

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

Розробка навчальної системи для підготовки учнів до ЗНО

Текстова частина до курсової роботи

за спеціальністю „Інженерія програмного забезпечення” 121

Керівник курсової роботи

к.т.н., доц. Жежерун О.П.

(підпис)

“ ____ ” _____ 2022 р.

Виконав студент МП ІПЗ-1

Ревков Є.Г.

“ ____ ” _____ 2022 р.

Київ 2022

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики
ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем,
доцент, к.т.н. _____ Жежерун О.П.

(підпис)

«_____» _____ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

Студенту Ревкову Євгенію Геннадійовичу

Факультету інформатики 1 р.н. магістерської програми

ТЕМА: Розробка навчальної системи для підготовки учнів до ЗНО

Зміст текстової частини до курсової роботи:

Календарний план

Вступ

Розділ 1. Аналіз предметної області. Постановка завдання курсової роботи

Розділ 2. Теоретичні відомості

Розділ 3. Опис реалізації програмного продукту

Висновки

Додатки(за необхідністю)

Список використаної літератури

Дата видачі «_____» _____ 2022 р. Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Тема: Розробка навчальної системи для підготовки учнів до ЗНО

Календарний план виконання роботи:

№	Назва етапу виконання	Термін	Примітка
1.	Отримання теми курсової роботи	30.10.2022	
2.	Пошук тематичної літератури	15.11. 2022	
3.	Ознайомлення з літературою	01.12. 2022	
4.	Вивчення аналогів	20.12. 2022	
5.	Планування веб-сайту	05.01. 2022	
6.	Ознайомлення з платформою Node.js	12.01. 2022	
7.	Ознайомлення з модулем Express	23.01. 2022	
11.	Реалізація клієнт-серверної частини	08.04. 2022	
12.	Оформлення сторінок	14.04. 2022	
13.	Написання текстової частини	01.06. 2022	
14.	Перегляд змісту роботи керівником	08.06. 2022	
15.	Внесення змін до курсової роботи відповідно до зауважень наукового керівника	09.06. 2022	

Студент Ревков Є.Г.

Керівник Жежерун О.П.

«____» _____

Зміст

Перелік умовних позначень	5
Вступ	6
Актуальність, наукове та практичне значення обраної теми	6
Мета та завдання курсової роботи	6
Використане програмне забезпечення	7
РОЗДІЛ 1 Аналіз предметної області. Постановка завдання курсової роботи	7
1.1 Аналіз сучасного стану питання та обґрунтування теми	7
1.2 Огляд існуючих аналогів розробки	8
РОЗДІЛ 2 ТЕОРЕТИЧНІ ВІДОМОСТІ	9
2.1 PostgreSQL	9
2.2 Express	9
2.3 React	9
2.4 Node	9
2.5 Three	10
РОЗДІЛ 3 Опис реалізації програмного продукту	10
3.1 Аналіз технічного завдання	10
3.1.1 Цільова аудиторія	10
3.1.2 Опис функціоналу сторінок та елементів інтерфейсу	10
3.2 Обґрунтування алгоритму й структури програми	14
3.3 Обґрунтування вибору засобів розробки	15
3.4 Створення об'єктів і розробка головної програми	19
3.4.1 Початкові налаштування	19
3.4.2 Реєстрація та авторизація	20
3.4.3 Створення проекту на React	20
3.4.4 Маршрутизація на клієнтській стороні	22
3.4.5 Відображення об'ємних фігур у просторі	22
Висновки	22
Список використаної літератури та електронних ресурсів	24
Додатки	25

Перелік умовних позначень

HTML – HyperText Markup Language

CSS – Cascading Style Sheets (Каскадні таблиці стилів)

JS – JavaScript

TS - TypeScript

СКБД – система керування базами даних

БД – база даних

PERN – P – PostgreSQL, E – Express.js, R – React.js, N – Node.js

JSX – JS + XML

SQL - Structured Query Language

SaSS - Syntactically Awesome Style Sheets

SCSS - Sassy CSS

Вступ

Актуальність, наукове та практичне значення обраної теми

Тема даної курсової роботи - Розробка навчальної системи для підготовки учнів до ЗНО. Актуальність цієї теми курсової роботи полягає в тому, що в нинішній час в учнів старших класів з'являється з кожним роком все більше проблем з вивченням шкільного матеріалу та підготовки до зовнішнього незалежного оцінювання. Останні пару років процес навчання та викладання в школах погіршився. Основною проблемою було засвоєння учнями такого предмету, як геометрія. Вчителям дуже складно зобразити коректний та зрозумілий малюнок до задачі, зобразивши його та демонструючи свій екран під час дистанційного навчання.

