

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РОЗРОБКА ШКІЛЬНОГО САЙТУ ІЗ  
ЗАСТОСУВАННЯМ ТЕХНОЛОГІЇ REACT»

Виконав: студент \_\_\_\_\_ 4 \_\_\_\_\_ курсу, групи \_\_\_\_\_ 6.1220  
спеціальності \_\_\_\_\_ 122 Комп'ютерні науки  
(шифр і назва спеціальності)  
освітньої програми \_\_\_\_\_ Комп'ютерні науки  
(назва освітньої програми)  
\_\_\_\_\_ Храпко В.В.  
(ініціали та прізвище)

Керівник \_\_\_\_\_ доцент кафедри комп'ютерних наук,  
доцент, д.е.н. Полуктова Н.Р.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент \_\_\_\_\_ заступник директора з навчально-виробничої роботи  
Томаківського професійного аграрного ліцею Русанова С.Г.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2024

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

Освітня програма Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Завідувач кафедри комп'ютерних наук,  
д.т.н., доцент

\_\_\_\_\_ Шило Г.М.

(підпис)

“ 25 ” грудня 2023 р.

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Храпку Вадиму Віталійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка шкільного сайту із застосуванням технології REACT

керівник роботи Полуектова Наталія Робертівна, д.е.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с

2. Строк подання студентом роботи 05.06.2024

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

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

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Розробка сайту для ліцею за технологією REACT.

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

презентація

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 25.12.2023

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	08.01.2024	
2.	Збір вихідних даних.	17.01.2024	
3.	Обробка методичних та теоретичних джерел.	22.01.2024	
4.	Розробка першого та другого розділу.	04.03.2024	
5.	Розробка третього розділу.	15.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	08.06.2024	
7.	Захист кваліфікаційної роботи.	22.06.2024	

Студент \_\_\_\_\_  
(підпис)

В.В. Храпко  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

Н.Р. Полуктова  
(ініціали та прізвище)

## Нормоконтроль пройдено

Нормоконтролер \_\_\_\_\_  
(підпис)

О.Г. Спиця  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка шкільного сайту із застосуванням технології REACT»: 36 с., 5 рис., 11 джерел, 2 додаток.

API, JAVASCRIPT, MYSQL, PHP, REACT, ІНФОРМОВАНІСТЬ, ОСВІТА, ОСВІТНЯ СИСТЕМА

Об'єкт дослідження – сайт освітнього закладу.

Мета роботи: розробка інформаційно-навчального сайту для потреб учнів, батьків, викладачів з можливістю доступу до індивідуальних матеріалів.

Методи дослідження – опитування, аналіз, синтез, аналогія, порівняння.

Був розроблений новий сайт на основі сучасних технологій для КЗО «Томаківський професійний аграрний ліцей» ДОР» який надає персоналізовану інформацію про новини закладу та освітній контент.

## SUMMARY

Bachelor's Qualification Theses " Development of a school website using REACT technology": 36 p., 5 figures, 11 sources, 2 supplements.

API, JAVASCRIPT, MYSQL, PHP, REACT, INFORMATION, INVENTORY, INVENTORY SYSTEM

Object of the study is the website of an educational institution.

Aim of the study: to develop an information and educational website for the needs of students, parents, and teachers with access to individual materials.

Research methods: survey, analysis, synthesis, analogy, comparison.

A new website based on modern technologies was developed for the Tomakivsky Professional Agricultural Lyceum, which provides personalized information about the institution's news and educational content.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат .....	4
Summary .....	5
Вступ.....	7
1 Аналіз предметної області .....	8
1.1 Сайти освітніх установ. Призначення, проблеми та рішення .....	8
1.2 Огляд сучасних технологій розробки сайтів.....	10
1.3 Особливості розробки сайтів освітніх установ .....	12
2 Розробка сайту ліцею .....	16
2.1 Постановка завдання .....	16
2.1.1 Опис вимог до сайту.....	17
2.1.2 Структура сайту.....	18
2.1.3 Вимоги до дизайну сайту .....	19
2.2. Обґрунтування вибору технологій розробки сайту .....	20
2.2.1 Обґрунтування вибору СУБД.....	24
3 Результати розробки сайту .....	27
Висновки.....	29
Перелік посилань .....	30
Додаток А Код основних сторінок.....	32
Додаток Б Формування картків для курсів .....	35

## ВСТУП

Сучасні технології надають велику можливість для удосконалення освітнього процесу, навчання для учнів та викладачів.

У сучасній Україні швидкими темпами йде прогрес сайтів для дистанційного навчання з тих причин що наразі в країні війна а також була пандемія коронавірусу.

Для реалізації навчального процесу у дистанційній формі освітнім організаціям необхідно було вирішити кілька основних завдань:

- визначити для учнів та їхніх батьків основне джерело інформації про організацію та реалізації дистанційного навчального процесу;
- визначити засоби та інструменти організації навчальної діяльності;
- визначити форми роботи учнів зі змістом навчального матеріалу, а також завданнями для самостійної роботи та контрольнo-вимірювальними матеріалами;
- визначити способи та форми організації зворотного зв'язку учнів та вчителів для перевірки результатів навчальної діяльності.

Таким чином, метою даної кваліфікаційної роботи було створення нового сайту для «КЗО «Томаківський професійний аграрний ліцей» ДОР», який дозволяє розв'язувати наступні задачі:

- реєстрація та авторизований доступ всіх членів освітнього процесу;
- перегляд новин;
- індивідуалізований доступ до навчальних курсів;
- зворотній зв'язок учнів та батьків з адміністрацією закладу;
- можливості розширення функціоналу в майбутньому.

При реалізації сайту використовується сучасний стек інструментів розробки, включаючи бібліотеку React, яка дозволяє більш ефективно виконувати браузерну обробку сторінок.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Сайти освітніх установ. Призначення, проблеми та рішення

Призначення сайтів освітніх установ полягає в інформаційній підтримці викладачів, абітурієнтів, учнів та батьків. Вона включає в себе доступ до необхідної інформації що включає в себе правила прийому, розклад занять, новини, навчальні програми, інформацію про освітній заклад, виховну роботу з учнями.

Сайти освітніх установ відіграють важливу роль у сучасній освіті, надаючи доступ до інформації та ресурсів як для студентів та викладачів, так і для батьків та зовнішніх організацій. Нижче розглянемо основні функції таких сайтів, типові проблеми та можливі рішення.

