

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

## **ВИПУСКНА РОБОТА**

**на тему:**

**«Створення сайту для загальноосвітньої школи  
№1 м. Середини-Буди»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Тиркусова Н.В.**

**Студента групи ІІІ – 64-8**

**Шуст С.І.**

**СУМИ 2020**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

**Кафедра комп'ютерних наук**

Затверджую \_\_\_\_\_

Зав. кафедрою Довбиш А.С.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 г.

### **ЗАВДАННЯ**

#### **до випускної роботи**

Студента четвертого курсу, групи ІН-64-8 спеціальності “Інформатика”  
денної форми навчання Шуста Сергія Івановича.

**Тема: “Створення сайту для загальноосвітньої школи № 1 м.Середини-  
Буди”**

Затверджена наказом по СумДУ

№ \_\_\_\_\_ от \_\_\_\_\_ 2019 г.

**Зміст пояснювальної записки:** 1) інформацій огляд 2) вибір  
методу рішення; 3) програмна реалізація проекту.

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 2019 г.

Керівник випускної роботи \_\_\_\_\_ Тиркусова Н.В.

Завдання прийняв до виконання \_\_\_\_\_ Шуст С.І.

## РЕФЕРАТ

**Записка:** 38 стор., 12 рис., 1 додаток, 27 джерел.

**Об'єкт дослідження** – веб сайт для загальноосвітньої школи № 1 м.Середини-Буди”.

**Мета роботи** – розробка веб сайту для загальноосвітньої школи № 1 м.Середини-Буди”.

**Методи дослідження** – в процесі програмної реалізації проекту було застосовано стек технологій для побудови веб сайту (React,Node.js, Express, MongoDB).

**Результати** – створено веб сайт для загальноосвітньої школи № 1 м.Середини-Буди. Реалізовано повний функціонал для ознайомлення з статтями, їх створення та редагування. Після завершення текстів проект був завантажений на хостинг.

ВЕБ-САЙТ ЗОШ №1 WEB,REACT,NODE.JS, EXPRESS,  
MONGODB.

## Зміст

|  |    |
|--|----|
| <b>Вступ</b> .....                             | 5  |
| <b>1 Інформаційний огляд</b> .....             | 7  |
| <b>1.1 Огляд існуючих рішень</b> .....         | 7  |
| <b>1.2 Постановка задачі</b> .....             | 8  |
| <b>2 Вибір методу рішення</b> .....            | 10 |
| <b>2.1 Огляд методів розробки</b> .....        | 10 |
| <b>2.2 Вибір мови програмування</b> .....      | 13 |
| <b>2.3 Вибір фреймворку</b> .....              | 14 |
| <b>2.4 Вибір середовища розробки</b> .....     | 15 |
| <b>2.5 Вибір бази даних</b> .....              | 16 |
| <b>2.6 Вибір середовища розміщення</b> .....   | 17 |
| <b>2.7 Вибір системи контролю версій</b> ..... | 18 |
| <b>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ</b> .....    | 19 |
| <b>3.2 Опис функцій</b> .....                  | 19 |
| <b>3.3 Структура проекту</b> .....             | 21 |
| <b>3.2 Результат виконаної роботи</b> .....    | 22 |
| <b>ВИСНОВКИ</b> .....                          | 30 |
| <b>СПИСОК ДЖЕРЕЛ</b> .....                     | 31 |
| <b>Додаток А</b> .....                         | 34 |
| <b>Додаток В</b> .....                         | 38 |

## Вступ

Ми живемо в такий час, де вже приватна компанія запускає в космос ракету з екіпажем. І в цю епоху розвитку технологій кожен поважаючий себе освітній заклад повинен мати свій власний сайт, котрий доказує що заклад йде в ногу з часом.

Сьогодні самою поширеною інформаційною технологією являється Інтернет. Беззаперечно він саме глобальне і продуктивне джерело інформації. Для більш кращого ознайомлення з інформацією і новинами, навчальному закладу потрібен свій персональний сайт, де учні та їх батьки можуть з ними ознайомитися.

Користувачам набагато приємніше відвідувати веб сайти які не вводять в ступор своїм оформленням, швидко завантажуються і правильно відображаються у вікні Web-браузера.

Головною метою проекту це розробка веб сайту для ЗОШ №1 на якому учні зможуть отримувати актуальну інформацію що до шкільних оголошень та перегляд головних новин із життя школи. Також однією причиною створення оновленого шкільного сайту було вдосконалення UI та UX, оскільки в попередньому сайті було занадто багато зайвих віджетів, а також мобільна версія не була адаптивною, а зараз на наш час більша кількість користувачів користуються переважно смартфонами, адаптація сайту під мобільні пристрої є просто необхідною [1].

Практична значимість проекту – це створення сучасного і простого використанні сайту для ЗОШ №1. А попередній сайт закладу вже устарів для теперішніх користувачів і його потрібно замінити на більш сучасний.

Шкільний сайт – це обличчя школи в Інтернеті. Він показує що заклад використовує всі можливості сучасних інформаційних технологій. Свій власний сайт піднімає імідж закладу в очах оточуючих. Сайт школи це головний аспект для всіх хто пов'язані із навчальним закладом.

Актуальність роботи полягає в вільному доступі до інформації яка пов'язана із навчанням і середовищем освіти через веб-додаток, піднімає рівень довіри батьків до закладу і його робітників. Збільшує імідж школи серед інших закладів, що вкрай важливо в умовах сучасної конкуренції між освітніми установами. Також це один із засобів заручитися підтримкою спонсорів. Шкільний сайт це візитна картка школи в інтернеті. Шкільний сайт створює нові можливості взаємодії в освітньому процесі. Сайт корисний для професійної рефлексії школи.

## **1 Інформаційний огляд**

### **1.1 Огляд існуючих рішень**

Основне призначення сайту школи – це можливість зручно переглянути наступні речі:

- Останні новини школи;
- Загальні відомості школи
- Шкільний фотоальбом

Тому, дослідження проводилось саме серед таких сайтів, які мають вище перелічені можливості.

#### **Школа №16 м.Одеса**

Один із великої кількості сайтів шкіл. У нього є наступні можливості:

- Перегляд інформації про заклад;
- Календар подій;
- Перегляд новин;

Недоліки сайту:

- Сайт не має адаптивної версії;
- Сайт виконаний у старому стилі, на якому дуже складно знайти необхідну інформацію.

Недоліки є достатньо складними, оскільки це заважає користувачу шукати потрібну інформацію, та мати можливість доступу до сайту з мобільних пристроїв [2].

#### **Школа №1 м.Суми**

У даного сайту є наступні переваги:

- Можливість авторизації;
- Перегляд новин;
- Застарілий дизайн;

- Корисні посилання.

Недоліки сайту:

- Немає можливості переглядати розклад занять;
- Важко знайти необхідну інформацію;
- Немає адаптивності.

Проаналізовані сайти мають великий перелік переваг, але також і дуже значні недоліки які потрібно виправити у новому сайті для школи. За приклад будуть взяті саме ці сайти [3].

## 1.2 Постановка задачі

Метою роботи є створення зручного сайту для перегляду новин, різної інформації перегляду шкільної галереї.

Основні вимоги до проекту:

- Адаптивність
- Можливість авторизації для перегляду більш детальної інформації;
- Перегляд останніх новин;
- У користувача повинна бути одна з ролей, а саме public, user, moderator, admin.

Адміністратор та модератор повинні мати змогу створювати статті, редагувати та видаляти їх, завантажувати фотографії до галереї сайту. Незареєстровані та зареєстровані користувачі мають змогу лише ознайомлюватися зі статтями та переглядати фотогалерею.

Можливості ролі адміністратора:

- Створення та редагування галереї фотографій;
- Управління користувачами;
- Оновлення, створення, перегляд та видалення статей.

Можливості ролі модератора:



- Оновлення, створення, перегляд та видалення статей;
- Створення та редагування галереї

Можливості ролі зареєстрованого користувача:

- Перегляд статей;
- Перегляд інформації на сайті;
- Перегляд галереї фотографій.

Користувачі повинні мати змогу пройти реєстрацію без довгого очікування підтвердження реєстрації адміністратором чи модератором.

Також сайт має бути адаптивним та кроссбраузерним, для того щоб користувачі могли легко користуватися ним з будь яких пристроїв, та з будь якого браузера.