Для вирішення цієї проблеми, було вирішено створити відповідний застосунок для полегшення вивчення та сприйняття обраного матеріалу, а саме стереометрії – розділу геометрії, в якому вивчаються фігури у просторі.

Мета та завдання курсової роботи

Метою даної курсової роботи є створення сайту, який поліпшить вивчення нового матеріалу та підготовку до ЗНО для учнів старших класів навчальних закладів.

Завданням даної курсової роботи є розробка веб-застосунку для підготовки до ЗНО та доступного та зручного, враховуючи недоліки існуючих аналогів, вивчення матеріалу зі стереометрії використовуючи стек технологій React, TypeScript, PostgreSQL, Node.

Використане програмне забезпечення

- **Мови програмування**
 - HTML, CSS, SaSS, JavaScript, TypeScript
- **UI/UX фреймворки**
 - React
 - UIKit
- **Платформи**
 - Node
- **Основні бібліотеки**
 - Express
 - Three
 - LightGL
- **СУБД**
 - PostgreSQL

РОЗДІЛ 1 Аналіз предметної області. Постановка завдання курсової роботи

1.1 Аналіз сучасного стану питання та обґрунтування теми

Основні можливості в розробленому застосунку – проходження курсу з обраного предмету та можливість вирішення задач зі стереометрії шляхом взаємодії з фігурами у просторі на веб-сайті. Завдяки цьому функціоналу, полегшується сприйняття 3Д простору, а ніж на звичайному малюнку. Зазвичай людина, вирішуючи задачу зі стереометрії, дуже розгублюється під час вказання позначок на просторій фігурі у двовимірному просторі. І завдяки цьому застосунку, користувач зможе взаємодіяти з фігурою, обертаючи, збільшуючи, зменшуючи її і т.д.

1.2 Огляд існуючих аналогів розробки

В інтернет-просторі присутня дуже велика кількість застосунків пов'язаних з вивченням стереометрії, але в жодному з них немає зручного функціоналу. Велика кількість веб-додатків, у яких не передбачена зручність у використанні. І основна кількість шкільних курсів зі стереометрії не являють собою зручну систему, в якій використовують як зображення для задачі - 3д фігуру у двомірному просторі, що може викликати велику кількість проблем у користувача при початковому та подальшому вивченні та засвоєнні матеріалу. З усіх доступних ресурсів є такий застосунок, як Geogebra, котрий описує головний функціонал даної курсової роботи, а саме процес роботи з об'ємними фігурами у просторі. Головною проблемою даного застосунку є незручна реалізація взаємодії з об'єктами та процес вивчення матеріалу. Основними з них є:

- Велика кількість позначок, котру не можна редагувати: прибирати, додавати, змінювати;
- Невелика кількість взаємодії з даними задачі;
- Відсутня можливість розв'язання задач у реальному часі;
- Через погану верстку з'являється тяжке сприйняття задачі;
- Незручна навігація вибору курсу.

Постановка задачі

Розробити зручний, адаптивний під мобільні пристрої веб-застосунок з для підготовки до ЗНО та вирішення задач зі стереометрії шляхом взаємодії з фігурами у просторі, на якому будуть доступні такі основні можливості:

- Проходження курсу
 - Проходження звичайного матеріалу
 - Можливість проходження розділу стереометрії оригінальним та зручним способом

- Проходження тестування
- Отримання результатів тестування
 - Перегляд результатів
 - Можливість перездати тест

РОЗДІЛ 2 ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1 PostgreSQL

PostgreSQL – це популярна вільна об'єктно-реляційна система управління базами даних. PostgreSQL базується на мові SQL та підтримує численні можливості.

