МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ Національний університет «Острозька академія» Навчально-науковий інститут інформаційних технологій та бізнесу Кафедра інформаційних технологій та аналітики даних

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

на тему: «Створення інтерактивної віртуальної карти Національного університету "Острозька академія" з використанням React, Three.js та технологій WebGL»

Виконала: студентка 4 курсу, групи КН-42 першого (бакалаврського) рівня вищої освіти спеціальності 122 Комп'ютерні науки освітньо-професійної програми «Комп'ютерні науки» Перепелиця Діана Олександрівна

Керівник: викладач кафедри інформаційних технологій та аналітики даних Місай Володимир Віталійович

Рецензент: ФОП в галузі інформаційних технологій, викладач кафедри менеджменту та маркетингу НаУОА Шпак Андрій Григорович

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ

Завідувач кафедри інформаційних технологій та аналітики даних проф., д.е.н. Кривицька О.Р. Протокол №10 від 28 травня 2025 р.

Острог, 2025

АНОТАЦІЯ кваліфікаційної роботи на здобуття освітнього ступеня бакалавра

Тема: Створення інтерактивної віртуальної карти Національного університету "Острозька академія" з використанням React, Three.js та технологій WebGL

Автор: Перепелиця Діана Олександрівна

Науковий керівник: викладач Місай Володимир Віталійович

Пояснювальна записка до кваліфікаційної роботи: 49с, 22 рис. 21 джерело

Ключові слова: карта, інтерактивна, React, Three.js, WebGL, 3D, віртуальна, інструменти, Blender, модель, створення.

Короткий зміст праці:

Завданням кваліфікаційної роботи/ проєкту, було створення інтерактивної віртуальної карти Національного університету "Острозька академія" з використанням React, Three.js та технологій WebGL.Основна робота полягала у створенні тривимірної моделі Національного університету "Острозька академія" за допомогою сучасних інструментів 3D-моделювання, а також інтеграції цієї моделі у веб-застосунок на базі React. Особлива увага була приділена реалізації інтерактивних функцій, які дозволяють користувачам зручно орієнтуватися на території університету, переглядати інформацію про аудиторії та інші об'єкти.

(підпис автора)

ANNOTATION of qualification paper for bachelor's degree

Theme: Creating an interactive virtual map of the National University "Ostroh Academy" using React, Three.js and WebGL technologies

Author: Diana Perepelytsia

Scientific supervisor: Misai V.V., lecturer at the ITAD department.

Defended «......»...... of 2025.

Explanatory note to the qualification work: 49 p., 22 pic., 21 sources.

Keywords: map, interactive, React, Three.js, WebGL, 3D, virtual, tools, Blender, model, creation

Summary of the paper:

The task of the qualification work/project was to create an interactive virtual map of the National University "Ostroh Academy" using React, Three.js and WebGL technologies. The main work consisted in creating a three-dimensional model of the National University "Ostroh Academy" using modern 3D modeling tools, as well as integrating this model into a web application based on React. Particular attention was paid to the implementation of interactive functions that allow users to conveniently navigate the university territory, view information about classrooms and other objects.

Зміст

ВСТУП	4
РОЗДІЛ І. ТЕОРЕТИЧНІ ЧАСТИНА СТВОРЕННЯ ІНТЕРАКТИВНОЇ КА	РТИ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ "ОСТРОЗЬКА АКАДЕМІЯ"	6
1.1.Постановка проблеми створення інтерактивної карти Національного університету "Острозька академія"	6
1.2.Аналіз та вибір інструментів створення віртуальної карти	7
1.3. Аналіз аналогічних розробок	12
Висновки до розділу 1	15
РОЗДІЛ ІІ. ПРИКЛАДНА РОЗРОБКА ВІРТУАЛЬНОЇ КАРТИ	
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ "ОСТРОЗЬКА АКАДЕМІЯ"	17
2.1. Збір основної інформації для виконання завдання	17
2.2. Процес створення 3D-моделей основного корпусу університету в середо Blender.)вищі 20
2.3. Створення базового середовища для проекту з використанням React	24
2.4. Інтеграція Three.js: створення сцени та додавання 3D-моделей	26
2.5 Реалізація функціональності інтерактивної карти	30
Висновок до розділу 2	44
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	48

ВСТУП

В наш час інформаційні технології все більше розвиваються та стають невід'ємною частиною нашого життя, зокрема в освітній сфері. Доступ до інформації та ефективна навігація навчальним середовищем є основою успішного навчання. Одним із перспективних напрямків є створення віртуальних карт, які можуть бути використані для візуалізації територій, планів будівель або інфраструктури. Дану навігацію було розроблено з використанням різних технологій. Завдяки такій сучасній навігації можна отримати додаткову інформацію про об'єкти, які нас оточують. Інтерактивна карта стане важливою складовою інформаційного середовища університету, створюючи комфортні умови для навчання та розвитку.

Мета дослідження полягає у розробці віртуальної карти Національного університету "Острозька академія" з використанням сучасних веб-технологій, таких як React, Three.js та WebGL, для полегшення навігації для перших курсів.

Об'єктом дослідження є віртуальна карта Національного університету "Острозька академія".

Предметом дослідження є вивчення всіх теоретичних та технічних аспектів створення інтерактивної віртуальної карти для полегшення навігації та отримання додаткової інформації про об'єкти в університеті.

Завданнями кваліфікаційної роботи є проведення аналізу існуючих віртуальних карт з метою вивчення їх функціональних можливостей та принципів реалізації, а також аналіз сучасних технологій, які застосовуються для створення інтерактивних карт. У межах проєкту передбачено вибір оптимальних інструментів для розробки на основі порівняльного аналізу, зокрема дослідження функціоналу таких технологій, як React, Three.js та WebGL, що дозволяють реалізовувати тривимірні об'єкти. Наступним етапом є створення прототипу віртуальної карти університету з урахуванням його території, включаючи основні об'єкти та структуру забудови. Після побудови базової сцени передбачено наповнення карти текстовими, графічними та мультимедійними матеріалами, а також налаштування інтерактивної взаємодії користувачів із елементами карти. Завершальним етапом є проведення тестування та оптимізації розробленої системи для забезпечення її стабільної роботи та зручності використання

Актуальність даного проєкту полягає у впровадженні та створенні нових інформаційних технологій у сферу освіти, що важливим етапом модернізації навчання. Створення віртуальної карти полегшить процес орієнтування для студентів та гостей. Це особливо актуально у сучасному світі, традиційні паперові карти вже практично не використовуються. Інтерактивна карта це дуже зручно, сучасно, гарно та актуально.