## 2 Вибір методу рішення

### 2.1 Огляд методів розробки

Можна підкреслити 5 методів створення веб-сайту:

1. Ручний за допомогою звичайних HTML, CSS, JS;
2. За допомогою програмних засобів створення сайтів;
3. На SaaS платформах;
4. За допомогою інструментальних систем таких як CMS;
5. За допомогою фреймворків.

Ручний метод – раніше це був дуже загальноприйнятий метод. Але цей метод має дуже великий мінус, цей спосіб дуже копіткий [4].

Програмні засоби – це вже готові програмні засоби які дають можливість генерувати html код та створювати сайт у візуальному режимі. Серед найвідоміших можна виділити такі як:

- Microsoft FrontPage — це простий і раціональний Web-редактор для проектування, розробки Web-сайтів. Він не складний в вивченні, вистачить декількох годин вільного часу. Аналогічне меню як і в інших програмах Microsoft Office роблять знайомство з програмою значно легше. При цьому FrontPage дозволяє створювати сайти будь-якої складності. Програма дає доступ до широких функціональних можливостей при розробці. А в 2009 році продукт став абсолютно безкоштовним, що є великим плюсом продукту. [5]

- Adobe DreamWeaver — один з потужних HTML-редакторів, що створюють HTML код. Він дозволяє працювати в декількох режимах одночасно, з HTML кодом або у візуальному режимі. Створювати код можна дуже швидко і ефективно при розробці сайту, дизайн підтримує усі існуючі стандарти Інтернет (ASP.NET,Flash і PHP). Для спрощення розробки проектів Adobe Dreamweaver включає в свій вміст велику систему підказок і уроків, також є шаблони і приклади веб-сторінок. Але

одним із основним недоліків є те, що програма генерує надто "складний" код, додаючи багато непотрібного коду. А якщо гарно ознайомлені з HTML, це не є великою складністю тоді текст HTML можна легко відредагувати. Adobe DreamWeaver випускалася до 2005 року компанією Macromedia, після чого була придбана фірмою Adobe. І ціна на продукт стало великою. І це другий недолік - дорога вартість продукту. [6]

CMS — у перекладі з англійської мови (Content Management System). Це система управління контентом сайту, тобто тим що буде знаходитися на сайті. Також її називають системою управління вмістом сайту. Все вище перелічене і є CMS. CMS дає змогу самостійно розробляти і видаляти розділи сайту, робити редагування інформації без підключення стороннього фахівця. Крім того CMS робить можливим прискорити і значно знизити ціну розробку проекту, так як тимчасові витрати на створення багатьох елементів знижуються [7]. Найбільш розповсюдженими CMS системами:

- Drupal — надійна, з великим функціоналом, а головне безкоштовна CMS. Яка підходить в першу чергу для великих порталів і сайтів великих організацій. Наприклад, якщо ви хочете створити якийсь блог, розділи з різноманітними послугами, та інтернет магазин і це все на одному сайті. Має гарний захист від атак хакерів, стабільна система яка вимагає від спеціаліста розвинутих навичок розробки сайтів. Вона підійде, якщо є можливість наняти кваліфікованого веб-розробника[8].

- Joomla — один із простих і легкодоступних движків, на якому можна розробляти майже все що завгодно. Він займає друге місце у світі за поширеністю. На ньому таки досить просто створювати сайти, але все-таки не можна сказати, що цей движок дуже гарний. Він «тонкий» і його дуже легко взломати. Також часто створює непотрібні дублі сторінок, через що може створювати проблеми з розробкою. Адмін панель складна для розуміння для тих хто користується вперше. Але все таки Joomla

підходить для великих проєктів, а також її розширення дають можливість створити інтернет-магазин. Із плюсів має хорошу розширюваність, і відкритість, має великий набір шаблонів та доповнень. Недоліком є високе навантаження на сервер і мала швидкість роботи [9].

- WordPress — дозволяє розробляти проєкти будь-якого типу, так як має гарну розширюваність. На ньому працює кожен четвертий сайт в світовій павутині. І це означає що він популярний движок в світі. Він повністю безкоштовний, для нього розроблено велика кількість модулів. В результаті на WordPress можна зробити що завгодно - як невеликий блог, так і повноцінний інтернет-магазин. Він простий в установці, дозволяє легко змінювати шаблони і теми оформлення, має гарні SEO-характеристики [10].

Фреймворк – це структура програмної системи, іншими словами це програмне забезпечення, що полегшує розробку і об'єднання різних компонентів масштабного програмного проєкту. На відміну від бібліотек, які об'єднують набір підпрограм близькою функціональністю, фреймворк містить в собі велику кількість різних за призначенням бібліотек. Вживається також слово «каркас», а деякі автори використовують його в якості основногор [11].

Фреймворк спрощує розробку і робить написання коду набагато легше. Велика кількість фреймворків спрощують доступ до баз даних, розробку інтерфейсу, а також роблять менше дублювання коду. Для створення фронтенд частини зараз використовують:

- Vue.js — це сучасний фреймворк для розробки користувацьких інтерфейсів. Vue створений придатним для поступового впровадження. Його ядро в першу чергу вирішує завдання рівня уявлення (view), що спрощує інтеграцію з іншими бібліотеками та існуючими проєктами. З іншого боку, Vue повністю підходить і для створення складних

односторінкових додатків якщо використовувати його спільно з сучасними інструментами та додатковими бібліотеками [12];

- React — багаторазові компоненти, це і його основна перевага. Фахівець створює невеликі частини коду, які можна об'єднувати, щоб сформувати більш великі компоненти або використовувати їх як окремі елементи інтерфейсу. А головне це те що компоненти можна використовувати повторно [13];

- Angular 2.X — основною метою якого це створення односторінкових сайтів. Має високу продуктивність, не потребує дуже великих пізнань в програмуванні. Також в світовій павутині є велика кількість джерел стосовно цієї мови програмування, але в основному це іноземні сайти. А головний плюс, це коли при написанні неправильного синтаксису буде виведена підказка [14].

SaaS-платформи для розробки сайтів. SaaS-платформи це сайт з виглядом сервісу. І це означає нам надається не тільки конструктор сайтів а і хостинг для створюваного сайту. Тобто у нас з'являється можливість самостійно створити і розмістити сайт в інтернеті. Це вибір підходить для простих «легких» сайтів, тимчасових проєктів і для сайтів візиток SaaS в останній час активно розвивається та використовується розробниками [15].

Мною було обрано варіант розробити сайт за допомогою фреймворків. Таким чином у мене буде краще управління над проєктом, а якість і швидкість створенного проєкту буде значно більша ніж при написанні сайту вручну.

## **2.2 Вибір мови програмування**

Для створення клієнтської частини сайту я буду використовувати такі мови:

- HTML5 — це нова версія найпопулярнішої мови розмітки в котрій реалізовано безліч нових синтаксичних особливостей [16];
- CSS3 — це нова версія каскадних таблиць стилів CSS. В CSS3 з'явилася величезна кількість нових можливостей, які в багато разів спрощують великий масштаб роботи [17];
- JavaScript — це одна із мов програмування, який дозволяє створити динамічно оновлюваний контент [18].

На відміну від клієнту, сервер можна написати на великій кількості мов програмування. Оскільки для написання клієнтської частини проекту я буду використовувати JavaScript, то було б дуже зручно використовувати аналогічну мову програмування для створення і серверної частини. В мене є така можливість, я буду створювати сервер на Node.js.

З Node.js. можна легко масштабуватися. Іншими словами, при одночасному підключенні до сервера тисячі користувачів Node.js. працює асинхронно, тобто ставить пріоритети і розподіляє ресурси грамотніше. Наприклад Java виділяє на кожне підключення окремий потік, що є не дуже доцільно.

На цій мові можна писати кросплатформені додатки - в зв'язці мобільний телефон і десктоп Node.js допомагає досягати синхронності. Наприклад, коли пишете повідомлення з телефону, воно відразу ж з'являється і на ноутбучі, і в інтернеті [19].

### **2.3 Вибір фреймворку**

Фреймворки вже містять підігнані один до одного сумісні бібліотеки, так що розробнику не потрібно нічого оновлювати, слідкувати за обмеженням і турбуватися про сумісність.