2.2 Express

Express - це мінімалістичний і гнучкий веб-фреймворк, написаний на JavaScript і працює всередині середовища виконання Node.js, надає широкий набір функцій для мобільних і веб-прикладів.

Хоча сам Express є досить мінімалістським, розробники створили сумісні пакети проміжного програмного забезпечення для вирішення практично будь-яких проблем веб-розробки. Існують бібліотеки для роботи з файлами cookie, сеансами, логінами користувачів, параметрами URL, даними POST, заголовками безпеки та багатьма іншими.

2.3 React

React - це бібліотека для створення користувацьких інтерфейсів. React не є фреймворком – він навіть не розрахований виключно для веб-розробки. Він використовується для візуалізації та зв'язку з іншими бібліотеками

2.4 Node

Node.js надає середу виконання коду JavaScript, який побудований на основі рухомого JavaScript Chrome V8, який дозволяє транслювати виклики мови JavaScript у машинний код. Node.js перш за все призначений для

створення серверних додатків на мові JavaScript. Хоча також існують проекти щодо написання настільних програм (Electron) і навіть створення коду для мікроконтролерів. Але раніше всього ми говоримо про Node.js, як про платформу для створення веб-застосунків.

2.5 Three

Three.js – це бібліотека JavaScript, що містить набір готових класів для створення та відображення інтерактивної 3D графіки у WebGL.

Three.js для WebGL це те ж саме, що jQuery для JavaScript. Бібліотека пропонує декларативний синтаксис і абстрагує від головного болю пов'язаного з 3D в браузері. Давайте проведемо загальний огляд та подивимося, як розпочати роботу, якщо ви новачок у світі 3D.

РОЗДІЛ 3 Опис реалізації програмного продукту

3.1 Аналіз технічного завдання

3.1.1 Цільова аудиторія

Розроблений сайт призначений для людей, котрі прагнуть засвоїти новий матеріал шкільний матеріал та підготуватись до Зовнішнього Незалежного Оцінювання (ЗНО). Також цей сайт можуть використовувати вчителі для підготовки учням демонстрації взаємодії з об'ємними фігурами в просторі під час проведення уроку.

Цільова аудиторія – учні старшої школи, люди, які готуються до ЗНО, студенти, репетитори, та вчителі.

3.1.2 Опис функціоналу сторінок та елементів інтерфейсу

Так як проект знаходиться на етапі розробки, він має включати в себе наступні пункти:

Навігаційне меню:

- Присутнє на кожній сторінці;
- Фіксованість;

- Елементи навігації;
 - Логотип;
 - Дропдаун з вибором класу та вибору предмету;
 - Посилання на тести.
- Для неавторизованого користувача містить посилання курси, сторінку реєстрації, сторінку авторизації

Футер

- Логотип з посиланням на головну сторінку;
- Містить посилання на сторінки;
- Містить посилання на соціальні мережі;
- Містить посилання на сторінку підтримку.

Головна сторінка

- Hero Section, в якій буде прустній заголовок, короткий опис та посилання на сторінку підготовчих курсів;
- Секція зі статистикою результатів складання ЗНО протягом останніх 10 років. Статистика буде мати вигляд графіку та дропдауну, в якому можна буде обрати рік складання;
- Секція з описом.

Сторінка авторизації

- Поле вводу для логіну;
- Поле вводу для паролю;
- Кнопка підтвердження авторизації;
- Посилання на сторінку реєстрації;
- Посилання на сторінку відновлення паролю;
- Після успішної авторизації редірект на сторінку, з якої був редірект на сторінку авторизації.