Призначення сайтів освітніх установ [1]

Інформаційна підтримка:

- надання актуальної інформації про діяльність установи, навчальні плани, розклад занять та заходів;
- публікація новин та оголошень щодо освітнього процесу.

Комунікація та взаємодія:

- забезпечення каналу зв'язку між студентами, викладачами, адміністрацією та батьками;
- можливість онлайн-консультацій та спілкування з представниками освітньої установи.

Електронні освітні ресурси:

- доступ до навчальних матеріалів, методичних посібників, бібліотек та баз даних;
- платформи для дистанційного навчання та вебінари.

Адміністративні функції:

- автоматизація процесів подання заяв, реєстрації на курси, оплати та інші адміністративні завдання;



- надання інформації про вакансії та кар'єрні можливості.

Підтримка іміджу та репутації:

- формування позитивного іміджу установи, демонстрація досягнень та успіхів;
- залучення нових студентів та партнерів.

Проблеми сайтів освітніх установ

- нестача актуальної інформації:

Часто інформація на сайтах старіє, що ускладнює користувачам отримання нових даних.

- проблеми з навігацією та інтерфейсом:

Складність пошуку необхідної інформації через незручну структуру сайту та відсутність логічної навігації.

- низька продуктивність та доступність:

Повільне завантаження сторінок та відсутність оптимізації для мобільних пристроїв, що обмежує доступність сайту.

- відсутність інтеграції із зовнішніми системами:

Неможливість інтеграції з іншими освітніми платформами та сервісами, що ускладнює обмін даними.

- низький рівень безпеки:

Уразливості сайту, які можуть призвести до витоку даних та інших проблем безпеки.

- недостатня підтримка зворотного зв'язку:

Відсутність ефективних інструментів для зворотного зв'язку з користувачами та вирішення їхніх питань.

Обмежена функціональність для користувачів з особливими потребами:

Недостатня увага доступності для людей з обмеженими можливостями.

Рішення для покращення сайтів освітніх установ

- регулярне оновлення контенту.

Призначення відповідальних осіб за актуалізацію інформації та своєчасне оновлення новин та оголошень.

- поліпшення навігації та інтерфейсу користувача:

Розробка логічної структури сайту, створення зручних меню та використання сучасних стандартів дизайну.

– оптимізація продуктивності:

Використання сучасних технологій для прискорення завантаження сторінок та сумісності з мобільними пристроями.

– інтеграція із зовнішніми сервісами:

Впровадження API та інших методів інтеграції з освітніми платформами та системами керування навчанням.

– підвищення рівня безпеки:

Регулярний аудит безпеки, впровадження SSL сертифікатів та інших заходів для захисту даних користувачів.

– підтримка ефективного зворотного зв'язку:

Використання чатів, форумів та інших інструментів для швидкого реагування на запити користувачів.

– поліпшення доступності:

Сайт допомагає комунікації між викладачами, учнями та батьками для швидкого обміну повідомленнями. Комунікація допомагає в виборі форми навчання: як дистанційно або змішано навчатися при перебуванні учнів за кордоном або перебуванні в іншій області країни в умовах військового стану.

## **1.2 Огляд сучасних технологій розробки сайтів**

Розробка сайтів сьогодні включає безліч технологій, які забезпечують функціональність, продуктивність та безпеку веб-додатків. Нижче розглянемо ключові сучасні технології, що використовуються у веб-розробці, їх функції та переваги.

Фронтенд-технології

### **HTML (HyperText Markup Language)**

Мова розмітки, яка використовується для створення структури веб-сторінок.

Версія HTML5 підтримує нові теги та елементи, такі як <video>, <audio>, <canvas>.

### **CSS (Cascading Style Sheets)**

Мова стильових таблиць, яка використовується для опису зовнішнього вигляду та оформлення веб-сторінок.

CSS3 додає такі можливості, як анімація, переходи та гриди.

### **JavaScript**

Основна мова програмування для створення інтерактивних та динамічних елементів на веб-сторінках.

Підтримує роботу з DOM (Document Object Model) та AJAX (асинхронні запити).

Фреймворки та бібліотеки JavaScript:

React: бібліотека для створення інтерфейсів, розроблена Facebook. Підтримує компонентний підхід та односпрямований потік даних.

Angular: фреймворк від Google для створення односторінкових програм (SPA) за допомогою TypeScript.

Vue.js: прогресивний фреймворк для створення інтерфейсів користувача, який легко інтегрується в проекти.

Технології для верстки та стилів:

Bootstrap: популярний фреймворк для створення адаптивних веб-сторінок із визначеними стилями та компонентами.

Sass та Less: препроцесори CSS, які додають можливості змінних, вкладеності та міксинів.

Бекенд-технології

Серверні мови програмування:

Node.js: середовище виконання JavaScript на сервері, яке дозволяє створювати масштабовані та високопродуктивні програми.

Python: мова, яка часто використовується у веб-розробці завдяки фреймворкам, таким як Django та Flask.

PHP: популярна мова для створення динамічних веб-сторінок та роботи з базами даних. Використовується у таких системах, як WordPress.

Ruby: мова програмування, відома своїм фреймворком Ruby on Rails, яка спрощує створення веб-додатків.

Java: використовується для розробки корпоративних веб-додатків, часто разом із фреймворками Spring та Hibernate.

Фреймворки для бекенда:

Express.js: мінімальний фреймворк для Node.js, який полегшує створення веб-додатків та API.

Django: високорівневий фреймворк для Python, відомий своєю швидкістю розробки та вбудованою адміністративною панеллю.

Spring Boot: фреймворк для Java, що спрощує створення мікросервісів і додатків, що масштабуються.

Бази даних:

Реляційні бази даних: MySQL, PostgreSQL широко використовуються для зберігання структурованих даних з підтримкою складних запитів SQL.

NoSQL бази даних: MongoDB, CouchDB – призначені для зберігання неструктурованих даних і часто використовують у проектах з великими обсягами даних.

### **1.3 Особливості розробки сайтів освітніх установ**

Розробка сайтів для освітніх установ має свої унікальні вимоги та виклики, що обумовлені потребами користувачів, специфікою контенту та завданнями, які мають виконувати ці сайти. У цій статті ми розглянемо ключові аспекти розробки таких сайтів, включаючи функціональні вимоги, дизайн, технології та найкращі практики.