З точки зору бізнесу розробка на фреймворку майже завжди економічно ефективніше і якісніше за результатом, ніж написання

проекту на чистій мові програмування без використання будь-яких зайвих платформ. Розробка без використання платформи може бути правильним рішенням тільки в двох випадках - або проект зовсім простий і не передбачає подальшого розвитку, або проект дуже складний і потребує дуже низькорівневої оптимізації.

Якщо порівнювати фреймворки з іншими класами платформ - SaaS, CMS - то фреймворки значно ефективніше використовувати в проектах зі складною бізнес-логікою і з високими вимогами до швидкості роботи, надійності і безпеки. Але в простих і типових проектах без значущих вимог швидкість і вартість розробки на фреймворку буде вище, ніж на SaaS або CMS.

З усього перерахованого вище я вирішив розробляти веб-сайт за допомогою фреймворку, отже для створення я вирішив взяти React, оскільки він водночас і достатньо легкий для вивчення і поступливий на відміну від того самого Angular. А ще у React більше джерел інформації ніж у Vue.js, саме через це писати клієнтську частину на React буде набагато зручніше.

Для розробки серверної частини я вирішив обрати фреймворк Express. Зараз він самий популярний для створення серверу на Node.js [20].

## **2.4 Вибір середовища розробки**

Для розробки проекту я обрав IDE Visual Studio Code від компанії Microsoft. Ліцензія є безкоштовною. Visual Studio Code є дуже зручним середовищем для розробки веб-сайтів.

У свідомості середньостатистичного користувача ім'я компанії Microsoft незмінно асоціюється з пакетом додатків Microsoft Office. Але при цьому досить багато інших випущених програмних продуктів залишаються незвіданими звичайному користувачеві, або їм не

достається належної уваги. Іноді складається враження, що і веб-розробники нічого не знають або не хочуть знати окрім лише Visual Studio.

Цей продукт гарно зроблений і він став би ще краще, якби розробники перенесли його на інші платформи. Переносити цю IDE на інші ОС в Microsoft і не збираються, але розробниками компанії був створений Visual Studio Code - дуже непоганий редактор коду, використовувати який можна не тільки на Windows, але і на Linux, і OS X.

Visual Studio Code представляє із себе «легкий» редактор коду для кроссплатформної розробки веб-додатків.

Visual Studio Code розроблений на Electron - фреймворк, який дозволяє з використанням Node.js розробляти настільні додатки, які працюють на движку Blink. Незважаючи на те, що редактор заснований на Electron, він не використовує редактор Atom. Замість нього реалізується веб-редактор Monaco, розроблений для Visual Studio Online.

Серед підтримуваних мов і технологій: HTML, CSS, JavaScript, PHP, C++, C#, TypeScript, jade, Python, XML, Batch, F#, DockerFile, Coffee Script, Java, HandleBars, R, Objective-C, PowerShell, Luna, Visual Basic, Markdown, JSON, LESS і SASS, Haxe [21].

## **2.5 Вибір бази даних**

Для створення сайту школи я обрав нереляційну СУБД, а саме MongoDB. MongoDB реалізує новий підхід до побудови баз даних, де немає таблиць, схем, запитів SQL, зовнішніх ключів і багатьох інших речей, які притаманні об'єктно-реляційним базам даних. На відміну від реляційних баз даних MongoDB пропонує документо-орієнтовану модель даних, завдяки чому MongoDB працює швидше, має кращу масштабованість, її легше використовувати. [22]



Вона має такі переваги:

- Зручність в управлінні – СУБД не потребує окремого адміністратора бази даних. Завдяки достатній зручності у використанні, їй легко можуть користуватися як розробники, так і системні адміністратори;
- Швидкість – Висока продуктивність при виконанні простих запитів;
- Гнучкість – В MongoDB можна без шкоди для існуючих даних, їх структури і продуктивності СУБД додавати поля або колонки.
- Масштабованість – MongoDB горизонтально масштабуєма, що дозволяє легко зменшити навантаження на сервера при великих обсягах даних;
- Кросплатформеність – MongoDB написана на C ++, тому її легко перенести на найрізноманітніші платформи. MongoDB може бути розгорнута на платформах Windows, Linux, MacOS, Solaris. Можна також завантажити вихідний код і самому скомпілювати MongoDB, але рекомендується використовувати бібліотеки з офсайта;
- Простота у використанні – відсутність складної схеми бази даних і в зв'язку з цим потреби при щонайменшій зміні концепції зберігання даних пересоздавати цю схему значно полегшують роботу з базами даних MongoDB і подальшим їх масштабуванням. Крім того, економиться час розробників. Їм більше не треба думати про повторне створення бази даних і витратити свій час на побудову складних запитів.

## **2.6 Вибір середовища розміщення**

Кожен сайт в Інтернеті складається з великої кількості файлів, скриптів, баз даних та інших програмних компонентів. Всі вони повинні десь зберігатися і бути цілодобово доступними. Місце, де фізично розташовується сайт, і має назву хостингом. Необхідний сайту дисковий

простір виділяється на серверах - потужних комп'ютерах зі спеціально налаштованим ПО. Вони знаходяться під постійним технічним контролем і функціонують в безперервному режимі, забезпечуючи цілодобовий доступ до web-ресурсу. Для постійної роботи сайту, потрібно розмістити його на одному із веб хостингів. Для даного проекту я вирішив розмістити сайт на сервері ресурсу [www.heroku.com](http://www.heroku.com). [23]

## **2.7 Вибір системи контролю версій**

Кожен проект складається з незліченних числа маленьких змін. У кінцевому разі, вони формують сайт, додаток, або будь-якої іншої продукт. Ваша система контролю версій стежить за цими змінами.

Система контролю версій - це система, яка вносить зміни у файли або набір декількох файлів протягом якогось часу, таким чином, що ви зможете повернутися до певної версії що була раніше. Для керування версіями розробленого проекту я вирішив використовувати GitHub.

GitHub один з найбільших веб-сервісів для спільного створення програмного забезпечення. Існують безкоштовні та платні тарифні плани для використання сайту. GitHub ґрунтується на системі керування версіями Git.[24]

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ

### 3.1 Загальна характеристика сайту

Вході виконання дипломного проекту були використані такі критерії що до створення:

- Адаптивність сайту;
- Швидка авторизація;
- Надання права користувачам на сайті;
- Кроссбраузерність сайту.

Користувачі мають змогу пройти реєстрацію без довгого очікування підтвердження реєстрації адміністратором чи модератором.

Також сайт набув адаптивність та кроссбраузерність, це було зроблено для того щоб користувачі могли легко користуватися ним з будь яких пристроїв, та з будь якого браузера.

Адміністратор та модератор повинні мають змогу створювати статті, редагувати та видаляти їх, завантажувати фотографії до галереї сайту. Незареєстровані та зареєстровані користувачі мають змогу лише читати статті та переглядати фотогалерею.

### 3.2 Опис функцій

При розробці проекту було використані деякі функції. Нижче буде перелічено деякі із ни.

Авторизація через passport.js. Це middleware (сполучне програмне забезпечення) для авторизації під node.js. Passport.js підтримує авторизацію за допомогою величезної кількості сервісів, включаючи «ВКонтакте» та інші сервіси. Passport.js дозволяє легко налаштувати будь-який веб-додаток на базі Express, а так як ми його використовуємо, то це гарний вибір для нашого проекту.

Підключення авторизації виглядає так:

```
var passport = require('passport');
```

```

var LocalStrategy = require('passport-local').Strategy;

passport.use(new LocalStrategy({
  usernameField: 'email',
  passwordField: 'password'
}, function(username, password, done) {
  User.findOne({ username : username }, function(err, user) {
    return err
      ? done(err)
      : user
        ? password === user.password
          ? done(null, user)
          : done(null, false, { message: 'Incorrect password.'
        : done(null, false, { message: 'Incorrect username.'
  });
  });
}));

```

Однією з найбільш потрібних завдань в Node.js є завантаження файлів на сервер. На жаль в Node.js для цього немає вбудованих інструментів, проте ми можемо використовувати додаткові спеціальні пакети. Одним з популярних подібних пакетів є `multer`, який працює зверху Express.

