку (рисунок 2.15), та в ручну відредагувати тему.

Для того, щоб користувач мав змогу налаштувати деякі дані зв’язані з його особистим профілем, буде реалізована сторінка налаштувань особистого профілю (рисунок 2.16). На якій буде розміщено ряд функцій для змінення тої чи іншої характеристики користувача.

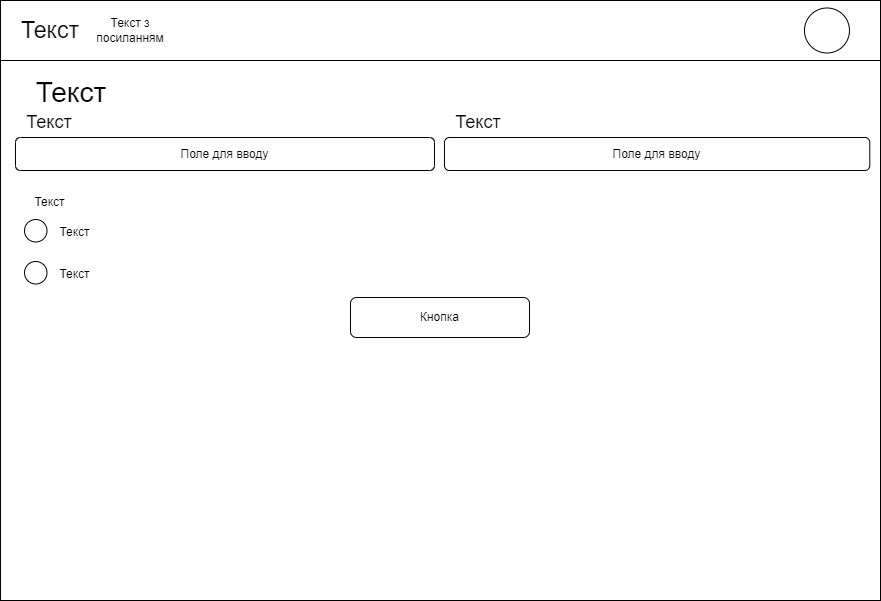


Рисунок 2.15 — Сторінка налаштувань особистого профілю



Рисунок 2.16 — Сторінка редагування теми

Після проведеного дослідження та проєктування, створено макети сторінок для веб-системи, які в подальшому будуть використовуватися в реалізації програмним кодом.

2.5 Проєктування бази даних

При розробці будь-якого інформаційного системи ключовим етапом є проєктування бази даних, яке визначає основну структуру, організацію та зв'язки між даними. Ціллю є забезпечення бази даних, яка буде надійним фундаментом для роботи та реалізації всієї системи забезпечуючи її стабільність, продуктивність та гнучкість.

Цим етапом передбачено створення таблиць та полів, які взаємодіятимуть між собою один одного та мінімізують кількість потрібних комірок з даними, тобто ті, які відповідатимуть всім вимогам. Далі продемонстровано який зміст міститиме база даних в таблиця 2.1, таблиця 2.2, таблиця 2.3 та таблиця 2.4.

Таблиця 2.1 — Таблиця з даними користувача

|  |  |  |
| --- | --- | --- |
| user | | |
| Назва поля | Тип даних | Ключове поле |
| user\_id | Число | Так |
| user\_name | Короткий текст | Ні |
| user\_surname | Короткий текст | Ні |
| user\_gender | Короткий текст | Ні |
| user\_login | Короткий текст | Ні |
| user\_password | Короткий текст | Ні |
| user\_permition\_level | Число | Ні |

Таблиця 2.2 — Таблиця з даними про спроби проходжень тестувань

|  |  |  |
| --- | --- | --- |
| attempts\_to\_pass\_topic\_test | | |
| Назва поля | Тип даних | Ключове поле |
| attempt\_id | Число | Так |
| user\_id | Число | Ні |
| topic\_test\_id | Число | Ні |
| attempt\_result | Число | Ні |