Сторінка реєстрації

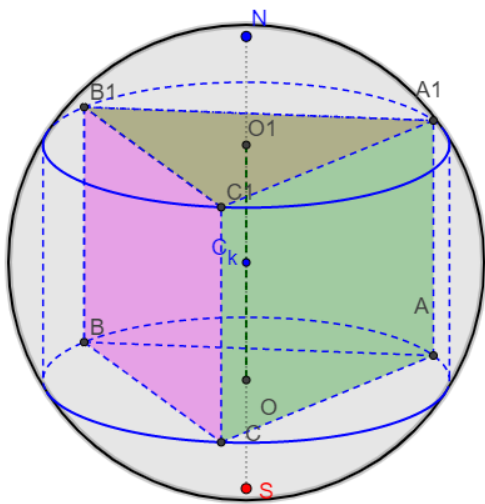
- Поле вводу для ім'я;
- Поле вводу для прізвища;
- Поле вибору статусу (студент, учень, вчитель тощо) ;
- Поле вводу для пошти;
- Поле вибору цілей користувача;
- Поле вводу для паролю;
- Поле вводу для підтвердження паролю;
- Кнопка підтвердження реєстрації;
- Посилання на сторінку авторизації;
- Після успішної реєстрації редірект на сторінку, з якої був редірект на сторінку реєстрації.

Сторінка навчальних курсів

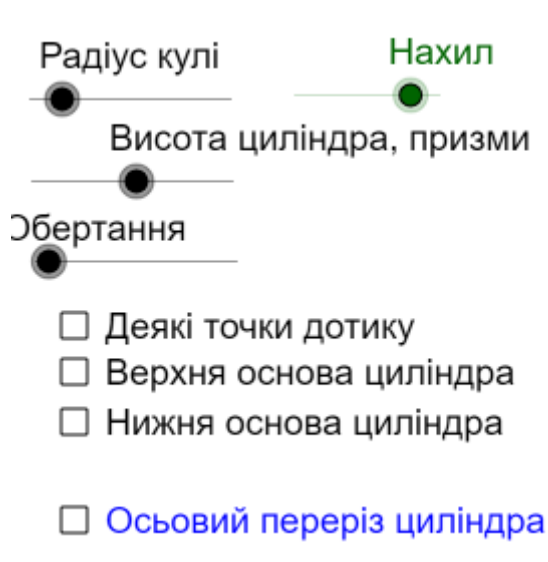
- Вибір предмету у вигляді табів;
- Есайд бар(навігація зліва від екрану), в якому будуть присутні акордеони, в яких назва – номер класу, контент – список тем за обраним класом;
- В головній частині екрану буде реалізована частина з викладанням обраної підтеми: пояснення всіх теорем, правил, аксіом та велика кількість прикладів. Також в кінці проходження підтеми буде присутнє невелике тестування, результат якого не буде впливати на можливість проходження наступної підтеми;
- В кінці проходження всієї теми буде доступне тестування, результат якого можна буде побачити на сторінці результатів;
- Можливість проходження тестування необмежена.

Сторінка проходження навчальних курсів з розділу ‘Стереометрія’

- Огляд об'ємної фігури в просторі. Присутнє графічне поле з об'ємною фігурою в просторі, в якому є можливості (Мал. 3.1.2.1):
 - Збільшення фігури;
 - Зменшення фігури;
 - Обертання фігури;
 - Рендж слайдер з градусом нахилу фігури (для вписаних фігур в сферу) (Мал. 3.1.2.2);
 - Рендж слайдер з радіусом сфери, в яку вписана об'ємна фігура (для вписаних фігур в сферу) (Мал. 3.1.2.2);
 - Рендж слайдер з висотою фігури (для вписаних фігур в сферу) (Мал. 3.1.2.2);
 - Рендж слайдер з обертанням фігури в середині сфера (для вписаних фігур в сферу) (Мал. 3.1.2.2);
 - Чекбокс з позначенням точок дотику вписаної фігури в сферу (Мал. 3.1.2.2);
 - Чекбокс з позначенням осьового перерізу фігури. (Мал. 3.1.2.2)



Мал. 3.1.2.1



Мал. 3.1.2.2

Сторінка результатів

- Присутній список пройдених користувачем тем та підтем з відповідними результатами;
- В кожному елементі списку буде можливість переглянути свої результати, натиснувши на кнопку перегляду;
- В кожному елементі списку буде можливість перескласти тест, якщо користувача не буде влаштовувати його поточний результат, натиснувши кнопку перескладання тесту підтеми або теми;
- В кожному елементі списку буде можливість перейти на сторінки підтеми обраного тесту, в якій зможе ще раз пройти матеріал та скласти тест, після чого результати оновляться та буде обраний найкращий результат.