Спочатку його треба додати в файл, в якому буде розташовуватися форма для завантаження файлу. А потім в головному файлі `app.js` використовуємо `multer` для отримання відправленого файлу. І код буде мати такий вигляд:

```

const express = require("express");
const multer = require("multer");

const app = express();

app.use(express.static(__dirname));
app.use(multer({dest:"uploads"}).single("filedata"));
app.post("/upload", function (req, res, next) {

  let filedata = req.file;

```

```
console.log(filedata);  
  
if(!filedata)  
    res.send("Помилка при завантажені");  
  
else  
    res.send("Файл завантажений");  
  
});  
  
app.listen(3000);
```

### 3.3 Структура проекту

Структура проекту ввідображено на рисунку 3.1.

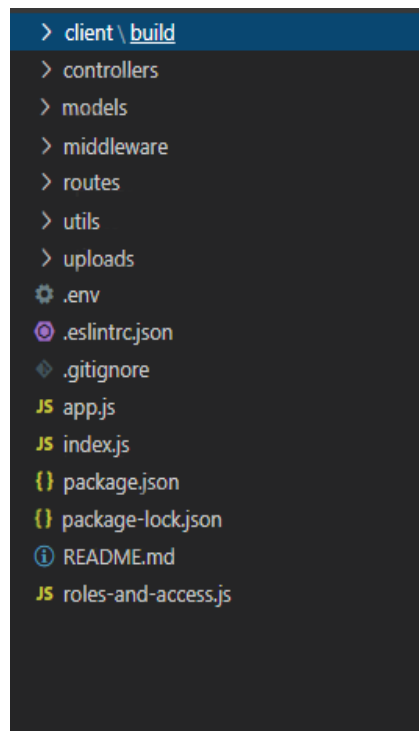


Рисунок 3.1 – Структура проекту

Опис файлів та каталогів:

- client – папка в якій зберігається клієнтська частина веб сайту;
- controllers – папка в якій зберігаються контроллери;
- models – папка з моделями БД;
- middleware – папка для middlewares;
- routes – папка всіх роутів створеної API;

- utils – папка з утилітами;
- uploads – папка с завантаженими фото;
- .env – конфігурація;
- .eslintrc.json – правила для js лінтера;
- .gitignore – файл в якому описуються каталоги та файли які не будуть завантажуватись до репозиторію;
- app.js – конфігурація сервера, підключення бібліотек та ін.;
- index.js – стартова точка програми;
- roles-and-access.js – файл для перевірки ролей та доступу до ресурсів;
- package.json – опис проекту, версія, список залежностей для роботи.

### **3.2 Результат виконаної роботи**

У результаті виконання дипломного проекту було реалізовано заплановану мету проекту, тобто – основний функціонал: головну сторінку сайту, механізм авторизації та реєстрації користувачів, створення та редагування статей, та користувачами, і також фотогалерею школи.

При першому вході на сайт ми з’являємося на вкладці «Статті» де відображенні останні новини із життя школи і освіти відображена на рисунку 3.2.

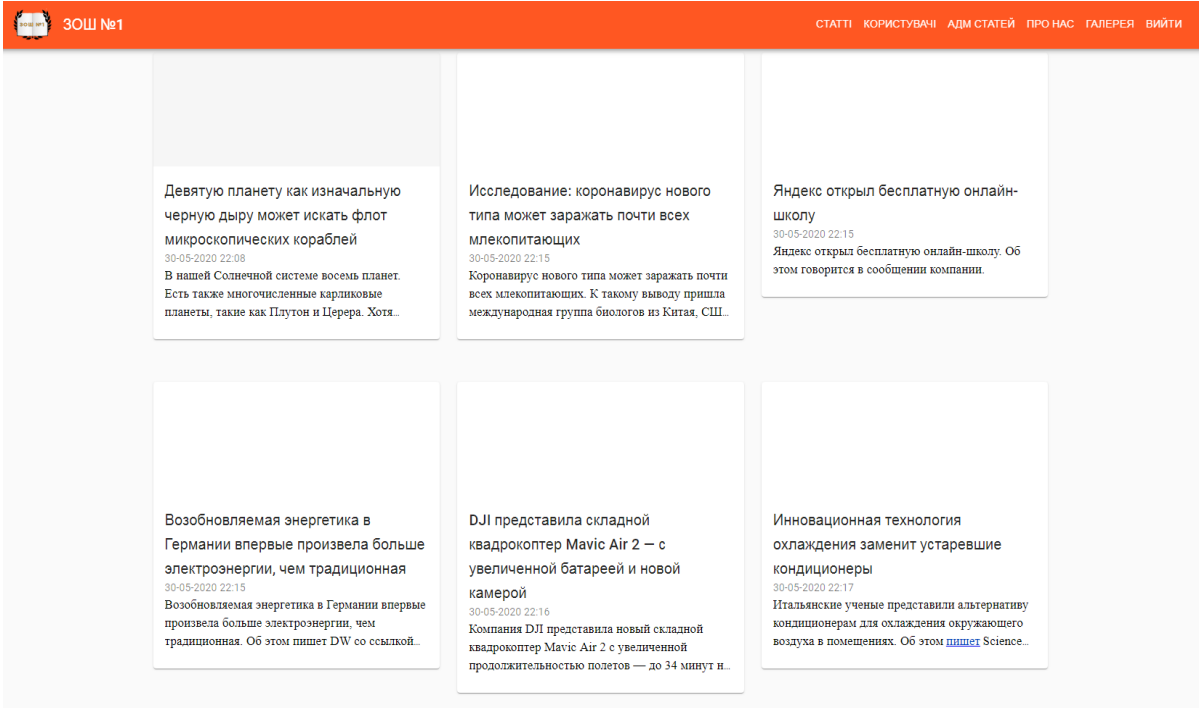


Рисунок 3.2 – Головна сторінки з ПК

Для входу на сайт зроблена зручна і швидка рестрація яка не заставляє чекати підтвердження реєстрації і не втрачати зайвий час зображена на рисунку 3.3.

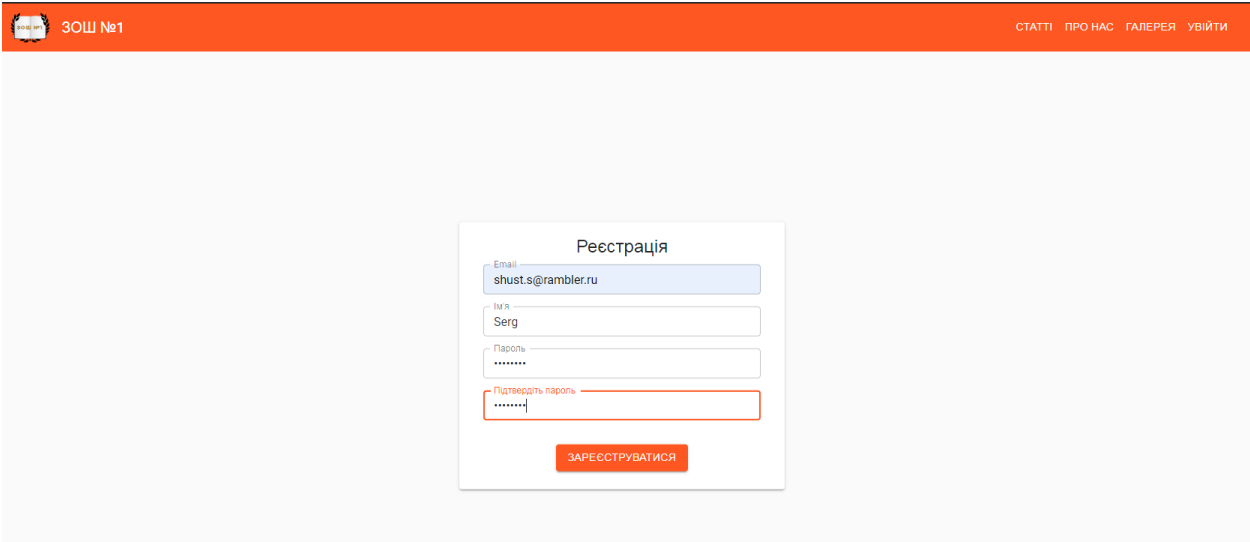


Рисунок 3.3 – Сторінка реєстрації користувача

Вікно авторизації легке і зрозуміле, зображена на рисунку 3.4.