Таблиця 2.3 — Таблиця з даними про тестування тем

|  |  |  |
| --- | --- | --- |
| topic\_test\_information | | |
| Назва поля | Тип даних | Ключове поле |
| topic\_test\_id | Число | Так |
| topic\_question\_one | Довгий текст | Ні |
| topic\_answer\_one | Довгий текст | Ні |
| topic\_question\_two | Довгий текст | Ні |
| topic\_answer\_two | Довгий текст | Ні |
| topic\_question\_three | Довгий текст | Ні |
| topic\_answer\_three | Довгий текст | Ні |
| topic\_question\_four | Довгий текст | Ні |
| topic\_answer\_four | Довгий текст | Ні |
| topic\_question\_five | Довгий текст | Ні |
| topic\_answer\_five | Довгий текст | Ні |
| topic\_question\_six | Довгий текст | Ні |
| topic\_answer\_six | Довгий текст | Ні |

Таблиця 2.4 — Таблиця з даними про тему

|  |  |  |
| --- | --- | --- |
| topic\_information | | |
| Назва поля | Тип даних | Ключове поле |
| topic\_id | Число | Так |
| topic\_name | Короткий текст | Ні |
| topic\_description | Текст | Ні |
| topic\_creator\_information | Число | Ні |
| topic\_main\_information | Довгий текст | Ні |
| topic\_test\_id | Число | Ні |

Наступним рисунком (рисунок 2.17) продемонстровано як саме взаємопов’язані, раніше показані, таблиці. Вимоги та функціонал не вимагають складних рішень для реалізації саме даної веб-системи.



Рисунок 2.17 — Зв’язки таблиць в базі даних

У висновку цього підрозділу слід зазначити, що проєктування бази даних відіграє ключову роль у забезпеченні ефективності, надійності і гнучкості інформаційних систем. Всі основні кроки проєктування були успішно описані та впроваджені.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

Розділ реалізації та тестування системи є ключовим. В ньому відображено весь процес використання новітніх технологій реалізації функціоналу та полегшення взаємодії користувачем з системою.

3.1 Реалізація інтерфейсу

Інтерфейс для системи, яка прогнозована використовуватись великим колом користувачів, опираючись на можливий брак досвіду роботи з схожими платформами, повинна бути інтуїтивно зрозумілою.

Рішень для побудови інтерфейсу існує безліч, як і варіантів їх використання. Одним із найпростіших для використання фреймворків є BootStrap (рисунок 2.5). Перевагою буде вбудована адаптація інтерфейсу та дизайну під мобільні пристрої. В реалізації даної системи це важливо через те, що переважною більшістю користувачів будуть саме учні, які користуватимуться системою під час уроків.

Передбачаючи, що уроки будуть проводитись не в приміщеннях оснащеними персональними комп’ютерами дуже важливим буде саме адаптування.

Перед цим, використовуючи мову розмітки HTML, потрібно реалізувати, так званий «каркас» системи. Сторінки не будуть мати належного привабливого вигляду, вони міститимуть лише функціональні поля, з якими можна взаємодіяти, приблизний порядок розташування об’єктів такі як фотографія, кнопка, таблиці, поля для вводу та текстові поля.

Для цього, існують групи тегів <form>, які призначені для розміщення функціональних об’єктів, як поле, кнопка і тому подібне. Основними тегами для реалізації будуть:

* Input. Це один із найпоширеніших елементів форм, який застосовується для різних типів полів, залежно від їх призначення.
* Label. Використовується для визначення підпису для поля форми. Це допомагає користувачам легко ідентифікувати елементи введення. Елемент label зазвичай асоціюється з елементом input.
* Select. Створює випадаючий список, дозволяючи користувачам вибирати один або кілька варіантів з переліку.
* Text area. Забезпечує багаторядкове текстове поле, яке часто використовується для збору коментарів або відгуків. Розмір цього поля можна налаштувати за допомогою відповідних атрибутів.
* Button. Створює натискну кнопку, яку можна використовувати для різних дій у формі.
* Dropdown menus. Використовуються для створення списків з варіантами, що випадають, забезпечуючи можливість вибору з декількох опцій.
* Radio buttons. Використовуються для вибору одного варіанту з групи можливих, часто застосовуються для вказання статі або інших одноразових виборів.

Для початку, потрібно використати все вище перераховане та створити сторінку для реєстрації користувача. Розміщуватимуться на сторінці такі елементи:

* Input, для вводу інформації про користувача та можливості отримати серверній частині достатньої кількості інформації для її опрацювання та додавання нового користувача.
* Radio button використовуватиметься на даній сторінці лише для того, щоб користувач мав змогу вказати стать.
* Текстові поля, для надання додаткової інформації, що допоможе користувачеві зорієнтуватись в діях.
* Button, для двох завдань, підтвердження реєстрації користувача та переходу на сторінку входу, за умови наявності в користувача профілю.