3.2 Обґрунтування алгоритму й структури програми

Для створення програми мною був використаний стек PERN. Стэк PERN – це JavaScript-стек, котрий розроблений для поліпшення процесу розробки. Цей стек складається з PostgreSQL, Express, React і Node.js. Комбінуючи ці технології, можна створити повний стек веб-додаток з

операціями CRUD. Кожен, напевно, чув про MERN Stack, який по суті є стеком JavaScript для розгортання повного стека веб-розробки і містить 4 технології, а саме: MongoDB, Express, React і Node.js. але PostgreSQL, здається, набирає все більшої популярності, тому що він пропонує широку підтримку функцій NoSQL, транзакцій і стандарт.

Також основною перевагою використання цього стека є те, що кожна стрічка коду написана на мові програмування JavaScript і ця мова використовується як для коду на стороні сервера, так і для коду на стороні клієнта.

3.3 Обґрунтування вибору засобів розробки

3.3.1 React

React.js - це лише спосіб у зручному вигляді представити код JavaScript і HTML, зробити його повторюваним і наочним. Компоненти React.js пишуть особливою мовою - JSX, який виглядає як суміш JavaScript та HTML.

React.js має три потужні переваги:

- JSX набагато простіше читати та розуміти, ніж JavaScript, а значить, на підтримку та налагодження коду витрачається набагато менше часу;
- У бібліотеці зручно реалізована система компонентів - частин коду, що повторюються, які використовуються в різних умовах і обставинах і змінюються в залежності від контексту.
- Кожен компонент залежить тільки від свого стану - отже, набагато легше усувати помилки в коді, якщо щось не працює або працює не так, як очікувалося. Тобто помилки стають очевидними: компонент, який працює коректно сам собою, завжди працюватиме коректно, якщо йому не передати якийсь неправильний стан.

3.3.2 TypeScript

Оскільки TypeScript являється динамічною мовою програмування, а JavaScript – ні, контролювати та перевіряти всі параметри та змінні, які використовуються, набагато легше, і в той же час набагато складніше зробити помилку в коді, наприклад, забути оголосити змінну, випадково викликати неіснуючу функцію, або передати як параметр змінну, яка зламає весь код.

TypeScript дуже схожий на JavaScript, але він підтримує типи даних. Це спрощує читання коду та допомагає уникнути помилок.

Також однією з особливостей даної мови є підтримка останніх функцій JavaScript, але не всі сучасні браузерери можуть зрозуміти такий код, в якому використано нові функції, і, як правило, щоб зробити це можливим, потрібно включати додаткові інструменти, такі як Babel.

3.3.3 UIKit

Bootstrap описується як «Простий і гнучкий HTML, CSS і JS для популярних компонентів інтерфейсу користувача та взаємодії». Bootstrap – це найпопулярніший фреймворк HTML, CSS і JS для розробки адаптивних, мобільних проєктів у мережі. Але з іншого боку, UIKit детально описується як «Легка і модульна інтерфейсна платформа для розробки швидких і потужних веб-інтерфейсів». UIKit надає повну колекцію компонентів HTML, CSS і JS, яка проста у використанні, легко налаштовується та розширюється. UIKit надає такі ключові функції:

- LESS - UIKit розроблено на LESS для написання добре структурованого, розширюваного коду, який легко підтримувати.
- Компоненти – набір невеликих компонентів, що реагують, з використанням послідовних і безконфліктних умов імен.

- Customizer – базовий стиль UIKit можна розширити за допомогою тем і його легко налаштувати, щоб створити свій власний вигляд.