РОЗДІЛ І. ТЕОРЕТИЧНІ ЧАСТИНА СТВОРЕННЯ ІНТЕРАКТИВНОЇ КАРТИ НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ "ОСТРОЗЬКА АКАДЕМІЯ"

1.1.Постановка проблеми створення інтерактивної карти Національного університету "Острозька академія"

Національний університет "Острозька академія" є одним із провідних освітніх закладів України, відомим своєю історією та унікальною архітектурою. Основний корпус академії має складне планування. Для студентів першого курсу, нових співробітників та відвідувачів орієнтація в такому приміщенні може бути проблематично. Зокрема, труднощі виникають при пошуку потрібних аудиторій, деканатів або ж інших важливих приміщень. Це все може створювати незручності для студентів та втрату часу.

Звичайні підходи до вирішення цієї проблеми, наприклад, друковані карти або схеми на стінах, мають багато недоліків. Вони мають обмежену функціональність, не мають інтерактивності, незрозумілі, складно виглядають.Традиційні схеми не забезпечують можливості швидкого пошуку за ключовими словами (наприклад, назвою аудиторії чи факультету), не дозволяють отримувати додаткову інформацію про об'єкти. Вони не інтегруються з мобільними пристроями, що обмежує зручність використання в сучасному середовищі. У випадках великого потоку нових студентів або під час заходів, подібні карти виявляються неефективними, оскільки не дозволяють швидко орієнтуватися великій кількості людей одночасно.Також вони вже давно не актуальні, переважно ними вже не користуються.

Одним із вагомих недоліків традиційних карт є складність їх оновлення. У разі зміни назв аудиторій, переміщення кафедр або реконструкції приміщень — друковані карти швидко втрачають актуальність. Віртуальна карта, натомість, дозволяє оперативно оновлювати інформацію без потреби у додаткових витратах на друк чи розповсюдження.

З огляду на вищезазначені проблеми, актуальним стає створення інтерактивної віртуальної карти університету, з використанням сучасних технологій візуалізації.

Віртуальна карта Острозької академії буде корисним інструментом для студентів, викладачів та гостей університету. Її використання дозволить:

-легко знаходити потрібні аудиторії, кафедри та інші об'єкти інфраструктури;

-отримувати додаткову інформацію про об'єкти, розташовані на території університету;

-ознайомитися з структурою Острозької академії;

-зробити процес навігації по університету більш зручним та комфортним.

Найбільша перевага інтерактивних карт полягає в їхній інтерактивності. Користувачі можуть взаємодіяти з картами, отримувати вказівки та навіть ділитися інформацією з друзями і колегами. Це відкриває нові можливості для спільної роботи та співпраці в реальному часі, що робить інтерактивні карти незамінним інструментом для різних галузей, включаючи освіту, туризм, бізнес та багато інших.

Підсумовуючи можна сказати, що проблема створення інтерактивної карти Острозької академії полягає у необхідності надання студентам зручного інструменту для орієнтації на території університету. Інтерактивна 3D-карта є ідеальним рішенням, яке відповідає потребам університету, покращуючи навігацію та підвищуючи рівень комфорту для студентів і гостей навчального закладу.

1.2.Аналіз та вибір інструментів створення віртуальної карти

Для створення інтерактивної віртуальної карти Національного університету "Острозька академія" було проведено аналіз різних технологій.

Також було використано GitHub для контролю версій.Завдяки системі керування версіями Git стало можливим зберігати історію змін у коді, повертатися до попередніх версій у разі помилок, а також ефективно працювати над окремими модулями незалежно один від одного.

React – це популярна бібліотека JavaScript, яка дозволяє створювати динамічні високою веб-застосунки 3 продуктивністю. React використовує концепцію віртуального DOM (Document Object Model), який дозволяє відстежувати зміни у стані додатка і оновлювати тільки ті елементи інтерфейсу, які потребують змін [1]. Це значно знижує навантаження на браузер і забезпечує швидкий відгук додатку. Для інтерактивної карти, де змінюється багато графічних елементів, така оптимізація є важливою. Також React має компонентний підхід, тобто організовує код у вигляді окремих компонентів. ЩО можуть повторно використовуватися та легко змінюватись. У контексті розробки віртуальної карти це дозволяє легко створювати та керувати різними частинами інтерфейсу. [2]

Також React має потужні інструменти для роботи зі станом додатка. Це зручно для інтерактивних елементів карти, які повинні змінюватися залежно від взаємодії користувача (наприклад, натискання на об'єкт або переміщення по карті). React добре інтегрується з кількома популярними бібліотеками для 3D-візуалізації, які використовуються у веб-розробці.

Vue.js - це сучасний JavaScript-фреймворк для створення інтерфейсів користувача.[3]

Він має простий інтерфейс та високу продуктивність. Vue.js автоматично відстежує зміни у стані додатка і оновлює лише ті частини інтерфейсу, які цього потребують. Vue.js більше орієнтований на побудову інтерфейсів, аніж на створення універсальних додатків.

Переваги Vue.js:

- Vue.js використовує реактивну модель даних, яка автоматично синхронізує зміну стану з DOM. Це означає, що коли змінюється стан або дані, інтерфейс оновлюється автоматично, без потреби вручну маніпулювати елементами сторінки.

- Vue має інтуїтивно зрозумілий синтаксис, що робить його популярним серед новачків і досвідчених розробників.