Ключові особливості сайтів освітніх установ:

а) багатофункціональність:

- сайти мають підтримувати широкий спектр функцій: інформаційні сторінки, навчальні матеріали, адміністративні ресурси, інтерактивні елементи;

- забезпечення доступу до електронних журналів, бібліотек, систем управління навчальним процесом (LMS);

б) різноманітність контенту:

- необхідність розміщення текстової інформації, мультимедіа (фото, відео, аудіо), інтерактивних елементів (форумів, опитувань);
- контент має бути доступний різним групам користувачів: студентам, викладачам, адміністрації, батькам;

в) високі вимоги до безпеки:

- захист персональних даних студентів, викладачів та співробітників;
- використання SSL-сертифікатів, регулярний аудит безпеки, захист від кібератак;

г) доступність та інклюзивність:

- забезпечення доступу для людей з обмеженими можливостями відповідно до стандартів доступності (WCAG);
- підтримка мультимовності, зокрема для міжнародних студентів;

д) адаптивність та мобільність:

- оптимізація під різні пристрої та розширення екранів (мобільні телефони, планшети, настільні ПК);
- забезпечення швидкого завантаження та зручної навігації на всіх пристроях;

Інтеграція з іншими системами:

- підключення до зовнішніх навчальних платформ, електронних бібліотек, систем обліку;
- взаємодія з соціальними мережами та іншими комунікаційними сервісами.

Процес розробки сайту освітньої установи

Аналіз вимог і планування:

- визначення цілей сайту, цільової аудиторії та основних функцій;
- аналіз потреб студентів, викладачів, батьків та інших зацікавлених сторін.

Розробка технічного завдання (ТЗ):

- документування всіх вимог до функціональності, дизайну, безпеки;
- узгодження з усіма зацікавленими сторонами.

#### Вибір технологій і платформи:

- вибір між використанням CMS (наприклад, WordPress, Joomla) чи розробкою індивідуального рішення;
- визначення необхідних інструментів та фреймворків для розробки.

#### Дизайн і прототипування:

- розробка макетів інтерфейсу, враховуючи принципи юзабіліті та доступності;
- створення інтерактивних прототипів для тестування користувачами.

#### Розробка і тестування:

- написання коду згідно з вимогами ТЗ, інтеграція необхідних функцій;
- регулярне тестування для виявлення та виправлення помилок, забезпечення сумісності з різними пристроями та браузерами.

#### Запуск і підтримка:

- розгортання сайту на обраному хостингу, забезпечення безперебійної роботи;
- регулярне оновлення контенту, виправлення помилок, підтримка безпеки та масштабування функціональності.

#### Найкращі практики

##### Фокус на користувацькому досвіді (UX):

- простота навігації, інтуїтивно зрозумілий інтерфейс;
- використання сучасних веб-дизайн стандартів та принципів юзабіліті.

##### Оптимізація для пошукових систем (SEO):

- використання ключових слів, оптимізація метаданих, швидкість завантаження;
- створення контенту, який відповідає запитам користувачів.

#### Впровадження інноваційних технологій:

- використання штучного інтелекту для персоналізації контенту, чат-ботів для підтримки;
- Ввпровадження адаптивних технологій для покращення взаємодії з користувачами.

#### Акцент на безпеку:

- регулярне оновлення системи та плагінів, використання надійних паролів, шифрування даних;
- проведення навчання співробітників з питань безпеки.

#### Залучення користувачів до процесу розробки:

- проведення опитувань, збору зворотного зв'язку, тестування з участю кінцевих користувачів;
- використання результатів для покращення функціональності та зручності.

#### Постійне вдосконалення та оновлення:

- врахування змін в освітній сфері та технологіях, адаптація сайту під нові вимоги;
- оновлення контенту, удосконалення інтерфейсу на основі нових тенденцій.

Розробка сайтів освітніх установ потребує комплексного підходу та врахування специфічних вимог цієї сфери. Використовуючи найкращі практики та сучасні технології, можна створити ефективний і зручний для користувачів ресурс, який буде сприяти розвитку освітнього процесу та покращенню взаємодії між усіма зацікавленими сторонами.

## 2 РОЗРОБКА САЙТУ ЛЦЕЮ

### 2.1 Постановка завдання

Постановка завдання розпочиналася із неактуальністю сайту який є в наявності. Сайт освітнього закладу який є в наявності застарілий та не настільки зручний в використанні та не має адаптивності для різних пристроїв важкий у користуванні для користувачів. Були такі обговоренні наступні вимоги:

- удосконалення головної сторінки;
- удосконалення для зручного використання сайту;
- використання сучасних технологій для удосконалення дизайну сайту;
- інформування про освітні стандарти;
- удосконалення сторінки про загальну інформацію про викладачів та їх досягнення;
- навчальні програми та плани;
- історія досягнень;
- створення умов для дистанційного навчання;
- створення умов для звязку с іншими освітніми установами для обговорення освітніх процесів ;
- створення умов для звязку міє викладачами, абітурієнтами, учнями та батьками;
- надання абітурієнтам відповіді на їх питання;
- удосконалення сторінки про загальну інформацію про освітній заклад;
- удосконалення сторінки про висвітлення досягнень закладу, учнів, викладачів;
- удосконалення сторінки про зручне використання меню для переходу між сторінками;
- підключення учнів до курсів для поліпшення навчання;



- удосконалення сторінки про стимулювання творчої активності для викладачів, працівників, здобувачів освіти;
- удосконалення сторінки про статут закладу освіти;
- удосконалення сторінки про ліцензії на провадження освітньої діяльності;
- удосконалення сторінки про структура та органи управління закладу освіти;
- удосконалення сторінки про кадровий склад закладу освіти згідно з ліцензійними умовами;
- удосконалення сторінки про освітні програми;
- удосконалення сторінки про наявність вакантних посад;
- удосконалення сторінки про матеріально-технічне забезпечення;
- удосконалення сторінки про центр кар'єри;
- удосконалення сторінки про методичну роботу;
- удосконалення сторінки про національне патріотичне виховання;
- удосконалення сторінки про абітурієнту на замітку;
- удосконалення сторінки про психологічну службу;
- удосконалення сторінки про батьківську сторінку;
- удосконалення сторінки про виховну роботу;
- удосконалення сторінки про план роботи;
- удосконалення сторінки про нормативні документи;
- удосконалення сторінки про профорієнтацію;
- удосконалення сторінки про правила вступу та документи для вступу;
- удосконалення сторінки про контактну інформацію.