3.3.5 Express

На сервері повинна працювати програма, яка прийматиме і оброблятиме запити браузера і надсилатиме йому відповіді — будь-то створювані динамічно або просто у вигляді статичних відправлених файлів. Коли ми запитуємо сторінку, адже запит йде не в порожнечу – він йде на певну ір-адресу та порт, і цей порт має хтось слухати. Ось ця програма, що постійно слухає порт на залізному сервері, отримує запити та відсилає відповіді, і називається теж сервером. Один із різновидів сервера – це написаний за допомогою Express скрипт, його запускає Node.js і складається з інструкцій, що робити із запитами та як на них відповідати. Express просто набір готових засобів (бібліотека), за допомогою яких такий скрипт можна написати швидше, краще та надійніше, ніж з нуля, користуючись лише базовими модулями Node.js.

3.3.6 Three

Під час вибору бібліотеки для візуалізації, з'явилося головне питання: «Яку саме вибрати?». На перший погляд, вибрати відповідний інструмент нескладно: використовуємо `pixi.js` для 2d-графіків, а `three.js` – для 3d. Однак основна відмінність 2D від 3D - відсутність 3d-перспективи та фіксованого значення по третій координаті. Для того, щоб не було перспективи, можна використовувати ортографічну камеру. З цього випливає питання: «Що за камера?». Camera – це одне з ключових понять при реалізації графіки, поряд із `scene` та `renderer`.

Для наочності можна привести аналогію. Припустимо, що ви стоїте в кімнаті, тримаєте в руках смартфон і знімаєте відеоролик. Та кімната, де ви знімаєте відео – це `scene`. У кімнаті можуть бути різні предмети, наприклад

стіл та стільці – це об'єкти на scene. У ролі camera виступає камера смартфона, у ролі renderer – матриця смартфона, яка проектує 3d-кімнату на 2d-екран.

Основні переваги three.js над ріхі.js:

- У three.js більше прикладів реалізації тих чи інших завдань. До того ж, обравши цю бібліотеку, з'являється можливість працювати з 3d-графікою, а ось у разі ріхі.js такої переваги ця можливість відсутня.
- Інтерактивна взаємодія з елементами на сцені. Написати власну реалізацію досить трудомістко, але в обох бібліотеках вже є готові рішення: ріхі.js – з коробки, three.js – бібліотека three.interaction.

Після вибору бібліотеки для реалізації візуалізації можна зіштовхнутися з новою дилемою – використовувати react-обгортку або “канонічну” three.js. Для React реалізація обгортки – це react-three-fiber. На перший погляд, у ній досить мало документації, що може бути проблемою. Але насправді, на практиці все не так вже й складе, хоча при перенесенні коду three.js у react-three-fiber може виникнути низка питань щодо синтаксису.

3.3.7 PostgreSQL

PostgreSQL не просто реляційна, а об'єктно-реляційна СУБД, що дає змогу зберігати масиви значень для більшості існуючих типів даних. Це дає йому деякі переваги над іншими базами даних SQL з відкритим вихідним кодом, такими як MySQL, MariaDB і MongoDB. Також однією з переваг є підтримка JSON у PostgreSQL, котра дозволяє перейти до зберігання schema-less даних у SQL базі даних. Ця можливість є дуже корисною, коли структура даних потребує певної гнучкості: наприклад, якщо в процесі розробки

структура все ще змінюється або невідомо, які поля міститиме об'єкт даних.

3.3.8 SCSS

SCSS - "діалект" мови SASS. SASS – це мова схожа на шаблонізатор, але призначена для спрощення створення CSS-коду. Синтаксис цієї мови дуже гнучкий, він враховує безліч дрібниць, які так бажані в CSS.

Особливості в використанні цієї мови:

- Змінні
- Вкладеність
- Mixins
- Імпортування

3.4 Створення об'єктів і розробка головної програми

3.4.1 Початкові налаштування

Першим кроком в створенні веб-застосунку – це створення пустого каталогу для проекту. Це можна зробити звичайним способом

```
mkdir cursovaProject
```

```
cd cursovaProject
```

Після створення проекту потрібно його ініціалізувати

```
npm init -y
```

Ця команда створює файл `package.json` та ініціалізує його з передвстановленими значеннями. Якщо з'явилось бажання самому заповнити його поля, потрібно видалити прапор `-y` і слідувати інструкціям.