Можна використовувати як для невеликих проєктів, так і для великих систем. Його легко інтегрувати з іншими бібліотеками або проєктами на JavaScript.

Недоліки Vue.js:

-Менша поширеність у великих корпоративних системах, ніж у React

чи Angular.

-Vue має менше сторонніх бібліотек та інструментів.[4]

Babylon.js - це бібліотека JavaScript, яка використовується для створення 3D-графіки.За допомогою цієї бібліотеки можна створювати складні анімацій з інструментами для управління.Також має великий набір інструментів.Ця бібліотека є складна для початківців.У порівнянні з іншими бібліотеками Babylon.js є важчою.[5]

Переваги Babylon.js:

-Великий набір інструментів для 3D-графіки.

-Має інтерфейси для роботи з WebGL.

-Інтерактивність та анімації (є вбудовані інструменти для створення складних анімацій, управління камерами та об'єктами в реальному часі.

Недоліки Babylon.js:

-Складна у використанні.

-Менша популярність у порівнянні з іншими бібліотеками.

Three.js – це бібліотека, яка значно спрощує роботу з WebGL та дозволяє легко працювати з 3d об'єктами у веб-браузерах. Він ідеально підходить для реалізації віртуальної карти. Three.js дозволяє працювати з зовнішніми форматами, такими як GeoJSON, OBJ, FBX, GLTF/GLB, або навіть власними.[6] Можна додати інтерактивні елементи, такі як клікабельні маркери, рух об'єктів, зміна виду камери, інтерактивне масштабування та панорамування.Також Three.js оптимізовує рендеринг 3D-сцени через WebGL, використовуючи апаратне прискорення (GPU). Це забезпечує швидку обробку навіть для складних карт з багатьма об'єктами.[7]

Переваги Three.js:

-Широкий набір функціоналу

-Можна імпортувати моделі з Blender чи інших редакторів у форматах GLTF/GLB.

Використовує WebGL і апаратне прискорення, дозволяючи створювати складні сцени з великою кількістю об'єктів.

Недоліки Three.js:

-Обмеження на продуктивність у слабких пристроях

-Немає вбудованих елементів інтерфейсу, як у React/Vue. Інтеграцію потрібно робити окремо.

WebGL є основою для відображення 3D-графіки у веб-браузерах. За допомогою Three.js ми можемо повною мірою скористатися можливостями WebGL без необхідності працювати з його складним API.[8]

Завдяки WebGL, можна реалізувати:

-Реалістичну 3D-графіку з тінями, відблисками, освітленням.

-Інтерактивність у реальному часі — можливість динамічної зміни сцени без перезавантаження сторінки.

-Високу продуктивність за рахунок використання GPU.[9]

Для створення інтерактивної віртуальної карти Національного університету "Острозька академія" було обрано такі технологій, як **React**, **Three.js** та **WebGL**.

Використання технологій **React**, **Three.js** та **WebGL** дозволяє реалізації таких функцій, як:

-Інтерактивна навігація: користувачі можуть переглядати аудиторії з різних ракурсів, масштабувати карту.

-Інформаційні підказки: коли користувач натискає на об'єкти відкриватиметься інформація про аудиторії, деканати чи інші приміщення.

-Сучасний дизайн: завдяки 3D-візуалізація дана карта матиме дуже стильний та сучасний вигляд.[10]

Для створення віртуальної карти було розглянуто декілька варіантів програм:

Blender -програмний пакет для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації.[11]

Переваги:

-Повністю безкоштовний: Blender повністю безкоштовний для використання, без ліцензійних обмежень.

-Відкритість коду: завдяки відкритому коду, спільнота може доповнювати його новими функціями та інструментами.

-Підтримка сучасних форматів експорту:Blender дозволяє експортувати моделі у формати, які легко інтегруються в Three.js, в даному випадку glTF.

-Доступний інтерфейс.

-Різноманітний функціонал(моделювання, текстурування, анімації, скульптинг).

-Рендер високої якості.

-Підтримка різних операційних систем: Blender підтримує Windows,macOS і Linux, що дозволяє користувачам вибирати операційну систему, яка їм найбільше підходить.

Недоліки:

-Може бути трохи складний для початківців: Blender може здатися складним через велику кількість інструментів, комбінацій клавіш та складну логіку роботи з деякими елементами.

3ds Max - тривимірний графічний редактор, повнофункціональний професійний застосунок, система для створення і редагування об'єктів та створення візуалізацій

Переваги:

-Хороший інструмент для створення високоякісних 3D-моделей, які можуть використовуватись у професійних проектах.

-Підтримує багато форматів(OBJ, FBX, STL, glTF, 3DS), що забезпечує зручність при інтеграції з іншими програмними засобами або бібліотеками, такими як Three.js, Unity, Unreal Engine.

-Багато функцій для моделювання.

-Інтеграція з іншими програмами: Cinema 4D добре інтегрується з Adobe After Effects, що дає змогу швидко переносити 3D-сцени у відеомонтаж без втрати якості.

Недоліки:

-Висока вартість ліцензії.

-Складний для початківців через великий набір функцій.[12]

Cinema 4D - програмний пакет для створення тривимірної графіки та анімації **Переваги:**

-Зрозумілий інтерфейс, який легко засвоюється.

-Стабільність роботи, рідко дає збої навіть при роботі з великими сценами, що є важливою перевагою у професійній діяльності.

-Функціонал для моделювання та анімації.

Недоліки:

-Висока вартість.

-Обмежені можливості для створення складних моделей у порівнянні з Blender чи 3ds Max.[13]

Тому серед розглянутих варіантів інструментів для створення 3D-моделей було обрано **Blender**, оскільки він є безкоштовним, універсальним і має всі необхідні функції для виконання завдань кваліфікаційної роботи.

1.3. Аналіз аналогічних розробок

Перед розробкою інтерактивної карти для Острозької академії було важливо провести аналіз конкурентів, щоб краще зрозуміти сучасні тенденції та підходи до створення подібних продуктів.