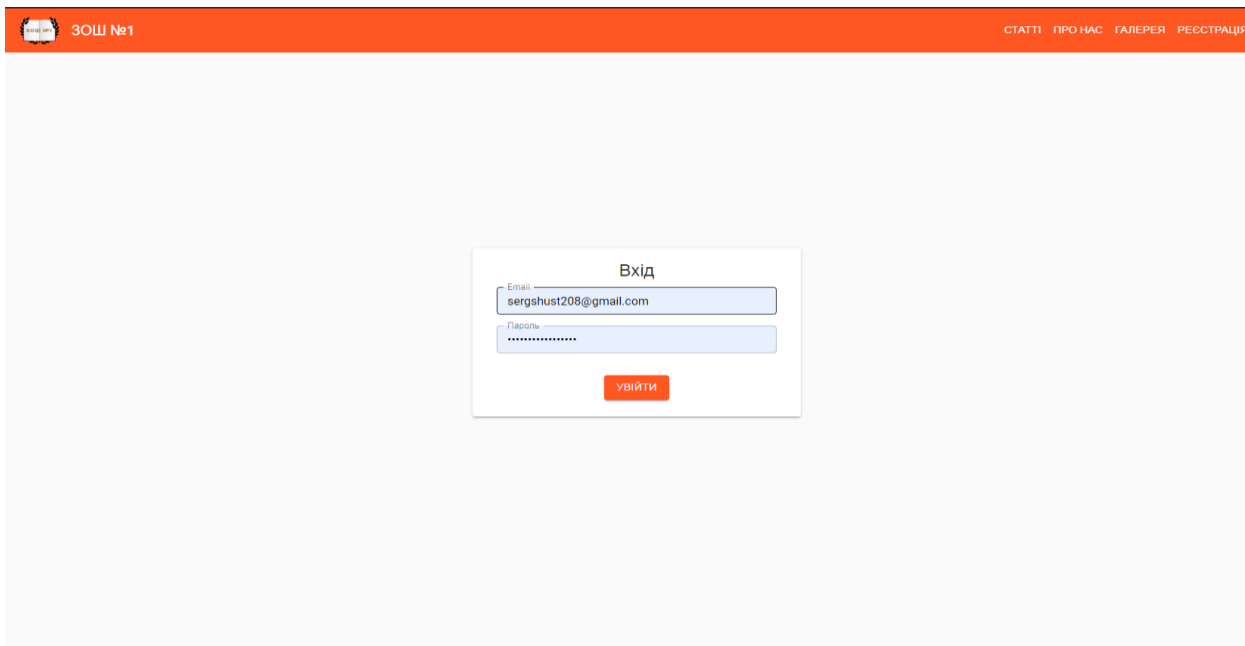


Рисунок 3.4 – Сторінка авторизації користувача

Admin може призначати користувачам роль на сайті, тобто назначати інших адміністраторів, назначати модераторів (Moderator). Звичайний користувач не має ніяких прав і має роль User с початку користуванням сайту. Керування користувачами зображена на рисунку 3.5.

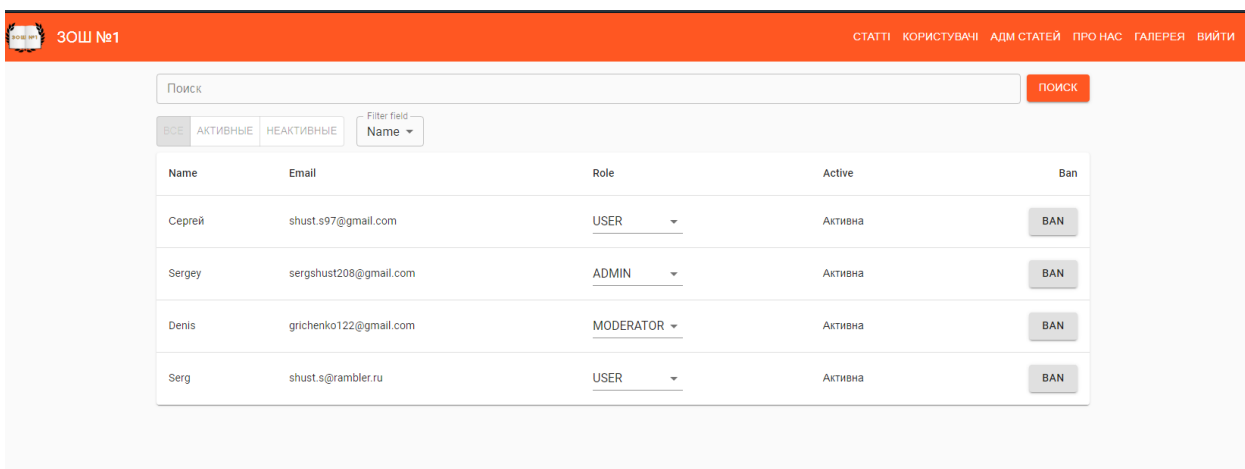


Рисунок 3.5 – Керування користувачами



Через вкладку «Адм Статей» на сайті, адміністратор або модератор може створювати чи редагувати статті на сайті школи. В текст статті можна вставляти зображення, посилання на інші сайти та смайлики. Створення статей зображена на рисунку 3.6, видалення і редагування статей на рисунку 3.7.