Без використання CSS та фреймворку Bootstrap, розміщення елементів дещо відрізнятиметься, тому сторінка реєстрації виглядатиме як на рисунку 3.1.

Далі, продемонстровано код даної сторінки.

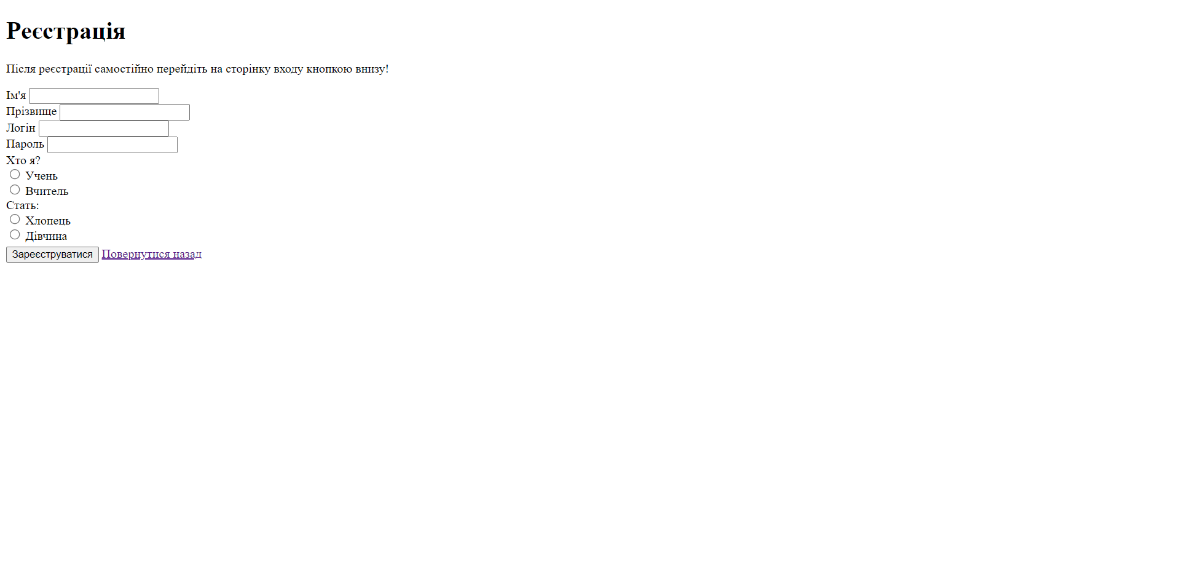


Рисунок 3.1 — Вигляд сторінки «реєстрації» з використанням форм

Надпис та підказка для користувача:

<h1 class="large-h1">Реєстрація</h1>

            <p>Після реєстрації самостійно перейдіть на сторінку входу кнопкою внизу!</p>

Поля для вводу даних користувача:

<div class="mb-3">

                <label for="" class="form-label-custom">Ім'я</label>

                <input type="text" class="form-control" name="accName" id="accName" aria-describedby="helpId" placeholder="" />

            </div>

            <div class="mb-3">

                <label for="" class="form-label-custom">Прізвище</label>

                <input type="text" class="form-control" name="accSurname" id="accSurname" aria-describedby="helpId" placeholder="" />

            </div>

            <div class="mb-3">

                <label for="" class="form-label-custom">Логін</label>

                <input type="text" class="form-control" name="accLogin" id="accLogin" aria-describedby="helpId" placeholder="" />

            </div>

            <div class="mb-3">

                <label for="" class="form-label-custom">Пароль</label>

                <input type="password" class="form-control" name="accPassword" id="accPassword" aria-describedby="helpId" placeholder="" />

            </div>

Вибір типу користувача. В подальшому це впливатиме на можливість створювати та редагувати теми та тести:

<div class="form-check">

                            Хто я?<br>

                            <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault1" value="Учень">

                            <label class="form-check-label" for="flexRadioDefault1">

                              Учень

                            </label>

                          </div>

                          <div class="form-check">

                            <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault2" value="Вчитель">

                            <label class="form-check-label" for="flexRadioDefault2">

                              Вчитель

                            </label>

                          </div>

Вибір статі користувача:

                        <div class="form-check">

                            Стать:<br>

                            <input class="form-check-input" type="radio" name="flexRadioDefault1" id="flexRadioDefault11" value="Хлопець">