Інтерактивних карт в інших навчальних закладах України не було знайдено. Проте, такі карти роблять великі торговельні центри та закордонні відомі університети. Це дозволяє покращити відвідуваність та зручність для клієнтів. Інтерактивні карти надають можливість відвідувачам швидко та легко знаходити бажані магазини, ресторани, туалети та інші службові приміщення.

Крім того, інтерактивні карти допомагають оптимізувати внутрішню логістику торговельного центру. Вони можуть стати потужним інструментом для управління

розташуванням товарів та роботою персоналу. Також впровадження інтерактивних карт може підвищити імідж торговельного центру, підкреслити його сучасність та інноваційність, що стає додатковим фактором конкурентоспроможності. Ось наприклад, мапа ТРЦ «Оазис», див.рис.1.1.



Рис.1.1 Мапа ТРЦ «Оазис»

Джерело: [14]

Карта охоплює три поверхи ТРЦ: 0, 1 та 2. Кожен поверх представлений у вигляді схеми з позначенням магазинів, сервісів та інфраструктурних елементів. Користувач може перемикатися між поверхами для перегляду відповідної інформації.

Карта дозволяє швидко знайти розташування магазинів, кафе, кінотеатру, фітнес-центру та інших об'єктів.Також при наведенні або натисканні на об'єкти відображається додаткова інформація, така як назва магазину чи послуги. Проте,деякі об'єкти можуть бути представлені без детальної інформації.Одним із недоліків є те, що немає функції пошуку за назвою магазину чи категорією,це може ускладнити навігацію для нових відвідувачів. Також для порівняння було знайдено ще одну інтерактивну карту торговельного центру «ЗлатаПлаза», див.рис.1.2.



Рис.1.2 Карта «Злата Плаза»

Джерело:[15]

Карта має просту інтерактивність, відображено всі поверхи ТРЦ із зазначенням місць розташування магазинів і зон відпочинку з інформацією про них. Доступна опція пошуку за категоріями. При наведенні або натисканні на об'єкти відображається додаткова інформація, така як назва магазину чи послуги. Також на карті є фільтрація за категоріями.

Також було знайдено ось таку мапу, Бостонського університету, див.рис.1.3.



Рис.1.3 Мапа Бостонського університету

Джерело: [16]

Дана карта добре зроблена, дуже інтерактивна, містить багато фото, інформації, але вона тільки по корпусах. Тобто на даній карті ми не зможемо знайти аудиторії. Також є пошук у реальному часі за назвою чи категорією, це досить хороша функція. Мапа інтерактивна, дозволяє наближати й віддаляти зображення, а також знаходити конкретні будівлі, як-от академічні корпуси, гуртожитки, спортивні споруди та кафе. Має високий рівень деталізації.

Ці мапи демонструють, як інтерактивні функції можуть полегшити навігацію, а також визначають, які елементи варто врахувати при розробці власного продукту.

Висновки до розділу 1

У цьому розділі було описано постановку проблеми створення віртуальної карти Національного університету "Острозька академія". Також було проведено аналіз та вибір інструментів для створення карти.

Том,підсумовуючи можна сказати, що проблема створення інтерактивної карти Острозької академії полягає у необхідності надання студентам зручного інструменту для орієнтації на території університету.Інтерактивна 3D-карта є сучасним рішенням, яке відповідає потребам університету.

Також було обрано такі технологій, як **React**, **Three.js** та **WebGL** для створення інтерактивної віртуальної карти.Серед розглянутих варіантів інструментів для створення 3D-моделей було обрано **Blender**, оскільки він є безкоштовним, універсальним і має всі необхідні функції для виконання завдань кваліфікаційної роботи. Він підтримує формат glTF, який легко інтегрується з **Three.js**, це є додатковою перевагою. Також дуже важливо що Blender є безкоштовною програмою, це дуже вагома перевага. Blender сучасний, доступний, багатофункціональний, повністю відповідає вимогам для створення віртуальної карти.

Проаналізувавши аналогічні розробки, можна сказати ,що впровадження подібної карти для університету буде сучасно, актуально, саме головне корисно для студентів та відвідувачів, підвищить імідж навчального закладу.

РОЗДІЛ ІІ. ПРИКЛАДНА РОЗРОБКА ВІРТУАЛЬНОЇ КАРТИ НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ "ОСТРОЗЬКА АКАДЕМІЯ"

2.1. Збір основної інформації для виконання завдання

На початковому етапі необхідно було найти та зібрати дані, які нам необхідні для нашої роботи. Тому перш за все було отримано технічний паспорт першого та другого поверхів головного корпусу академії.Це потрібно для того щоб орієнтуватися, які приблизно розміри треба, щоб на основі них побудувати 3D модель. На рис. 2.1, показано приклад технічного паспорту.



Рис.2.1 Технічний паспорт 1 та 2 поверху

Джерело: Національний університет «Острозька академія»

Також було взято план поверхів,для максимально схожого відтворення основного корпусу. На основі цих даних ми змогли визначити масштаб і основні пропорції майбутньої 3D-моделі (див.рис.2.2-2.3).



Рис.2.2. План першого поверху

Джерело: Національний університет «Острозька академія»



Рис.2.3. План другого поверху

Джерело: Національний університет «Острозька академія»

Також було взято дозвіл на фотофіксацію аудиторій. Фотографії покращать користувацький досвід, зробивши навігацію по віртуальній карті більш інформативною та наочною. Це рішення також сприяє більш ефективному представленню університетської інфраструктури для абітурієнтів та відвідувачів (див.рис 2.4).



Рис 2.4.Фотознімки основного корпусу *Джерело:* розроблено автором

2.2. Процес створення 3D-моделей основного корпусу університету в середовищі Blender.

1. Створення основи моделі

На першому етапі в 3D-сцені був створений плоский об'єкт типу **Plane**, який слугував базовою площиною для побудови контурів приміщення.