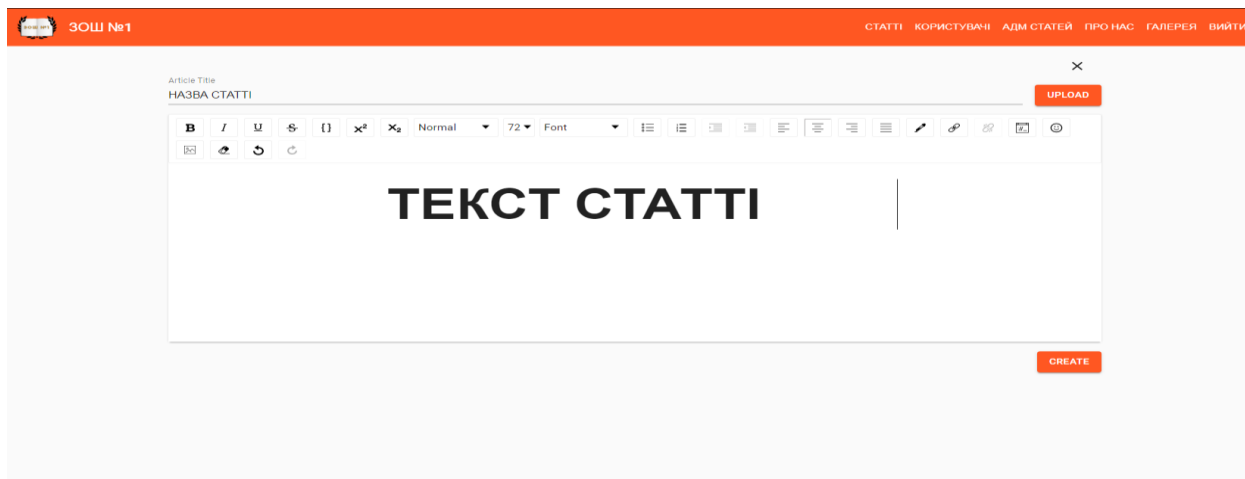


Рисунок 3.6 – Створення статей

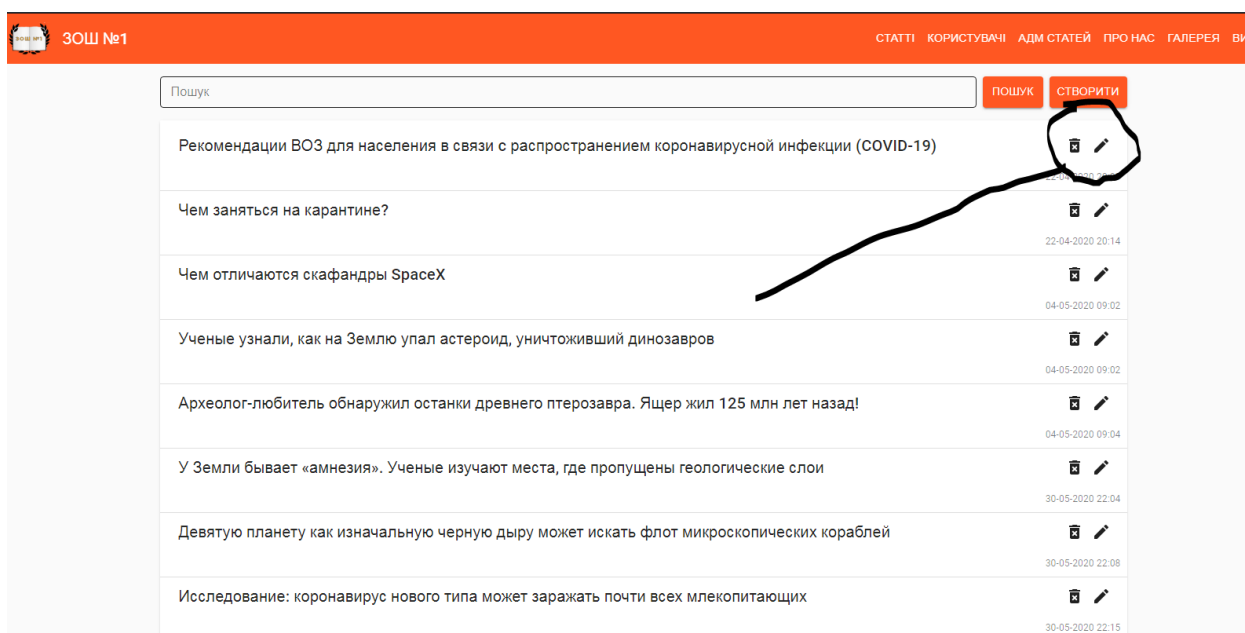


Рисунок 3.7 – Редагування і видалення статей

Також на сайті є своя галерея фотографій, які показують визначні події із життя школи. Галерея зображена на рисунку 3.8.



Рисунок 3.8 – Галерея школи

У кожного навчального закладу є своя історія. На сайті є можливість познайомитися з нею. Також є опис матеріально-технічної бази, де можна дізнатися про стан закладу. Меню «Про нас» зображено на рисунку 3.9.

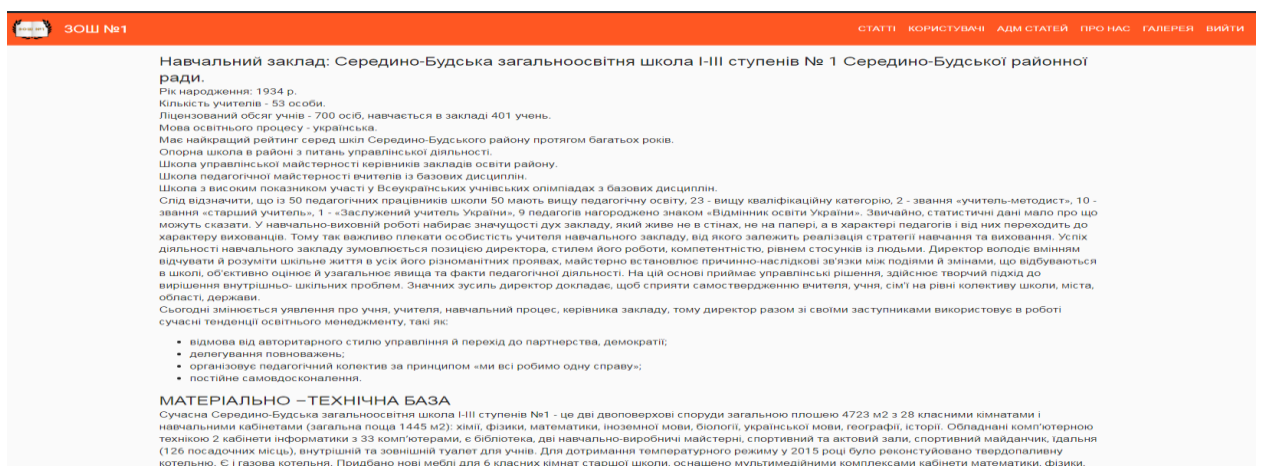
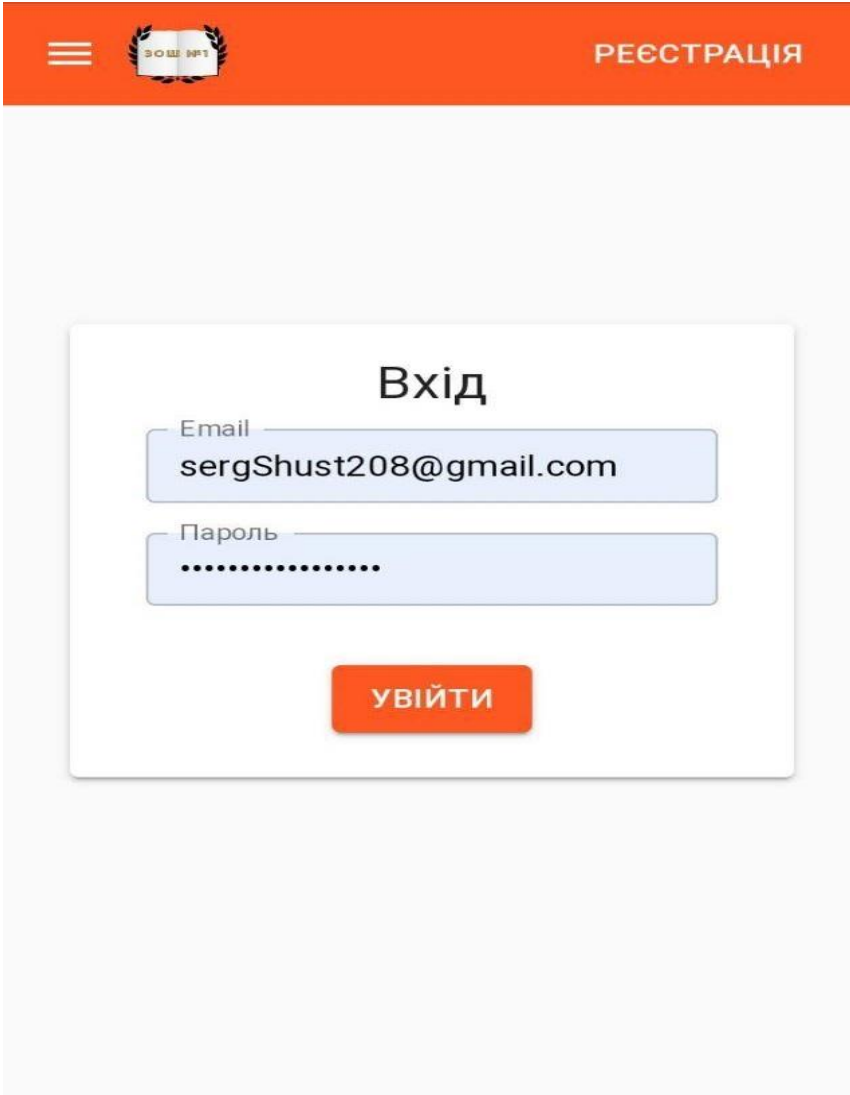


Рисунок 3.9 – Історія школи і матеріально-технічна база

Так як більшість користувачів зараз використовують смартфони для перебування в інтернеті, було розроблено адаптивність сайту під екрани мобільних пристроїв а зручність сайту не постраждала и не набула кардинальних змін.