                            <label class="form-check-label" for="flexRadioDefault1">

                              Хлопець

                            </label>

                          </div>

                          <div class="form-check">

                            <input class="form-check-input" type="radio" name="flexRadioDefault1" id="flexRadioDefault22" value="Дівчина">

                            <label class="form-check-label" for="flexRadioDefault2">

                              Дівчина

                            </label>

                          </div>

Кнопки:

                        <button style="margin-top: 5px;" type="submit" class="btn btn-primary" onclick="registerUser()">

                            Зареєструватися

                        </button>

                    <a style="margin-top: 5px;" href="./login\_page.html" class="button btn btn-primary">

                        Повернутися назад

                    </a>

Деяка кнопка матиме функціонал переходу на сторінку входу, як було вказано раніше, тому потрібно створити цю саму сторінку для входу в існуючий профіль. На даній сторінці міститиметься всього лиш два поля для вводу даних, текстові поля та кнопки. Виглядатиме сторінка, як на рисунку 3.2.

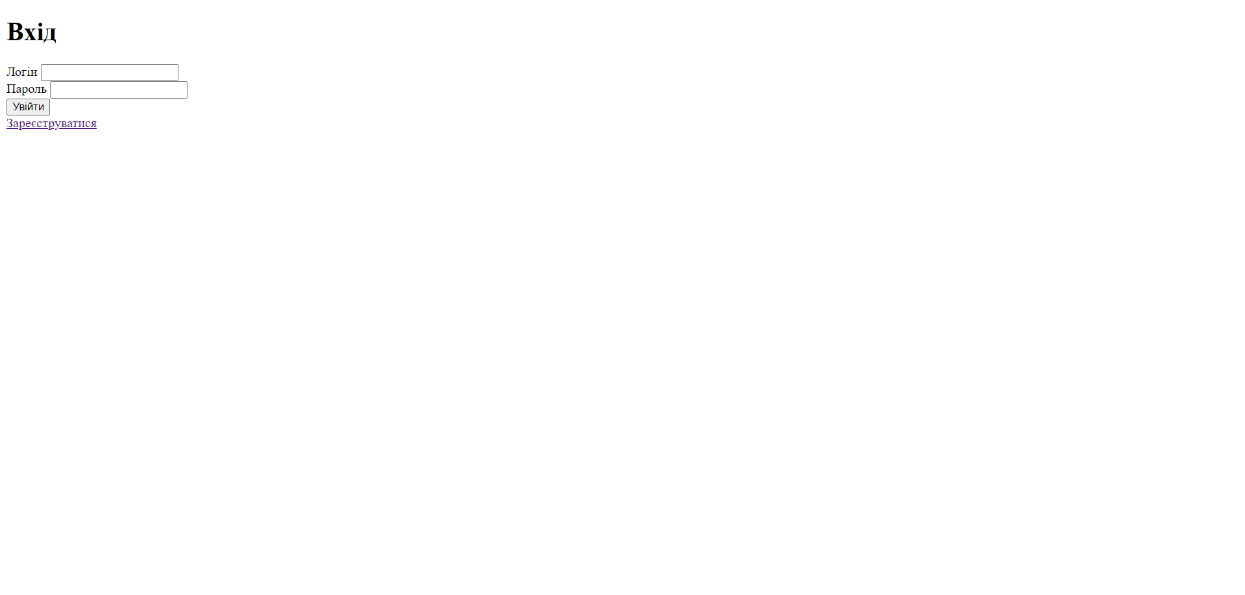


Рисунок 3.2 — Вигляд сторінки «входу» з використанням форм

Код текстового поля:

<h1 class="large-h1">Вхід</h1>

Код полів для вводу даних:

        <div class="mb-3">

            <label for="" class="form-label-custom">Логін</label>

            <input type="text" class="form-control" name="" id="input\_userLogin" aria-describedby="login\_input" placeholder=""/>

        </div>

        <div class="mb-3">

            <label for="" class="form-label-custom">Пароль</label>

            <input type="password" class="form-control" name="" id="input\_userPassword" aria-describedby="password\_input" placeholder=""/>

        </div>

Код кнопок:

                <button type="submit" class="btn btn-primary" onclick="send\_login\_request()">

                    Увійти

                </button>

                <div class="form-button-custom1">

                    <a href="./register\_page.html" class="button btn btn-primary">

                        Зареєструватися

                    </a>

Відповідно до підпису на кожній з кнопок, «Увійти» матиме функціонал для підтвердження введених даних та входу в систему, а «Зареєструватися», переадресовуватиме користувача на сторінку «реєстрації».