### **2.1.1 Опис вимог до сайту**

Головна сторінка розглядалася як більш інформаційна сторінка для профорієнтації, вона має містити коротку актуальну інформацію і постійне оновлення та повідомлення викладачів, працівників та учнів.

Зручне використання полягало в зручності пошуку тієї чи іншої інформації та зручності використання для викладачів, які можуть викладати свої посилання на відео, можливість перегляду презентацій та документів таких як Word, PowerPoint, Excel.

Новини сайту розміщені на верхньому меню сторінки для більш зручного використання та можливість перегляду актуальних новин, та новини які були раніше написані і щоб користувачі могли розуміти коли це відбулося тобто (дату, місяць, рік). Якщо користувач захоче бачити більш старіші новини то він зможе перейти на старий сайт освітнього закладу.

### **2.1.2 Структура сайту**

Планування структури веб-сайту розроблялося за принципом старої версії веб-сайту нашого освітнього закладу. До веб-сайту було виділено наступні основні розділи:

- головна сторінка;
- про нас;
- історія закладу;
- навчальний процес;
- планування діяльності;
- методичні матеріали;
- абітурієнтам;
- студентам;
- розклад занять;
- інформація про стипендії;
- бібліотека та ресурси;
- доступ до електронної бібліотеки та навчальних матеріалів;
- контакти;
- адреса, телефон та електронна пошта закладу;
- карта проїзду та години прийому;
- форма для зворотного зв'язку;

- педагогічний колектив;
- публічна інформація;
- психологічна служба;
- наші координати.

### **2.1.3 Вимоги до дизайну сайту**

Дизайн сайту освітнього закладу повинен бути інтуїтивно зрозумілим та зручним. Було обрано дизайн з акцентом на зручність навігації та доступ до основних функцій. І такий дизайн застосовується для усіх сторінок цього сайту (рис. 2.1).

Основні елементи інтерфейсу включають:

- дизайн: дизайн з збалансованою кольоровою палітрою, та легкі для сприйняття шрифти;
- навігація: навігація з доступним головним меню;
- контент: контент з легко читабельною контентною структурою;
- користувацький досвід (UX) : проста взаємодія з елементами, валідація форм та адаптивність веб-сайту;
- доступність: доступність інтерфейсу для користувачів з обмеженими можливостями та адаптивність до взаємодії з технічними пристроями.

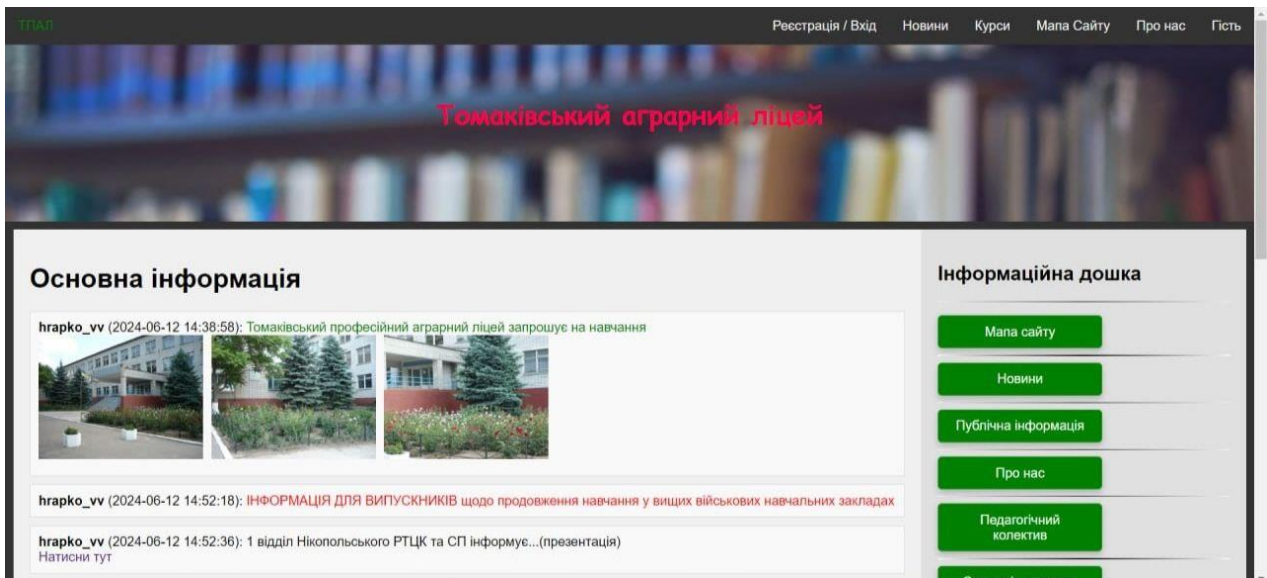


Рисунок 2.1 – Дизайн сайту

## 2.2 Обґрунтування вибору технологій розробки сайту

Для розробки сайту використовувався стек розробки, який включає:

Клієнтська частина: HTML+CSS+Bootstrap+JavaScript+React

Серверна частина: PHP+MySQL

Мова HTML для розмітки тексту використовується для розмітки веб-сторінок у браузерях. Щоб документ могли переглядати використовуються протоколи HTTP/HTTPS за допомогою цих протоколів відкривається код с локального сервера та інтерпретується в інтерфейс на екрані.

За допомогою CSS можна контролювати, як елементи розміщуються на сторінці. Це включає вирівнювання, розміщення елементів в рядок або колонку, створення сіток і складних макетів.

CSS підтримує моделі компоунвання, такі як Flexbox та Grid, які спрощують створення адаптивних та складних макетів.

Забезпечення адаптивного дизайну:

- CSS дозволяє створювати сайти, які добре виглядають і працюють на пристроях з різними розмірами екранів та роздільною здатністю, від мобільних телефонів до настільних комп'ютерів.

За допомогою медіазапитів (media queries) можна змінювати стилі в залежності від характеристик пристрою, таких як ширина екрана.

Створення візуальних ефектів та анімації:

- CSS підтримує створення анімацій, переходів та інших динамічних ефектів, які покращують взаємодію користувача.

Ви можете анімувати зміну властивостей елементів, таких як колір, розмір або позиція, без використання JavaScript.

Підтримка тем і стилів користувача:

- CSS дозволяє створювати різні теми та стилі, які можна легко змінювати без зміни структури HTML.

Це спрощує кастомізацію та підтримку сайтів, особливо у великих проектах.