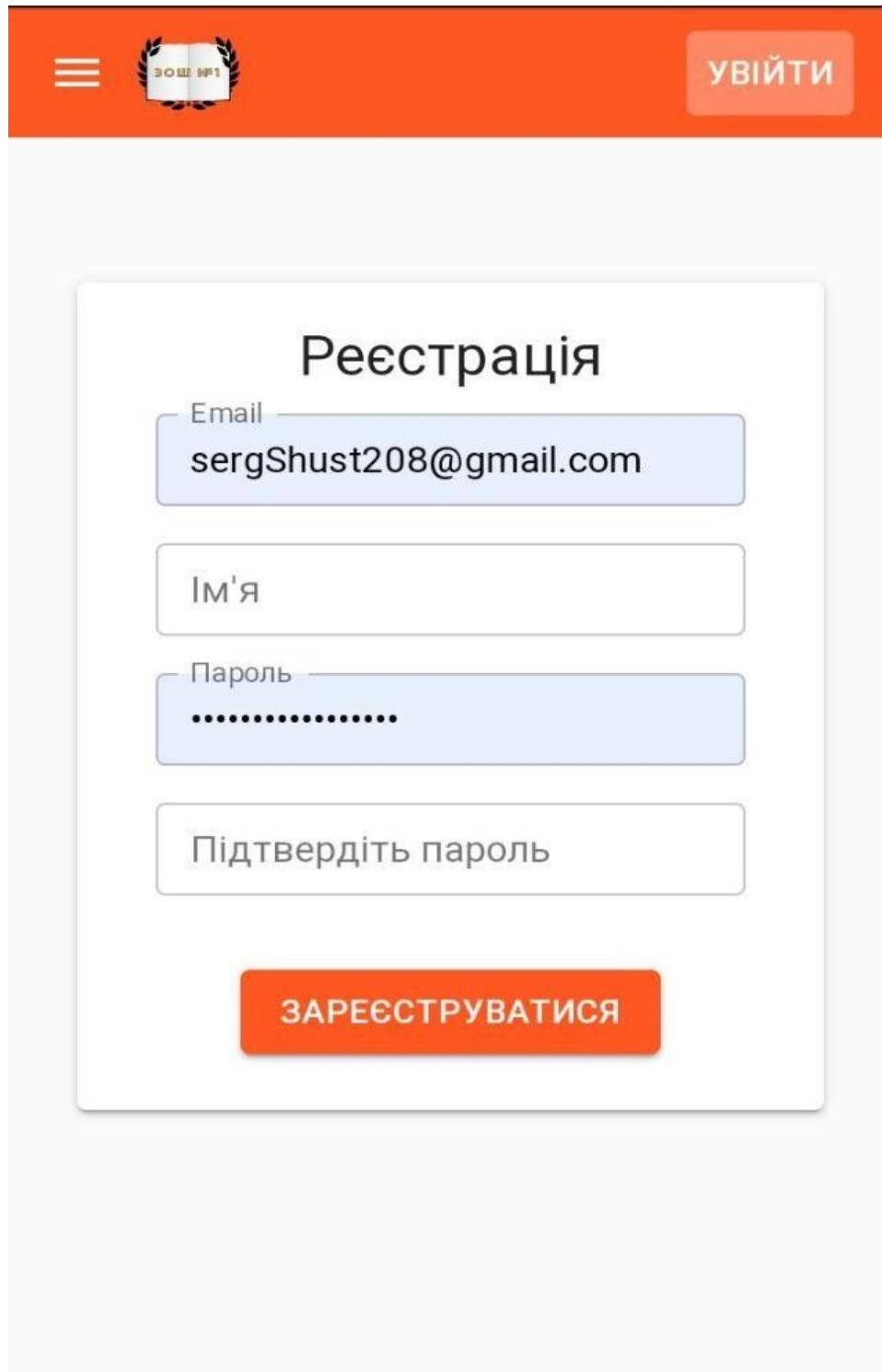
Адаптивне вікно входу зображено на рисунку 3.10.



The image shows a mobile-optimized login interface. At the top, there is an orange navigation bar containing a hamburger menu icon, a logo with the text 'З О Ш І М П Т', and the word 'РЕЄСТРАЦІЯ'. The main content area is white and features a central white box with a light gray border. Inside this box, the title 'Вхід' is displayed in a large, bold, black font. Below the title are two input fields: the first is labeled 'Email' and contains the text 'sergShust208@gmail.com'; the second is labeled 'Пароль' and contains a series of dots representing a masked password. Below these fields is a prominent orange button with the white text 'УВІЙТИ'.

Рисунок 3.10 – Вікно входу на сайт зі смартфона

Адаптивне вікно реєстрації відображено на рисунку 3.11.



The image shows a mobile application interface for registration. At the top, there is an orange header bar containing a menu icon (three horizontal lines), a logo with the text 'ЗОШ №1' (School No. 1), and a button labeled 'УВІЙТИ' (Log In). The main content area is white and features a registration form titled 'Реєстрація' (Registration). The form includes four input fields: 'Email' with the value 'sergShust208@gmail.com', 'Ім'я' (Name), 'Пароль' (Password) with masked characters, and 'Підтвердіть пароль' (Confirm password). Below the form is a large orange button labeled 'ЗАРЕЄСТРУВАТИСЯ' (Register).

Рисунок 3.11 – Вікно реєстрації на сайт зі смартфона

Зручне адаптивне меню сайту на рисунку 3.12.

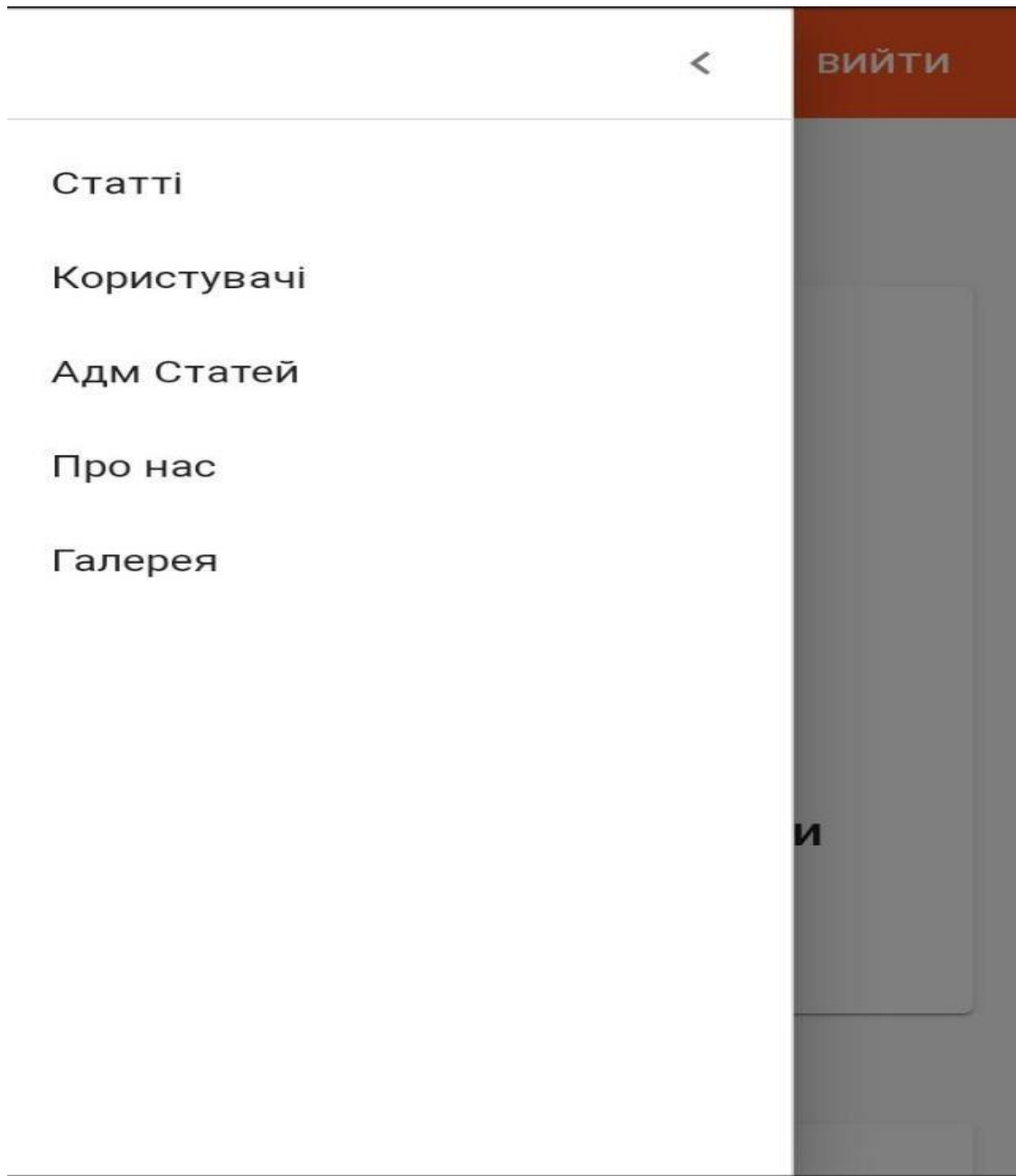


Рисунок 3.12 – Вікно меню на сайті зі смартфона

## ВИСНОВКИ

В ході виконання дипломної роботи мною був розроблений функціональний інформаційний освітній веб-сайт, який повністю готовий до роботи. Даний сайт орієнтований на учнів їхніх батьків і викладачів. З його допомогою користувачі зможуть отримувати необхідну інформацію і задавати свої питання в форумі. При розміщенні веб-сайту в глобальній мережі, імідж та популярність навчального закладу значно збільшиться. Веб сайт має простий у використанні і інтуїтивно зрозумілий інтерфейс.