Наступною сторінкою, після входу користувача, буде «особистий профіль». Згідно макету, розміщуватимуться таблиці, кнопки, текстові поля та елементи, з якими можна взаємодіяти. Виглядатиме сторінка, як на рисунку 3.3.

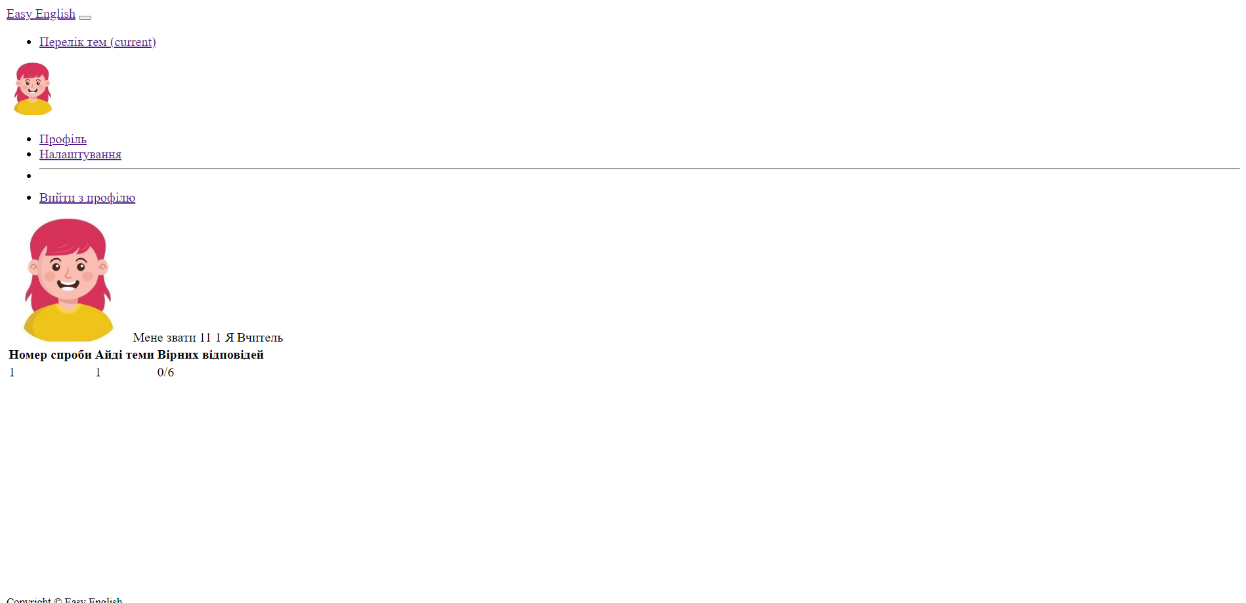


Рисунок 3.3 — Вигляд сторінки профілю з використанням форм

На деяких з наступних сторінок, включаючи дану, буде повторюватись код, що відповідатиме за навігаційну панель. Розміщення, функціонал та зовнішній вигляд практично не відрізнятиметься, тому наведення прикладу коду буде лише раз.

Код навігаційної панелі:

<nav

        class="navbar navbar-expand-sm navbar-dark bg-dark"

    >

        <div class="container">

            <a class="navbar-brand" href="#">Easy English</a>

            <button

                class="navbar-toggler d-lg-none"

                type="button"

                data-bs-toggle="collapse"

                data-bs-target="#collapsibleNavId"

                aria-controls="collapsibleNavId"

                aria-expanded="false"

                aria-label="Toggle navigation"

                aria-current="page"

                >

                <span class="navbar-toggler-icon"></span>

            </button>

            <div class="collapse navbar-collapse" id="collapsibleNavId">

                <ul class="navbar-nav me-auto mt-2 mt-lg-0">

                    <li class="nav-item">

                        <a class="nav-link fs-6 text" href="./topics\_list\_page.html" aria-current="page"

                            >Перелік тем

                            <span class="visually-hidden">(current)</span></a

                        >

                    </li>

                </ul>

                <div class="flex-shrink-0 dropdown">

                    <a href="#" class="d-block link-dark text-decoration-none dropdown-toggle" id="dropdownUser2" data-bs-toggle="dropdown" aria-expanded="false">