Plane — це базовий примітив, який представляє собою двовимірну поверхню та використовується як початкова форма для побудови більш складних об'єктів у 3D-сцені.

2. Формування контурів стін

Перейшовши в Edit Mode, було використано інструменти для роботи з вершинами, ребрами та гранями. На площині були вручну відтворені контури внутрішніх і зовнішніх стін згідно з кресленням або планом будівлі. Контури утворювали замкнуті полігони, які відповідали товщині стін.

Edit Mode — це режим редагування, у якому користувач може змінювати геометрію об'єкта: додавати, переміщувати або видаляти вершини (vertices), ребра (edges) та грані (faces).

Отримані контури стін були екструдовані вгору за допомогою інструмента **Extrude,** це дозволило створити тривимірні стіни з потрібною висотою (див.рис.2.5).



Рис.2.5 Формування контурів стін 3d моделі *Джерело:* розроблено автором

3. Створення отворів для дверей і вікон

Для цього було використано операції **Boolean.**Тобто були створені окремі об'єкти, такі як прямокутники, які підходили для розміру вікон і дверей.Ці об'єкти розміщувалися на певних ділянках стін,згідно плану.

Використовуючи модифікатор **Boolean** із режимом **Differenc**e, отвори для вікон і дверей були вирізані у стінах.

Boolean — це модифікатор у Blender, який дозволяє виконувати логічні операції над двома об'єктами в 3D-просторі. Він використовується для поєднання або вирізання геометрії одного об'єкта за допомогою іншого.

Difference (Віднімання) — вирізає частину з одного об'єкта за формою іншого (див.рис.2.6).



Рис.2.6 Створення отворів для дверей і вікон Джерело: розроблено автором

Також необхідно було зробити освітлення. У сцені було розміщено джерела cвітла Light, які виглядали як звичайне освітлення. За допомогою цього ми могли оцінити вигляд моделі в реальних умовах освітлення.

У Blender Light — це об'єкт, який імітує джерело світла в сцені. Світло необхідне для того, щоб зробити об'єкти видимими та реалістично освітленими при рендерингу або перегляді сцени в режимі перегляду матеріалів (Material Preview) чи рендерингу (Render View).

Для зменшення навантаження на систему та покращення продуктивності було застосовано процес запікання **Bake**.

Важливість процесу Bake:

-Зменшує навантаження на комп'ютер при рендері.

-Дозволяє швидше переглядати сцену.

-Робить можливим експортувати сцену до інших рушіїв або в браузер (наприклад, у Three.js), де складні ефекти не можуть обраховуватися в реальному часі.

Після завершення моделювання першого поверху, усі основні операції були застосовані також до другого поверху. Зокрема, повторно було використано плоску поверхню як основу, виконано побудову контурів стін, екструдовано їх у висоту, створено прорізи для вікон і дверей за допомогою Boolean-модифікатора. Обидва поверхи будівлі були змодельовані відповідно до плану приміщення, що дозволило забезпечити цілісну та реалістичну 3D-модель споруди.

На фінальному етапі модель 1 і 2 поверху були збережена у форматі .glb для подальшої взаємодії з технологіями WebGL.3D- модель, точно відтворює архітектуру основного корпусу університету та є готова надалі до використання, (див. рис.2.7-2.8).



Рис.2.7 Вигляд 3D-моделі 1 поверху

Джерело: розроблено автором



Рис.2.8 Вигляд 3D-моделі 2 поверху Джерело: розроблено автором

2.3. Створення базового середовища для проекту з використанням React

Для реалізації інтерфейсу віртуальної карти було обрано JavaScript-бібліотеку React, яка є однією з найпопулярніших технологій для розробки сучасних вебзастосунків.

React побудований на компонентному підході, що дозволяє розділяти інтерфейс на незалежні, повторно використовувані частини (компоненти). Це значно спрощує розробку, тестування та масштабування проєкту. Кожен компонент може відповідати за свою частину інтерфейсу (наприклад, меню, панель інформації, карта, кнопки взаємодії), що забезпечує гнучкість та підтримуваність коду.

Однією з ключових переваг React є висока продуктивність, яку забезпечує використання віртуального DOM (Virtual DOM). Замість повного оновлення сторінки React оновлює лише ті елементи, які змінилися, що робить взаємодію з користувачем швидкою та плавною.

Крім того, React має велику екосистему бібліотек і інструментів (Redux або Context API для управління станом, інтеграцію з Three.js для роботи з 3D-сценами), що дозволяє розширювати функціональність застосунку без значних зусиль.

Використання React є доцільним вибором для створення динамічного, масштабованого і зручного у користуванні інтерфейсу віртуальної карти, що адаптується до потреб користувача та підтримує інтерактивну взаємодію з 3D-моделлю.[17]

Проєкт було створено в сучасному середовищі розробки Visual Studio Code, яке забезпечує зручний інтерфейс, підсвічування синтаксису, автодоповнення, інтегрований термінал та підтримку плагінів, що значно спрощує розробку фронтенд-застосунків. Для початку було встановлено Node.js, який є необхідним для запуску React-застосунку, оскільки він надає середовище виконання JavaScript на стороні сервера. Разом із Node.js було встановлено **npm (Node Package Manager)** — менеджер пакетів, який використовується для встановлення бібліотек, необхідних для роботи застосунку.В даному проекті ми використовуємо команду **npm** для створення проектів та управління залежностями.

Перш за все використовуємо команду **create-react-аpp**, для налаштування проекту на базі React.Після виконання цієї команди вже можемо побачити базову структуру React-додатку.Також потім необхідно скористатися командою **npm start**, щоб перевірити чи правильно ми все налаштували.Дана команда запускає застосунок у браузері на локальному сервері (зазвичай http://localhost:3000), що дозволяє миттєво переглядати зміни під час розробки. Цей сервер також підтримує гаряче оновлення (Hot Reloading), завдяки чому будь-які зміни у коді автоматично оновлюються без необхідності перезавантажувати сторінку вручну.