На веб сайті реалізовано:

- статті –де розміщенні останні новини школи і освіти;
- керування користувачами – надання прав адміністратора, модератора та блокування інших користувачів;
- створення статей – створення інформаційних статей для розповсюдження інформації;
- інформація про заклад – відомості про історію та стан закладу;
- галерея – фотографії з визначних подій із життя закладу.

У підсумку, в ході дипломного проекту, мною був створений простий сайт, який в подальшому можна і потрібно буде розширити, додати новий функціонал.

## СПИСОК ДЖЕРЕЛ

1. Сайт ЗОШ №1 м. Середина-Буда [Електронний ресурс]. – Режим доступу: <http://sbuda-sh1.at.ua>
2. Комунальна установа Сумська спеціалізована школа I-III ступенів № 1 ім. В. Стрельченка [Електронний ресурс]. – Режим доступу: <https://www.sites.google.com/site/sumskaskola1/>
3. Одеська загальноосвітня школа №16 I-III ступеней одеської міської ради одеської області [Електронний ресурс]. – Режим доступу: <http://www.prvo.od.ua/school/s16/?p=&as=&as2=&as3=&as4=&as5=>
4. Методи написання web-сайтів [Електронний ресурс]. - Режим доступу <http://webstudio2u.net/ru/webdesign/354-site-development-methods.html>
5. Microsoft FrontPage [Електронний ресурс]. – Режим доступу [https://ru.wikipedia.org/wiki/Microsoft\\_FrontPage](https://ru.wikipedia.org/wiki/Microsoft_FrontPage)
6. Adobe Dreamweaver [Електронний ресурс]. – Режим доступу [https://ru.wikipedia.org/wiki/Adobe\\_Dreamweaver](https://ru.wikipedia.org/wiki/Adobe_Dreamweaver)
7. Н. Савельева – Системы управления контентом [Електронний ресурс]. – Режим доступу <http://masters.donntu.org/2007/fvti/kudin/library/article09.htm>
8. Документация Drupal [Електронний ресурс]. – Режим доступу: <https://drupal.ru/docs>
9. Документация Joomla [Електронний ресурс]. – Режим доступу: <https://joomla.ru/docs>
10. Стаття на Wikipedia про WordPress [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/WordPress>

11. Статья на Habr [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/31107/>
12. Прогрессивный JavaScript-фреймворк Vue.js [Электронный ресурс]. – Режим доступа: <https://ru.vuejs.org>
13. React.js: понятное руководство для начинающих [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/ruvds/blog/428077/>
14. Введение в Angular [Электронный ресурс]. – Режим доступа: <https://metanit.com/web/angular2/1.1.php>
15. SaaS-платформы для создания сайтов [Электронный ресурс]. – Режим доступа: [https://web-creator.ru/articles/about\\_saas](https://web-creator.ru/articles/about_saas)
16. Статья на Wikipedia про HTML5 [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/HTML5>
17. Статьи о CSS3 [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/css3>
18. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru>
19. Янг А., Мек Б., Кантелон М. - Node.js в действии [Электронный ресурс]. – Режим доступа: <https://ru.pdfdrive.com/nodejs-в-действии-e148600761.html>
20. Express. Быстрый, гибкий, минималистичный веб-фреймворк для приложений Node.js [Электронный ресурс]. – Режим доступа: <https://expressjs.com/ru/>
21. Visual Studio Code [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://ru.wikipedia.org/wiki/Visual_Studio_Code)
22. SQL против NoSQL на примере MySQL и MongoDB [Электронный ресурс]. – Режим доступа: <https://tproger.ru/translations/sql-vs-nosql/>
23. Обзор бесплатных хостингов [Электронный ресурс]. – Режим доступа: <https://ru.hostings.info/hostings/rating/free-hosting>



24. Стаття на Wikipedia про GitHub [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/GitHub>

25. ГОСТ 2.102 – 95 ЕСКД Загальні вимоги до текстової документації.

26. ГОСТ 2.105 – 68 ЕСКД Види та комплектація конструкторських документів.

27. ДСТУ ГОСТ 7.1:2006 "Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання"

## Додаток А

```
import React, { useEffect } from "react";
import "typeface-roboto";
import "slick-carousel/slick/slick.css";
import "slick-carousel/slick/slick-theme.css";
import { connect } from "react-redux";
import {
    Switch,
    Route,
} from "react-router-dom";
import Header from "../components/header";
import AuthPage from "../pages/auth";
import SideMenu from "../components/side-menu";
import HomePage from "../pages/home";
import NewsPage from "../pages/articles";
import authActions from "../actions/auth.actions";
import Alert from "../components/alert";
import alertActions from "../actions/alert.actions";
import ArticlePage from "../pages/article";
import UsersPage from "../pages/users";
import ArticlesListPage from "../pages/articles-list";
import AboutUsPage from "../pages/about-us";
import GalleryPage from "../pages/gallery";

const App = ({
    checkAuthorization,
    alertOpen,
```

```
    alertTitle,  
    alertContent,  
    alertCb,  
    closeAlert,  
    confirm,  
  }) => {  
    useEffect(() => {  
      checkAuthorization();  
    }, [checkAuthorization]);  
  
    return (  
      <>  
        <Header />  
        <SideMenu />  
        <Switch>  
          <Route exact path="/" />  
component={HomePage} />  
          <Route path="/article/:id" />  
component={ArticlePage} />  
          <Route path="/login" />  
component={AuthPage} />  
          <Route path="/registration" />  
component={AuthPage} />  
          <Route path="/posts" />  
component={NewsPage} />  
          <Route path="/users" />  
component={UsersPage} />  
          <Route path="/articles-list" />  
component={ArticlesListPage} />  
          <Route path="/about-us" />  
component={AboutUsPage} />  
        </Switch>  
      </>  
    );  
  }  
);
```

```

        <Route path="/gallery"
component={GalleryPage} />
        <Route render={() => <h1>404 Not
Found</h1>} />
    </Switch>
    <Alert
        open={alertOpen}
        handleClose={closeAlert}
        title={alertTitle}
        content={alertContent}
        cb={alertCb}
        confirm={confirm}
    />
</>
);
};
const mapStateToProps = ({
    alertReducer: {
        alertOpen, alertTitle, alertContent, alertCb,
confirm,
    },
}) => ({
    alertOpen,
    alertTitle,
    alertContent,
    alertCb,
    confirm,
});

```

```
const mapDispatchToProps = {
  checkAuthorization: () =>
authActions.checkAuthorization(),
  closeAlert: () => alertActions.closeAlert(),
};

export default connect(mapStateToProps,
mapDispatchToProps)(App);
```

## Додаток В

```

interface ValidationError {[key: string]: string }
export const emailErrors: ValidationError = {
  required: "Email is required",
  pattern: "Invalid email",
};
export const nameErrors: ValidationError = {
  required: "Name is required",
  minLength: "Too short name",
  maxLength: "Too long name",
};
export const passwordErrors: ValidationError = {
  required: "Password is required",
  minLength: "Minimum password length 7 characters",
  maxLength: "Maximum password length 7 characters",
};
export const passwordConfirmErrors: ValidationError = {
  validate: "Passwords does not match",
};
export const regExp = {
  email:
/^((([^<>() []\.\.,;:\s@"]+(\. [^<>() []\.\.\.,;:\s@"]+)*)|(".+"))@
((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\])|([a-zA-
Z\-0-9]+\.)+[a-zA-Z]{2,}))$/ ,
  name: /^(\d*([a-zA-Za-яA-Я]{1,})\d*)*$/ ,
  password: /^[A-Za-z0-9]+$/ ,
};

```