                      <img id="nav-profile-picture" src="" alt="mdo" width="64" height="64" class="rounded-circle">

                    </a>

                    <ul class="dropdown-menu text-small shadow" aria-labelledby="dropdownUser2">

                      <li><a class="dropdown-item" href="./user\_profile.html">Профіль</a></li>

                      <li><a class="dropdown-item" href="./profile\_settings.html">Налаштування</a></li>

                      <li><hr class="dropdown-divider"></li>

                      <li><a class="dropdown-item" onclick="clearSession()" href="./login\_page.html">Вийти з профілю</a></li>

                    </ul>

                  </div>

            </div>

        </div>

    </nav>

Код особистої інформації користувача, виключаючи елементи як фотографію та текстове поле:

                    <img id="profilePicture" class="rounded-circle mt-5" width="150px" src="/images/female\_pic.png" alt="Profile Picture">

                <span class="text-black-50">Мене звати 11 1 span>\

                <span class="text-black-50">Я Вчитель</span>

Код таблиці:

<table class="table table-striped">

    <thead>

        <tr>

            <th>

                Номер спроби

            </th>

            <th>

                Айді теми

            </th>

            <th>

                Вірних відповідей

            </th>

        </tr>

    </thead>

    <tbody class="table">

    <tr>

        <td>1</td>

        <td>1</td>

        <td>0/6</td>

    </tr>

    </tbody>

</table>

Наступним кроком, буде створення сторінки з можливістю налаштувати деяку особисту інформацію, таку як ім’я, прізвище та стать. Також, використовуватиметься навігаційна панель. Виглядатиме сторінка як на рисунку 3.4.

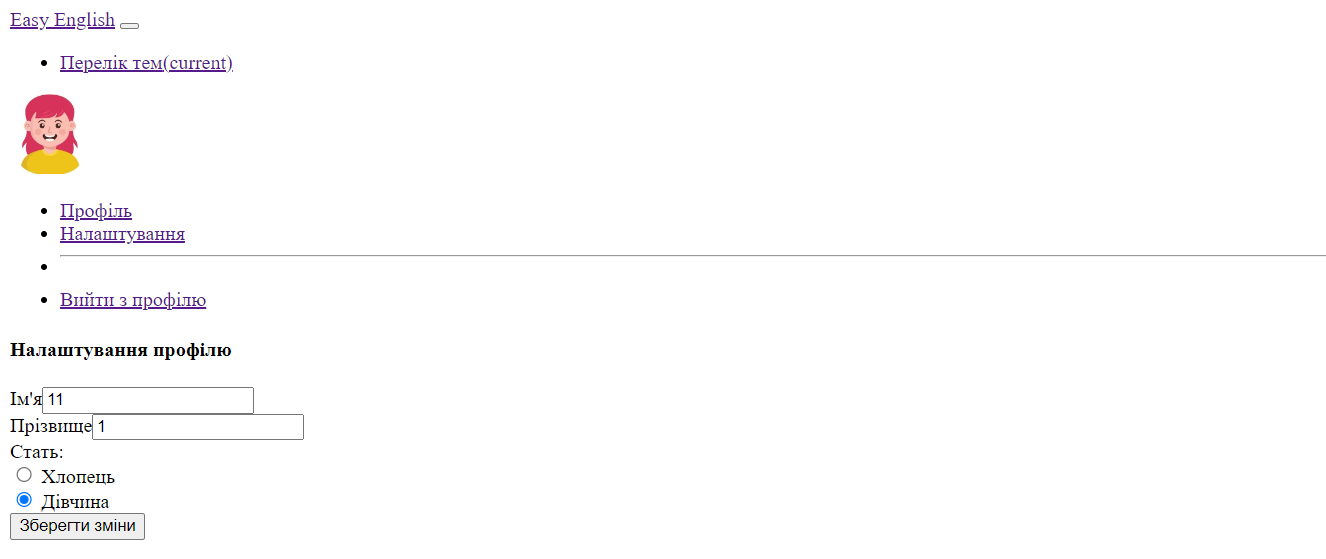


Рисунок 3.4 — Вигляд сторінки «налаштування профілю» з використанням форм

Розміщуватиметься на даній сторінці текстові поля, поля для вводу, зображення та кнопки.