Папка src містить основний код додатку. Також було створено папку components, яка використовується для зберігання окремих React-компонентів, таких як Sidebar.js, Footer.js, Scene.js та інших.

2.4. Інтеграція Three.js: створення сцени та додавання 3D-моделей

Для візуалізації 3D-моделі та створення інтерактивної сцени було інтегровано бібліотеку **Three.js**. Це дозволило створити реалістичну візуалізацію елементів карти.

Завдяки Three.js стало можливим додавати різноманітні 3D-об'єкти, налаштовувати освітлення, камери, матеріали та анімації, що значно підвищує якість і привабливість візуалізації.

Крім того, бібліотека підтримує завантаження зовнішніх 3D-моделей у популярних форматах (glTF, OBJ, FBX), що забезпечує гнучкість у роботі з різними типами контенту. За допомогою Three.js також реалізовані інтерактивні функції, такі як клікабельні зони, масштабування сцени, обертання камери та відображення додаткової інформації при взаємодії користувача з елементами карти.

Перш за все потрібно було додати бібліотека Three.js до проекту. Це було зроблено за допомогою команди **npm install three.**

Наступним кроком було створення сцени.Перед цим необхідно встановити ще бібліотеки за допомогою команд **npm install @react-three/fiber @react-three/drei**.

npm install @react-three/fiber — встановлює основну React-обгортку для бібліотеки Three.js, що дозволяє створювати 3D-сцени в React з використанням JSX.[18]

npm install @react-three/drei — встановлює набір готових компонентів і утиліт, які розширюють можливості React Three Fiber і спрощують роботу з 3D.[19]

У компоненті Scene.js було створено базову сцену. Там ми використовуємо Canvas, це дуже важливий компонент, за допомогою нього ми можемо створити сцену WebGL для 3D-рендерингу.Також використовуємо OrbitControls, він є допоміжним компонентом із бібліотеки drei, яка розширює можливості React Three Fiber.За допомогою OrbitControls користувач може обертати, наближати або віддаляти камеру навколо сцени за допомогою миші.[20]

З бібліотеки drei, було використано хук useGLTF, він спрощує завантаження 3D-моделі академії у форматі GLTF/GLB.

use GLTF — це спеціальний React-хук, який значно спрощує процес завантаження і використання 3D-моделей у форматі GLTF або GLB в проєкті на базі React Three Fiber. Він автоматично завантажує модель, кешує її для повторного використання і забезпечує оптимальну продуктивність. Завдяки цьому хук дозволяє легко імпортувати складні 3D-моделі з текстурами, матеріалами та анімаціями. Це пришвидшує розробку і робить код більш чистим і зручним для підтримки.[21]

Спочатку було завантажено основу сцени, тобто наша 3D-модель, за допомогою **useGLTF.** Використовуємо компонент <primitive> для рендерингу GLTF-сцени в WebGL.

Лістинг 2.1 Додавання сцени першого поверху.

```
const FloorModel1 = () => {
  const { scene, error, isLoading } =
  useGLTF("../models/floor1.glb");
  if (error) {
    console.error("Error loading floor1 model:", error);
  }
  if (isLoading) {
    return <div>3aBaHTaжeHHЯ Першого поверху...</div>;
  }
  console.log("First floor model loaded", scene);
  return <primitive object={scene} scale={0.05} />;
 };
```

Також використовуємо функцію setActiveRoom, для зміни активної кімнати. Додаємо колір заднього фону та загальне освітлення, що є дуже важливим.

Створено компонент FloorModel2 для завантаження моделі другого поверху з файлу floor2.glb.

Лістинг 2.2 Завантаження другого поверху.

```
const FloorModel2 = () => {
  const { scene, error, isLoading } =
  useGLTF("../models/floor2.glb");
  if (error) {
    console.error("Error loading floor2 model:", error);
  }
  if (isLoading) {
    return <div>Завантаження другого поверху...</div>;
  }
  console.log("Second floor model loaded", scene);
  return <primitive object={scene} scale={0.05} />;
};
```

У цьому коді використовується хук useGLTF з бібліотеки @react-three/drei для завантаження 3D-моделі другого поверху у форматі GLTF/GLB. Хук повертає три основні значення:

- scene — об'єкт сцени Three.js, що містить завантажену 3D-модель. Цей об'єкт використовується для відображення моделі в нашій сцені та дозволяє маніпулювати її положенням, масштабом та іншими параметрами.

- error — змінна, яка містить інформацію про помилку, якщо під час завантаження виникла проблема. Це дає можливість виявити помилки, наприклад, якщо файл не знайдено або він пошкоджений, і вивести відповідне повідомлення для користувача або розробника.

- isLoading — логічний прапорець, який інформує про те, що модель ще завантажується. Його можна використати для відображення індикатора завантаження або повідомлення, що покращує користувацький досвід, показуючи, що процес триває.

Завдяки даному підходу, можемо ефективно контролювати процес завантаження 3D-моделі, обробляти можливі помилки та відображати модель після успішного завантаження.

Використано useState для збереження поточного поверху.

Лістинг 2.3 Використання useState.

```
const [currentFloor, setCurrentFloor] = useState(activeFloor);
```

Так як в нас вже 2 поверхи, то було зроблено вибір поверху для відображення. У компоненті Canvas перевіряється значення **currentFloor.**Це дозволяє перемикатися між моделями першого та другого поверхів залежно від обраного поверху.

Лістинг.2.4 Вибір поверху.

```
{currentFloor === 1 && <FloorModel1 />}
{currentFloor === 2 && <FloorModel2 />}
```

Також необхідно було реалізувати перемикання між поверхами. У коді це реалізовано за допомогою пропса **activeFloor**, який передається в компонент Scene. В залежності від значення activeFloor вибирається модель поверху (FloorModel1 або FloorModel2) і відображаються відповідні інтерактивні зони.

На карті ці перемикачі мають такий вигляд, (див. рис.2.9).



Рис.2.9.Перемикачі для поверхів

Джерело: розроблено автором

При наведенні, в залежності від активного поверху змінюють колір.