Поділ контенту та оформлення:

- CSS допомагає відокремити вміст веб-сторінки (HTML) від її оформлення. Це робить код структурованішим і легко підтримуваним.

Стилі можна зберігати в окремих файлах, що полегшує керування та перевикористання стилів на різних сторінках.

Bootstrap – це популярний фреймворк з відкритим вихідним кодом, розроблений для створення адаптивних та сучасних веб-сайтів та веб-додатків. Він надає готовий набір інструментів для спрощення процесу розробки і дозволяє швидко створювати привабливі і функціональні інтерфейси користувача.

Основні цілі та функції Bootstrap

Спрощення та прискорення розробки:

- Bootstrap надає набір готових компонентів та стилів, що дозволяє розробникам створювати сторінки швидше та без необхідності писати багато CSS-коду з нуля.

Адаптивний та чуйний дизайн:

- за допомогою системи сіток (grid system) та медіазапитів Bootstrap дозволяє легко створювати макети, які адаптуються до різних пристроїв та роздільної здатності екранів.

Підтримка крос-браузерної сумісності:

- Bootstrap включає стилі та компоненти, які протестовані та працюють однаково у різних сучасних браузерах, що спрощує розробку крос-браузерних сайтів.

Готові компоненти інтерфейсу користувача:

- Bootstrap включає стандартні елементи інтерфейсу, такі як кнопки, форми, навігаційні панелі, модальні вікна та інші компоненти, які можна використовувати у своїх проектах.

Дотримання сучасних стандартів веб-дизайну:

- Bootstrap підтримує новітні стандарти HTML5 та CSS3, що дозволяє створювати сучасні та відповідні вимогам веб-сторінки.

Модульність та налаштування:

- Bootstrap надає можливість налаштування стилів та компонентів через змінні SASS, що дозволяє адаптувати зовнішній вигляд та функціональність фреймворку під потреби конкретного проекту.

React – це популярна бібліотека JavaScript з відкритим вихідним кодом, розроблена для створення інтерфейсів користувача. Вона дозволяє розробникам створювати складні та інтерактивні веб-додатки, спрощуючи процес розробки та підтримуючи високу продуктивність. React використовується для побудови компонентних структур, що полегшує керування станом програми та покращує повторне використання коду.

Основні функції та призначення React

Створення компонентних інтерфейсів користувача:

- React дозволяє розбивати інтерфейс на незалежні компоненти, які можна легко розробляти, тестувати та повторно використовувати;
- компоненти можуть бути класами або функціями, і кожен компонент визначає власний рендеринг і логіку.

Управління станом програми:

- React використовує концепцію стану (state) для керування даними та реакцією на їх зміни.

Зміна стану призводить до повторного рендерингу компонентів, що забезпечує оновлення інтерфейсу користувача в реальному часі.

Односпрямований потік даних:

- у React дані передаються від батьківських компонентів до дочірніх через властивості (props), що забезпечує передбачуваність та полегшує налагодження.

Це допомагає створити чітку архітектуру програми, де компоненти мають ясні та зрозумілі зв'язки.

Віртуальний DOM (Virtual DOM):

- React використовує віртуальний DOM для оптимізації оновлень інтерфейсу користувача;
- віртуальний DOM – це копія реального DOM, яка оновлюється на кожній зміні стану, а потім синхронізується з реальним DOM, мінімізуючи кількість маніпуляцій з реальним DOM;
- React має велику екосистему бібліотек та інструментів, таких як React Router для маршрутизації та Redux для керування станом.

Він також легко інтегрується з іншими JavaScript-бібліотеками та фреймворками, що робить його універсальним інструментом для веб-розробки.

PHP (Hypertext Preprocessor) — це широко використовувана мова програмування з відкритим вихідним кодом, спеціально розроблена для створення динамічних та інтерактивних веб-додатків. Він виконується на сервері та використовується для обробки даних, створення та управління веб-сторінками та інтеграції з різними базами даних.

Основні цілі та функції PHP.

Створення динамічних веб-сторінок:

- PHP дозволяє створювати веб-сторінки, вміст яких може змінюватись в залежності від дій користувача, запитів до бази даних або інших факторів.

Він генерує HTML код на сервері, який потім відправляється на клієнтський браузер.

Інтерактивність та обробка форм:

- PHP використовується для обробки даних, введених користувачами через форми, та виконання відповідних дій, таких як збереження даних до бази даних або надсилення електронної пошти.

Він забезпечує валідацію та захист даних, введених користувачами.

Робота з базами даних:

- PHP підтримує роботу з різними системами управління базами даних (СУБД), такими як MySQL, PostgreSQL, SQLite та інші.

Він дозволяє виконувати запити до баз даних, витягувати та оновлювати дані, а також керувати транзакціями та зв'язками між таблицями.

Обробка файлів та роботи з сервером:

- PHP надає функції для роботи з файловою системою, такі як читання, запис, створення та видалення файлів.

Він також може взаємодіяти з сервером для виконання різних завдань, таких як керування сесіями, відправлення та отримання файлів, обробка зображень та інших мультимедійних даних.

### 2.2.1 Обґрунтування вибору СУБД

MySQL – це одна з найбільш популярних систем управління базами даних (СУБД) з відкритим вихідним кодом, що широко використовується для розробки веб-додатків та управління даними. Вона забезпечує надійність, високу продуктивність і масштабованість, що робить її чудовим вибором для різних проектів, від невеликих веб-сайтів до великих корпоративних програм.

Основні причини вибрати MySQL

Широке поширення та підтримка спільноти:

- MySQL використовується безліччю організацій і має величезну спільноту, що забезпечує доступ до великої кількості ресурсів, документації та підтримки;



- існує безліч навчальних матеріалів, форумів та курсів, що полегшує процес навчання та впровадження.

Відкритий вихідний код та безкоштовне використання:

- MySQL є програмним забезпеченням з відкритим вихідним кодом, що дозволяє використовувати його безкоштовно та модифікувати за необхідності;
- це робить MySQL доступним вибором для стартапів, малого бізнесу та індивідуальних розробників.

Висока продуктивність та масштабованість:

- MySQL оптимізовано для роботи з великими обсягами даних та високим навантаженням, забезпечуючи швидку та ефективну роботу додатків;
- він підтримує горизонтальне та вертикальне масштабування, що дозволяє адаптуватися до зростаючих потреб вашого бізнесу.