Код полів для вводу:

                    <div class="col-md-6"><label class="labels">Ім'я</label><input type="text" id="nameforchange" class="form-control" placeholder="first name" value=""></div>

                    <div class="col-md-6"><label class="labels">Прізвище</label><input type="text" id="surnameforchange" class="form-control" value="" placeholder="surname"></div>

Код вибору статі:

                        <div class="form-check">

                            Стать:<br>

                            <input class="form-check-input" type="radio" name="flexRadioDefault1" id="flexRadioDefault11" value="Хлопець">

                            <label class="form-check-label" for="flexRadioDefault11">Хлопець</label>

                        </div>

                        <div class="form-check">

                            <input class="form-check-input" type="radio" name="flexRadioDefault1" id="flexRadioDefault22" value="Дівчина">

                            <label class="form-check-label" for="flexRadioDefault22">Дівчина</label>

                        </div>

Код кнопки:

                <button class="btn btn-primary profile-button" type="button" data-bs-toggle="modal" data-bs-target="#exampleModal">Зберегти зміни</button>

Таким чином, користувач в майбутньому матиме змогу змінити, не важливі для функціонування системи, дані самостійно використовуючи дану сторінку.

Для перегляду всіх існуючих тем, потрібно створити відповідну сторінку з переліком. Розміщуватимуться текстові поля, кнопки, зображення, текст з посиланням та навігаційна панель. Виглядатиме сторінка, як на рисунку 3.5.

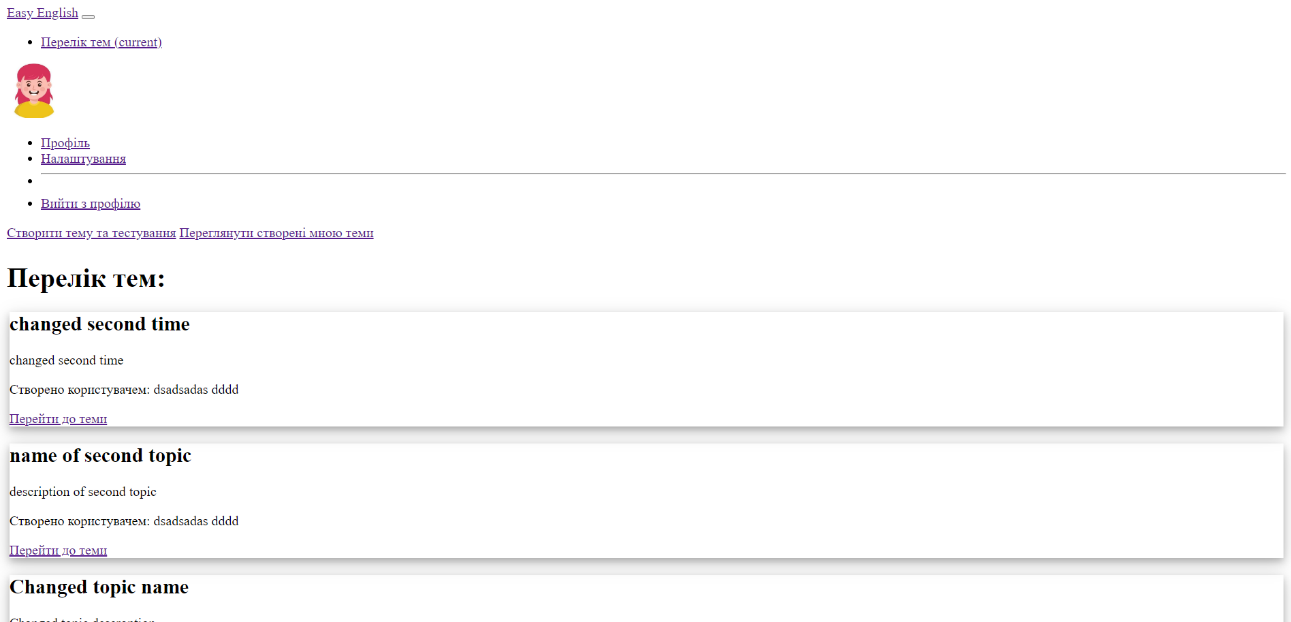


Рисунок 3.5 — Вигляд сторінки «перелік тем» з використанням форм

Код текстового поля:

    <h1 class="text-center" style="margin-top: 25px;">Перелік тем:</h1>

Наступний продемонстрований код має лише прикладний характер, в подальшій розробці буде реалізовано повноцінний код з функціоналом автоматичного виводу всіх тем без необхідності робити цього в ручну адміністраторами системи. Код блоку з інформацією про тему:

                    <div class="container">

                        <div style="margin-left: 3px; margin-right: 3px; margin-top: 10px; box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);" class="row py-5">

                            <div class="col-md-9">

                                <div class="lc-block">

                                    <div editable="rich">

                                        <h2><strongназва теми</strong></h2>

                                        <p>підпис теми</p>

                                        <p>Створено користувачем: автор теми</p>

                                    </div>

                                </div>

                            </div>

                            <div class="col-md-3 align-self-center text-center">

                                <div class="lc-block">

                                    <a class="btn btn-link btn-lg" href="./topic\_information.html?topicInformationIdайді теми" role="button">Перейти до теми</a>

                                </div>

                            </div>

                        </div>

                    </div>

Код кнопок:

                    <a style="margin-top: 10px" class="btn btn-primary" href="./create\_topic\_and\_test\_page.html" role="button">Створити тему та тестування</a>

                    <a style="margin-top: 10px" class="btn btn-primary" href="./edit\_created\_topics.html" role="button">Переглянути створені мною теми</a>

Користувач матиме можливість перейти до наступної сторінки клацнувши відповідну кнопку. З можливих: створення теми, перегляд створених користувачем тем та перехід до сторінки з інформацією теми.

Для функціонування системи, потрібна сторінка для створення цих самих тем. Сторінка матиме лише достатню кількість полів що відповідатимуть полям в базі даних та функціональну можливість додавати дані. «створення теми» виглядатиме як на рисунку 3.6.

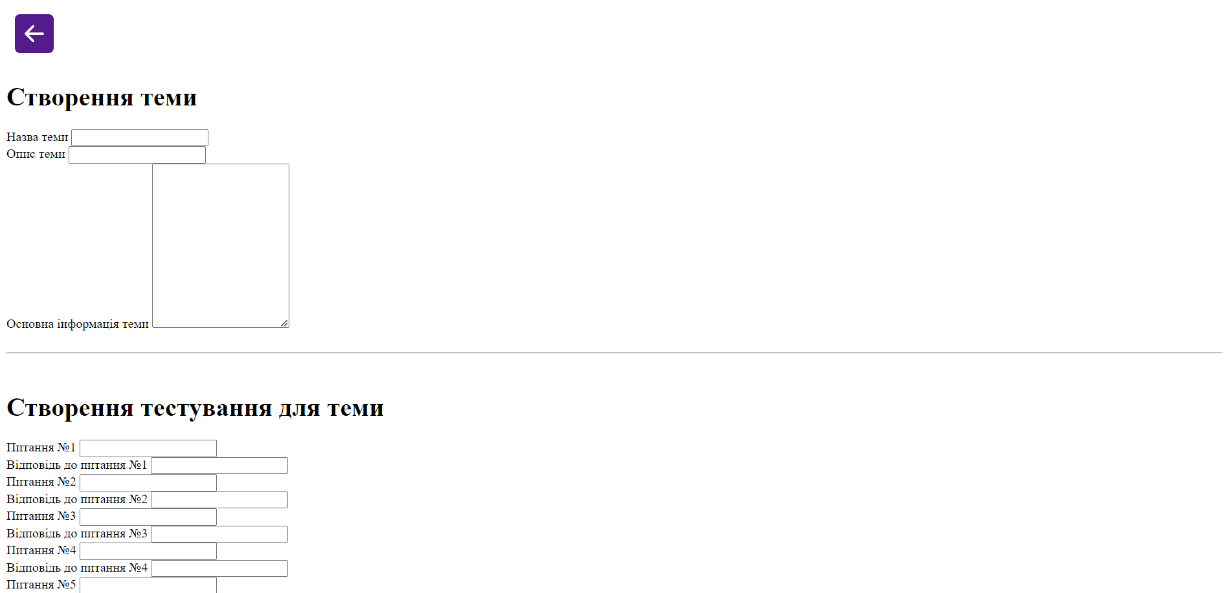


Рисунок 3.6 — Вигляд «створення теми» з використанням форм

3.2 Реалізація функціоналу

3.3 Реалізація бази даних

3.4 Тестування системи