Встановлення пакетів у Node.js виконується командою `npm install packageName`, в даному випадку – `npm install express`.

Завдяки цій бібліотеці на сервері створюються маршрути, котрий являє собою кінцеву точку, до котрої можуть звертатися користувачі, та прослуховування портів. Маршрут асоціюється з HTTP-глаголом (наприклад, GET, POST і пр.) і отримує шлях URL. Крім того, він має функцію, яка викликається при зверненні до цієї кінцевої точки.(Мал. 3.4.1.1)

```
router.post("/register", userController.register);  
router.post("/login", userController.login);  
router.get("/logout", userController.logout);
```

Мал. 3.4.1.1

3.4.2 Реєстрація та авторизація

Через експрес-маршрути запит HTTP, який відповідає маршруту, буде перевіряється проміжним програмним забезпеченням CORS перед тим, як перейти до рівня безпеки.

Рівень безпеки включає:

- Проміжне програмне забезпечення аутентифікації JWT: перевірка реєстрації, перевірте токену;
- Проміжне програмне забезпечення авторизації: перевірка ролі користувача із записом у базі даних;

І якщо ці проміжні програми видають помилку, повідомлення буде надіслано як відповідь HTTP.

Контролери взаємодіють з базою даних PostgreSQL через Sequelize і надсилають відповідь HTTP (токен, інформацію про користувача, дані на основі ролей тощо) клієнту.

3.4.3 Створення проекту на React

Код налаштування авторизації (Мал. 3.4.3.1)

```

const [user, setUser] = useState({
  email: "",
  password: "",
});

const onChangeInput = (e) => {
  const { name, value } = e.target;
  setUser({ ...user, [name]: value });
};

const loginSubmit = async (e) => {
  e.preventDefault();
  try {
    await axios.post("/user/login", { ...user });

    localStorage.setItem("firstLogin", true);

    window.location.href = "/";
  } catch (err) {
    alert(err.response.data.msg);
  }
};

```

Мал. 3.4.4.1

Код налаштування реєстрації (Мал. 3.4.4.2)

```

const [user, setUser] = useState({
  name: "",
  email: "",
  password: "",
});

const onChangeInput = (e) => {
  const { name, value } = e.target;
  setUser({ ...user, [name]: value });
};

const registerSubmit = async (e) => {
  e.preventDefault();
  try {
    await axios.post("/user/register", { ...user });

    localStorage.setItem("firstLogin", true);

    window.location.href = "/";
  } catch (err) {
    alert(err.response.data.msg);
  }
};

```

3.4.4 Маршрутизація на клієнтській стороні

Код налаштування маршрутизації (Мал. 3.4.4.1)

```
function Pages() {  
  const state = useContext(GlobalState);  
  const [isLoggedIn] = state.userAPI.isLoggedIn;  
  
  return (  
    <Switch>  
      { /*Main Page */}  
      <Route path="/" exact component={Home} />  
  
      { /*Login-Registration Pages */}  
  
      <Route path="/login" exact component={isLoggedIn ? NotFound : Login} />  
      <Route  
        path="/register"  
        exact  
        component={isLoggedIn ? NotFound : Register}  
      />  
  
      <Route path="*" exact component={NotFound} />  
    </Switch>  
  );  
}
```

Мал 3.4.4.1

3.4.5 Відображення об'ємних фігур у просторі

Налаштування відображення моделі:

- Імпорт моделі (Мал. 3.4.5.1);
- Налаштування світла (Мал. 3.4.5.2);
- Налаштування OrbitControls (Мал. 3.4.5.3);
- Налаштування відображення в канвасі (Мал. 3.4.5.4);
- Налаштування початкового відображення моделі (Мал. 3.4.5.5);
- Імпортування компонента моделі на сторінку.

Висновки

В процесі роботи було створено прототип веб-застосунків навчальної системи для підготовки до ЗНО. Після процесу ретельного дослідження

існуючих аналогічних ресурсів, було вирішено розробити удосконалений ресурс.