Потім необхідно додати 3D-моделі аудиторій, які будуть інтерактивними зонами..Для цього ми додаємо моделі аудиторій, які були експортовані у форматі glb.Bukopuctoвуємо **onClick** для виклику handleZoneClick, щоб встановити аудиторію як активну.Якщо зона є активною, то вона підсвічується.Також за

допомогою властивості position визначаємо розташування на сцені у 3D-просторі.Коли аудиторія буде активна вона буде підсвічуватись темно-синім кольором.

Лістинг 2.5 Додавання 3D-моделі аудиторії.

```
<InteractiveZone modelPath="../models/a1.glb" position={[0, 0, 0]}
color="#111a68" onClick={() => handleZoneClick("a1")}
isActive={activeRoom === "a1"} />
```

Наступним етапом було додавання усіх інтерактивні зон для другого поверху.

Лістинг 2.6 Приклад додавання інтерактивної зони для 2 поверху.

```
<InteractiveZone
	modelPath="../models/a6.glb"
	position={[0, 0, 0]}
	color="#111a68"
	onClick={() => setActiveRoom("a6")}
		isActive={activeRoom === "a6"}
	/>
```

Також це все було виконано і для решти аудиторій.

В загальному, цей код створює інтерактивний 3D-об'єкт на сцені, задає його розташування, зовнішній вигляд і визначає, що відбувається при кліку на нього. Він також контролює, чи є цей об'єкт активним, щоб змінити його вигляд або поведінку. Це дозволяє користувачеві взаємодіяти з картою або моделлю, вибираючи певну зону чи кімнату.

2.5 Реалізація функціональності інтерактивної карти

Далі було додано функціонал для інтерактивних зон.Першим компонентом в нас є InteractiveZone, який містить шлях до 3D-моделі,її позицію, функцію onClick,яка викликається при кліку на модель.Також має колір та логічне значення isActive,що визначає, чи активна зона.Змінна стану **орасіty** використовується для керування прозорістю моделі.

Якщо isActive -true, то прозорість змінюється циклічно (від 0.9 до 0.6) для створення ефекту "пульсації". Якщо isActive- false, прозорість скидається до 0.Використовується інтервал, який оновлює стан кожні 100 мс.

Лістинг 2.7 Створення інтерактивної зони з ефектом анімації прозорості

```
const InteractiveZone = ({ modelPath, position, onClick,
isActive, color }) => {
  const { scene } = useGLTF(modelPath);
  const [opacity, setOpacity] = useState(0);
  useEffect(() => {
    if (isActive) {
      setOpacity(0.9);
      const fadeInterval = setInterval(() => {
        setOpacity((prev) => {
          if (prev <= 0.6) {</pre>
            return 0.9;
          }
          return prev - 0.1;
        });
      }, 100);
      return () => clearInterval(fadeInterval);
    } else {
      setOpacity(0);
  }, [isActive]);
```

Цей фрагмент коду реалізує React-компонент InteractiveZone для інтерактивної роботи з 3D-моделями. За допомогою хука useGLTF завантажується модель за вказаним шляхом, а за допомогою стейту та useEffect керується прозорість (opacity) об'єкта. Цей підхід дозволяє користувачеві легко взаємодіяти з моделлю, надаючи зрозумілий візуальний сигнал про активний стан зони (див.рис.2.10-2.11)



Рис.2.10 - 2.11. Ефект анімації

Отже, завдяки цьому компоненту ми можемо відобразити інтерактивну зону,яка реагує на кліки та може змінювати прозорість.Це дуже зручно використовувати на віртуальній карті ,щоб виокремити одну аудиторію серед інших.

Далі оновлюємо матеріали всіх сіток (Mesh) у 3D-моделі, які знаходяться у сцені. Тобто створюється новий матеріал **MeshStandardMaterial**, який містить колір та прозорість, для забезпечення більш реалістичного освітлення та відображень.

Лістинг 2.8 Налаштування матеріалів сіток.

```
scene.traverse((child) => {
    if (child.isMesh) {
        child.material = new THREE.MeshStandardMaterial({
            color: color,
            transparent: true,
            opacity: opacity,
        });
    });
}
```

У коді використовується метод scene.traverse(), який проходить по всіх дочірніх об'єктах сцени. Якщо об'єкт є сіткою (child.isMesh), йому призначається новий матеріал з заданим кольором і прозорістю. Параметри transparent та opacity дозволяють візуально виділяти активні зони на карті — наприклад, підсвічувати аудиторії або інтерактивні ділянки. Такий підхід значно покращує візуальне сприйняття та взаємодію користувача з інтерфейсом.

Також перш за все було створено навігацію у вигляді "бургерного" меню.Це дуже необхідно і зручно, завдяки навігації є доступ до різних аудиторій у віртуальній карті, а також інтерактивна організація кабінетів у вигляді списку кнопок та акордеонів.

У файлі BurgerMenu.js було створено функції для зміни стану елементів.

Лістинг 2.9 Додавання функцій для зміни стану меню.

```
const toggleMenu = () => setIsMenuOpen(!isMenuOpen);
  const toggleFirstAccordion = () =>
setFirstAccordionOpen(!isFirstAccordionOpen);
  const toggleSecondAccordion = () =>
setSecondAccordionOpen(!isSecondAccordionOpen);
  const toggleThirdAccordion = () =>
setThirdAccordionOpen(!isSecondAccordionOpen);
```

Використано акордеони для групування аудиторій за категоріями, наприклад адміністрація академії зроблена окремими акордеоном.

Завдяки виклику даній будь-якої з функцій **toggle,** ми можемо оновлювати стан, це призводить до повторного рендерингу компонента і зміни його вигляду в інтерфейсі. Тобто може бути відкрите/закрите меню чи акордеони.

toggleMenu — перемикає стан відкриття/закриття меню. Якщо було відкрито закриває, і навпаки.

toggleFirstAccordion — розгортає або згортає перший акордеон.