Гнучкість та підтримка різних типів даних:

- MySQL підтримує широкий спектр типів даних, включаючи числові, текстові, часові та просторові типи даних;
- він надає потужні можливості для керування складними структурами даних та виконання складних запитів.

Сумісність та інтеграція:

- MySQL легко інтегрується з різними мовами програмування та платформами, такими як PHP, Python, Java та багато інших;
- він підтримує стандарти SQL та забезпечує сумісність з різними базами даних та веб-серверами.

Безпека та надійність:

- MySQL надає потужні інструменти для забезпечення безпеки даних, такі як керування доступом, шифрування та резервне копіювання;
- він активно підтримується та оновлюється, що забезпечує захист від уразливостей та надійну роботу.

### 2.3. Особливості реалізації сайту

Коди основних сторінок сайту представлено в додатку А. В основному, використано стандартні алгоритми реалізації функцій реєстрації та авторизації, використання сесій і т.п.

Особливості реалізації пов'язані з тим, як відбувається рендеринг сторінок з індивідуальним набором курсів, які доступні певному учню, відповідно до даних його авторизації. Це реалізовано за допомогою технології React.

В базі даних сайту зберігаються дані про те, які ідентифікатори курсів вивчає учень з певним ідентифікатором.

Спочатку спрацьовує файл, який відбирає ці дані та розміщує їх в файл JSON (рис. 2.2).

```

<?php
session_start();
$conn = new mysqli('localhost', 'root', '', 'travel_company');
if ($conn->connect_errno) {
    echo "неможливо підключитися до MySQL: " . $conn->connect_error();
}
if (isset($_SESSION['loggedin']) && $_SESSION['loggedin'] === true) {
    $kod = $_SESSION['id'];
}
$result = $conn->query("SELECT * FROM courses INNER JOIN user_courses
ON courses.id=user_courses.course_id WHERE user_courses.user_id=$kod");
$course = [];
while($myrow = $result->fetch_assoc()) {
    $course[] = $myrow;
}
//Зберігаємо дані отриманого масиву у текстовий файл формату JSON
echo json_encode($course);
?>

```

Рисунок 2.2 – Листинг файлу для визначення даних

Далі, описуємо скрипт React, який формує картки для курсів, які були відібрані в попередньому файлі.

Реалізація коду представлена у додатку Б.

І цей скрипт підключається до сторінки та виводить інформацію у підготовлену область root.

## 3 РЕЗУЛЬТАТИ РОЗРОБКИ САЙТУ

### 3.1 Встановлення сайту

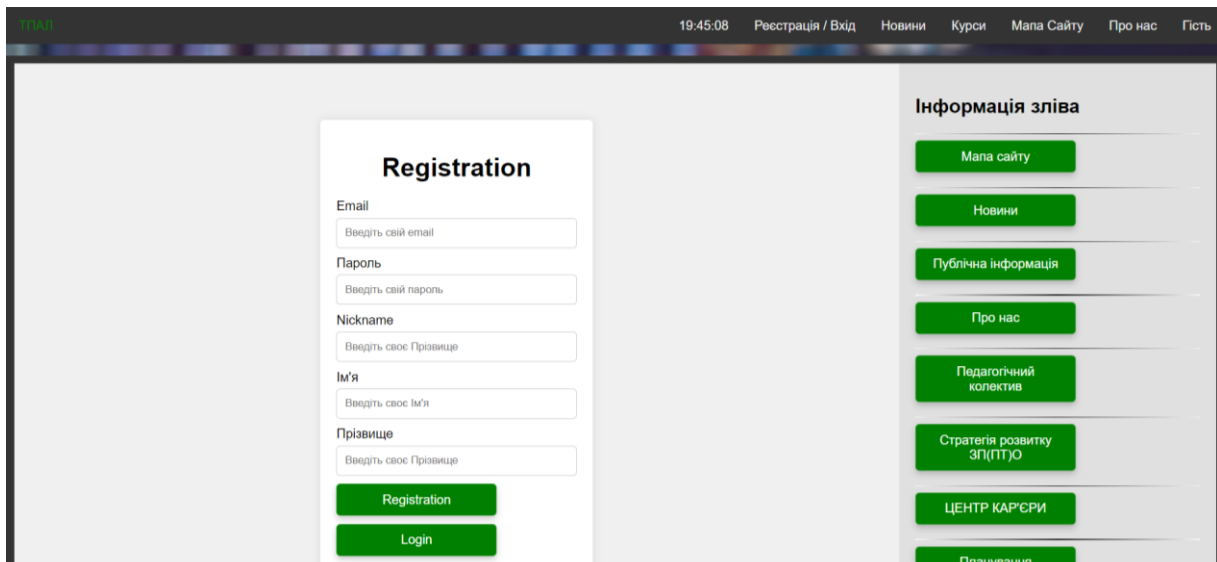
Для того, щоб використовувати розроблений сайт адміністратору необхідно виконати наступні кроки:

- реєстрація;
- авторизація

### 3.2 Використання сайту

Після встановлення сайту на хостингу користувачі можуть отримувати наступні можливості:

- реєстрація (рис. 3.1);



The image shows a screenshot of a web application's registration page. At the top, there is a navigation bar with the time '19:45:08' and several menu items: 'Реєстрація / Вхід', 'Новини', 'Курси', 'Мапа Сайту', 'Про нас', and 'Гість'. The main content area is divided into two columns. The left column contains a 'Registration' form with the following fields: 'Email' (with placeholder 'Введіть свій email'), 'Пароль' (with placeholder 'Введіть свій пароль'), 'Nickname' (with placeholder 'Введіть своє Прізвище'), 'Ім'я' (with placeholder 'Введіть своє Ім'я'), and 'Прізвище' (with placeholder 'Введіть своє Прізвище'). Below these fields are two green buttons: 'Registration' and 'Login'. The right column is titled 'Інформація зліва' and contains a vertical list of green buttons: 'Мапа сайту', 'Новини', 'Публічна інформація', 'Про нас', 'Педагогічний колектив', 'Стратегія розвитку ЗП(ПТ)О', 'ЦЕНТР КАР'ЄРИ', and 'Планування'.