Для виконання поставлених задач був використаний стек PERN та мови програмування: TS, HTML, CSS. Також використовувалися такі бібліотеки, як `express.js`, `three.js` тощо. Вибір цього стеку був зумовлений зручністю у використанні, так як в роботі були наведені багато переваг обраних ресурсів для розробки клієнт-серверного застосування.

Результатом роботи стало створення прототипу веб-застосунку навчальної системи, основною перевагою якого є зручне проходження відповідних курсів та курсів з розділу стереометрії, який може бути використаний для будь-якого тестування, як в навчальних закладах, так і в приватних цілях.

Список використаної літератури та електронних ресурсів

1. Документація бібліотеки React.js. [Електронний ресурс]. – Режим доступу:
https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started
2. Документація бібліотеки React.js. [Електронний ресурс]. – Режим доступу: <https://uk.reactjs.org/>
3. Навчальні матеріали для роботи з react-router. [Електронний ресурс]. – Режим доступу: <https://reacttraining.com/react-router/>
4. Навчальні матеріали для роботи з react-router. [Електронний ресурс]. – Режим доступу: https://reactdev.ru/libs/react-router/#_3
5. Навчальні матеріали для роботи з React Hooks. [Електронний ресурс]. – Режим доступу: <https://ru.reactjs.org/docs/hooks-overview.html>
6. Документація бібліотеки Three.js. [Електронний ресурс]. – Режим доступу: <https://threejs.org/docs/>
7. Документація платформи Node.js. [Електронний ресурс]. – Режим доступу: <https://nodejs.org/uk/docs/>
8. Документація платформи Node.js. [Електронний ресурс]. – Режим доступу: <https://metanit.com/web/nodejs/1.1.php>
9. PERN Stack. [Електронний ресурс]. – Режим доступу:
<https://www.geeksforgeeks.org/what-is-pern-stack/>
10. Документація платформи Тривимірна візуалізація результатів томографічного контролю. [Електронний ресурс]. – Режим доступу:
<https://moluch.ru/conf/tech/archive/123/6254/>
11. Курсова робота «Розробка веб-сайту для продажу дизайну інтер'єру та інтерактивного огляду продукту у 3D».

Додатки

```
function Model() {  
  const gltf = useGLTFLoader("/chair/cube.gltf", true);  
  return <primitive object={gltf.scene} dispose={null} />;  
}
```

(Мал. 3.4.5.1)

```
const Lights = () => {  
  return (  
    <>  
    <ambientLight intensity={0.3} />  
    <directionalLight position={[12, 10, 5]} intensity={1} />  
    <directionalLight  
      castShadow  
      position={[0, 15, 0]}  
      intensity={1.5}  
      shadow-mapSize-width={1024}  
      shadow-mapSize-height={1024}  
      shadow-camera-far={50}  
      shadow-camera-left={-20}  
      shadow-camera-right={20}  
      shadow-camera-top={20}  
      shadow-camera-bottom={-20}  
    />  
    <spotLight intensity={1} position={[1000, 0, 0]} castShadow />  
  </>  
);  
};
```

(Мал. 3.4.5.2)

```

const HTMLContent = () => {
  const ref = useRef();
  useFrame(() => (ref.current.rotation.y += 0.01));
  const {
    camera,
    gl: { domElement },
  } = useThree();
  return (
    <Section factor={1.5} offset={1}>
      <group position={[0, 250, 0]}>
        <mesh ref={ref} position={[0, -35, 0]}>
          <Model />
          <orbitControls args={[camera, domElement]} />
        </mesh>
      </group>
    </Section>
  );
};

```

(Мал. 3.4.5.3)

```

const Canvass = () => {
  return (
    <Canvas colorManagement camera={{ position: [0, 0, 120], fov: 70 }}>
      </* Lights Component */>
      <Lights />
      <Suspense fallback={null}>
        <HTMLContent />
      </Suspense>
    </Canvas>
  );
};

```

(Мал. 3.4.5.4)

```

import { createRef } from "react";

const state = {
  sections: 3,
  pages: 3,
  zoom: 1,
  top: createRef(),
};

```

(Мал. 3.4.5.5)