Рисунок 3.1 – Реєстрація

- авторизація (рис. 3.2);

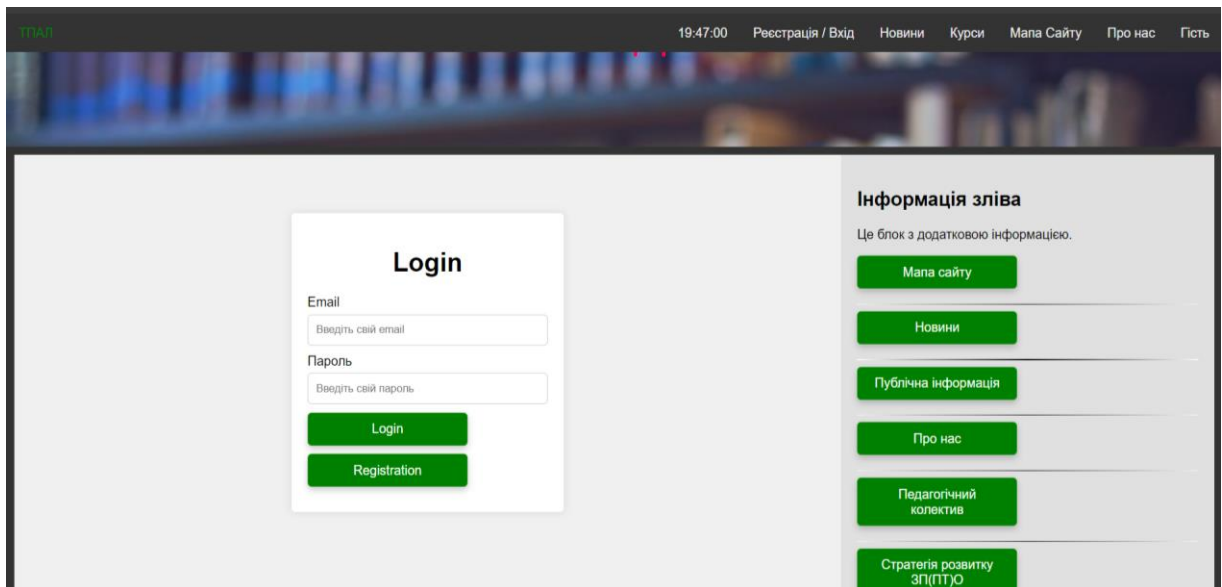


Рисунок 3.2 – Авторизація

– підключення до курсів (рис. 3.3).

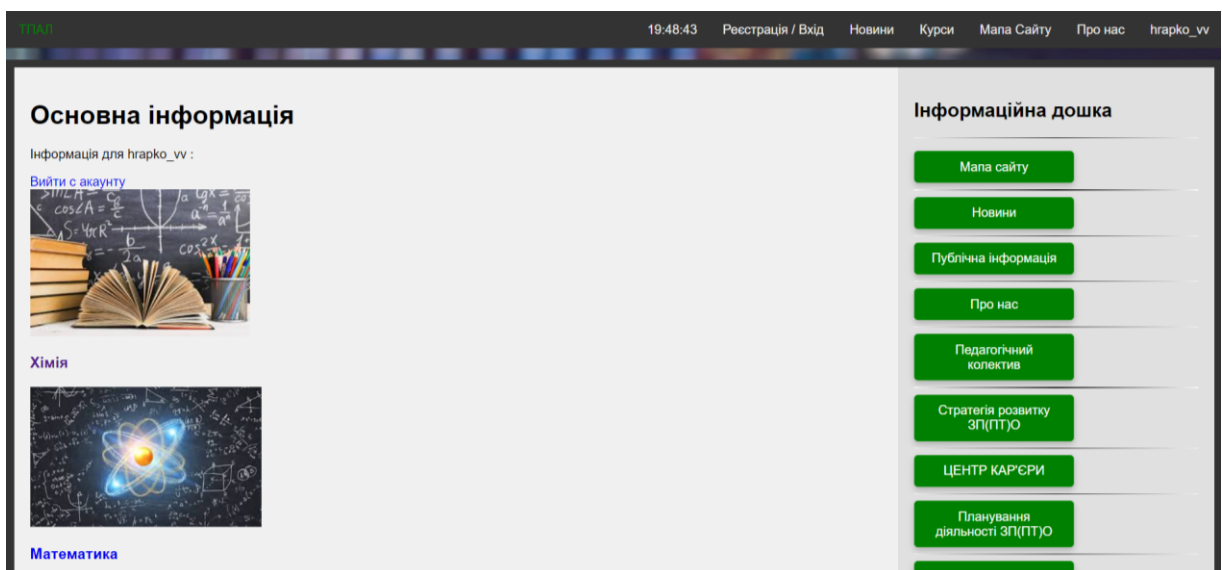


Рисунок 3.3 – Курси користувачів

## ВИСНОВКИ

У світі сайт є найважливішим інструментом для будь-якої освітньої організації, включаючи ліцеї. Він виконує безліч функцій та служить платформою для взаємодії зі студентами, батьками, викладачами та громадськістю. Розглянемо ключові причини, чому наявність сайту ліцею є важливою та необхідною.

В результаті виконання даної кваліфікаційної роботи був розроблений файл для Томаківського аграрного ліцею. Він відповідає більшості вимог замовника, однак ще потребують вдосконалення наступні функції:

PHP – серверна частина

React – сторінка користувача

В процесі виконання роботи були застосовані на практиці знання з основних технологій реалізації web-додатків, та вивчені основи бібліотеки React.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Закон України про фахову передвищу освіту. Кодекси. URL : [https://kodeksy.com.ua/pro\\_osvitu/statja-30.htm](https://kodeksy.com.ua/pro_osvitu/statja-30.htm) (дата звернення : 05.01.2024).
2. Наталія Бойко. Створюємо сайт закладу освіти: ключові кроки. URL : <https://naurok.com.ua/post/stvoryuemo-sayt-zakladu-osviti-klyuchovi-kroki> (дата звернення: 09.01.2024).
3. Що таке веб-дизайн і з чим його їдять?. Savelink. URL : <http://savelink.org.ua/shho-take-veb-dizajn-i-z-chim-jogo-yidyat/> (дата звернення : 28.01.2024).
4. Приклади хорошого веб-дизайну. Stfalcon. URL : <https://stfalcon.com/ru/blog/post/good-website-design-examples> (дата звернення : 10.03.2024).
5. Дизайн сторінки CSS. Webformyself. URL : <https://webformyself.com/category/verstka-2/css3-verstka-2/> (дата звернення : 12.03.2024).
6. Як создать свій сайт? URL : <https://webformyself.com/> (дата звернення : 15.03.2024).
7. CMS WordPress. Webformyself. URL : <https://webformyself.com/category/cms-2/wp/> (дата звернення : 28.03.2024).
8. JavaScript – Клієнтська мова програмування jQuery. Webformyself. URL : <https://webformyself.com/validaciya-dannux-na-php-chast-2/> (дата звернення : 10.04.2024).
9. Локальний веб-сервер. Валідація даних в PHP. URL : <https://webformyself.com/category/javascriptajax-2/javascriptajax/> (дата звернення : 25.04.2024).
10. Локальный веб-сервер. Создание собственного фреймворка. Авторизация администратора. URL : <https://webformyself.com/31-sozdanie-frejmvorka-avtorizaciya-administratora-2/> (дата звернення : 18.05.2024).

11. Использование компонентов Bootstrap с пользовательским JavaScript.

URL : <https://webformyself.com/ispolzovanie-komponentov-bootstrap-s-polzovatelskim-javascript/> (дата звернення : 24.05.2024).

## ДОДАТОК А

### Коди основних сторінок

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title id="title">Форма реєстрації</title>
<meta charset="UTF-8">
</head>
<link rel="stylesheet" type="text/css" href="style/style.css">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<meta name="generator" content="NeonHTML 4.5">
<html>
<head>
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.4/jquery.min.js"></script>
<script src="https://use.fontawesome.com/b9decc4b30.js"></script>
<script type="text/javascript" src="js/index.js"></script>
<script type="text/javascript" src="js/index1.js"></script>
<script type="text/javascript" src="js/index2.js"></script>
<script type="text/javascript" src="js/index3.js"></script>
<script type="text/javascript" src="js/1.js"></script>
<script type="text/javascript" src="js/2.js"></script>
<script type="text/javascript" src="js/4.js"></script>
<script type="text/javascript" src="js/3.js"></script>
<script type="text/javascript" src="js/5.js"></script>
<script type="text/javascript" src="1.js"></script>
<?php
// для підключення файлу
include 'php/course_username.php';
?>
<?php
// для підключення файлу
include 'style/style1.html';
?>
<title>ТПАЛ</title>

<div class="topnav" id="myTopnav">
<li><span><a href="/" title="Aieiaia"><font color="Green"><i>ТПАЛ</i></font></a></span></li>
<a href="javascript:void(0);" class="icon" onclick="myFunction()">
<i class="fa fa-bars"></i>
</a>
<form id="form54" style="display: none;">
<a href="#"><input type="search" name="q" value="" placeholder="ііоое.."><input type="submit" name="" value="ііоое"></a>
</form>
<a href="#" onclick="disp(document.getElementById('form2'))" title="<?php session_start(); if(isset($_SESSION['loggedin']) &&
$_SESSION['loggedin'] === true) {echo $_SESSION['username'];} else {echo "Гість";}?>"><?php session_start(); if(isset($_SESSION['loggedin']) &&
$_SESSION['loggedin'] === true) {echo $_SESSION['username'];} else {echo "Гість";}?></li></a>

<?php
// Включаем содержимое из другой страницы
include 'load_content/load_content1.php';
?>

<a href="#"><body onload="startTime()"><div id="timer"></div></body></a>
</div>

<body>

<div id="text">
<div class="layer1">
<div class="text_title"><h1><p align="center">
<font color="#FF004C">Томаківський аграрний лицей </font></h1></p>
</div>
</div>
</div>

<div class="container">

```





```
</div>

<script>
const translations = {
  'uk': {
    'title': 'Переклад сайту',
    'content': 'Це приклад вмісту на українській мові.',
    'button': 'English'
  },
  'en': {
    'title': 'Website Translation',
    'content': 'This is an example content in English.',
    'button': 'Українська'
  }
};

let currentLang = 'uk';

function translate() {
  currentLang = currentLang === 'uk' ? 'en' : 'uk';
  document.getElementById('title').innerText = translations[currentLang]['title'];
  document.getElementById('content').innerText = translations[currentLang]['content'];
  document.getElementById('translateButton').innerText = translations[currentLang]['button'];
}

window.onload = function() {
  translate(); // Перевести початковий вміст на українську мову
};
</script>

</body>

<?php
// Включаем содержимое из другой страницы
include 'load_content/load_content2.php';
?>

</html>
```

## ДОДАТОК Б

### Формування картків для курсів

```
function Course(props) {
  return (
    <div className="card">
      <div className="card-img">

        <img className="course-img" src={props.course.image} />

        <h3>

          <a className="course-ref"
            href={"course.php?id="+props.course.id}>{props.course.title}
          </a>
        </h3>
      </div>
    </div>
  )
}

/* Описуємо клас компонента App (додаток), що успадковується від компонента React у якому є його стан (state) – в даному випадку набір
всіх можливих значень та рендеринг (render) – відтворення на сторінці шаблону зі зв'язаним із ним станом */
class App extends React.Component
{
  /*оголошуємо об'єкт, який описує стан компонента state*/
  state={courses:[]}

  /*Створюємо об'єкт, який асинхронно може запросити у сервера дані через створений раніше файл api.php і розпізнати ці дані як JSON.
Для цього використовується сучасний інтерфейс JS, який називається Fetch API*/
  fetchQuotes = () => {
    this.setState({...this.state, isFetching: true})
    fetch("api.php")
      .then(response => response.json())
      /*Тут отримані дані зв'язуються з об'єктом, описаним вище як стан компонента */
      .then(result => this.setState({courses: result, isFetching: false}))
      .catch(e => console.log(e));
  }

  /*У компонентів React є методи життєвого циклу, які дозволяють зберігати актуальний стан. Метод componentDidMount() викликається,
коли компонент стає доступним. Тому тут ми безпосередньо отримуємо дані */
  componentDidMount() {
    this.fetchQuotes()
  }

  /*Тут ми вказуємо як на сторінці буде виконуватися малювання (рендеринг) компонента*/
  render(){
    console.log(this.state)
    return(
      /*Створюється обгортка, яка може бути використана потім у CSS*/
      <div className="app">
        <div className="list">
```

```
<div className="row">
{
  /*Виконується відображення кожного поточного стану компонента з використанням раніше описаної функції представлення Tours*/
  this.state.courses.map(course=>{
    return(
      <div class="col">
        <Course course={course} />
      </div>
    )
  }
)
}
</div>
</div>
</div>
)
}
}
}
//Виведення відмальованих шаблонів у відведене для них місце на сторінці index.php
ReactDOM.render(<App />, document.getElementById('root'))
```