Потім для кожної аудиторії створюється кнопка, яка викликає функцію onRoomClick з відповідним ідентифікатором.

Лістинг 2.10 Додавання кнопок для аудиторій.

```
<br/><button className="room-btn" onClick={() => onRoomClick("a1")}>
a1
</button>
<button className="room-btn" onClick={() => onRoomClick("a2")}>
a2
</button>
<button className="room-btn" onClick={() => onRoomClick("a3")}>
a3
</button>
```

Після цього у файлі BurgerMenu.css додали стилізацію кнопок, акордеонів та інших частин меню.

Ось такий вигляд це має на карті див.рис.2.12.

Закрити навігацію		Î
Пошук аудиторії	S)
Адміністрація		
Економічний факультет		
Аудиторії		
Аудиторія №1		_
Аудиторія №2		
Аудиторія №3		
Аудиторія №4		
Аудиторія №5		
Аудиторія №6		
Аудиторія №8		
Аудиторія №8а		



Перемикання розділів:

Для кожного з трьох розділів меню (admin, economic, auditorium) є кнопка **accordion-toggle**, яка відкриває або закриває відповідний розділ за допомогою функції toggleSection.Див.рис.2.13



Рис.2.13 Фрагмент Burger.Menu. Джерело: розроблено автором

Пошук:

Користувач вводить текст у поле для пошуку search-input, і цей текст зберігається в стані searchTerm.Пошук рефлексується на кожному кроці (при кожній зміні вхідного значення).

handleSearchChange — функція, яка викликається при зміні тексту в полі пошуку та оновлює значення searchTerm (див.рис.2.14).



Рис.2.14 Вигляд пошуку в меню

Джерело: розроблено автором

Фільтрація результатів пошуку:

Для кожного розділу меню є перевірка, чи потрібно відкривати цей розділ на основі результатів пошуку. Це визначається через функцію **shouldOpenSection**.Якщо

хоча б одна аудиторія в розділі відповідає пошуковому запиту, розділ буде відкрито автоматично.

useEffect: кожного разу, коли змінюється пошуковий запит (searchTerm), виконується перевірка всіх секцій і відкриваються ті, які містять аудиторії, що відповідають пошуку.

Для цього використовується функція **shouldOpenSection**, яка перевіряє, чи містить хоча б одна аудиторія в кожному розділі відповідний текст (див.рис.2.15).

Закрити навігацію		
1	4	
Адміністрація		
Економічний факультет		
Аудиторії		
Аудиторія №1		
Аудиторія П1		

Рис.2.15 Процес виконання пошуку Джерело: розроблено автором

Вибір аудиторії:

Коли користувач натискає на кнопку з назвою аудиторії, викликається функція **handleRoomClick**, яка змінює стан selectedRoom на ID вибраної аудиторії.

Потім викликається onRoomClick, передаючи йому ID аудиторії.

Адаптивність і автоматичне відкриття секцій:

Якщо користувач вводить запит у поле пошуку, програма автоматично відкриває відповідні розділи, якщо хоча б одна аудиторія в цьому розділі відповідає запиту.

Це дозволяє користувачеві легко знайти необхідну аудиторію, не витрачаючи час на вручну відкриття кожного розділу.

Фільтрація аудиторій:

Для кожного розділу **rooms** виконується фільтрація аудиторій на основі введеного пошукового запиту. Аудиторії, назва яких містить введений текст, будуть відображатися в меню, а інші - приховані.

Зовнішній вигляд і взаємодія:

Користувач може натискати на кнопки для відкриття секцій, вибору аудиторії та зміни пошукового запиту.

search-btn додає кнопку для запуску пошуку

room-btn — кнопка для кожної аудиторії, яка має інтерфейс для відображення її вибору (підсвічування, якщо аудиторія вибрана).

У меню є **пошуковий блок** з полем введення і кнопкою пошуку, а також секції з аудиторіями, що можна відкривати та закривати. Було використано **CSS класи** для визначення стилів, підключені через файл BurgerMenu.css.

Обробка подій на кліки поза меню:

За допомогою useEffect i handleClickOutside меню буде автоматично закриватися, якщо користувач натискає поза межами меню, покращуючи взаємодію.

Цей код реалізує зручну бокову навігацію з можливістю пошуку і фільтрації аудиторій. Використовуючи пошуковий запит, користувач може знайти потрібну аудиторію на різних поверхах та в різних секціях (адміністрація, економічний факультет, аудиторії). Програмно відкриваються тільки ті секції, які містять результати пошуку, що робить інтерфейс більш інтерактивним та зручним для користувача.

Всі реалізовані функції в меню є хорошим підходом для покращення UX.

Інтеграція зображень аудиторій

У процесі розробки інтерактивної карти Національного університету "Острозька академія" було здійснено фотографування всіх аудиторій з офіційного дозволу деканату. Отримані фотографії були використані для покращення інтерактивних зон у проєкті.

Для кожної інтерактивної зони, що відповідає певній аудиторії, було додано:

-Актуальні фотографії, що дозволяють користувачам отримати візуальне уявлення про приміщення.

-Короткий опис аудиторії, включаючи її призначення та особливості.

Додавання фотографій покращило користувацький досвід, зробивши навігацію по віртуальній карті більш інформативною та наочною. Це рішення також сприяє більш ефективному представленню університетської інфраструктури для абітурієнтів та відвідувачів.

Фото були додані до всіх аудиторій, до яких був доступ, ось так це виглядає на карті (див.рис.2.16).



Рис.2.16. Зображення реального фото аудиторії 2 поверху на карті *Джерело:* розроблено автором

Наступним кроком було створення Sidebar, щоб виводити інформацію про аудиторії на бічній панелі. Для цього ми створили об'єкт **roomInfo**, що містить дані про всі кімнати, а саме назву, зображення та опис. Також встановили ідентифікатори кімнат, такі як "a1", "a2", "rector", щоб використовувати як ключі в цьому об'єкті. Для панелі, зображень та тексту використовуються вбудовані стилі (sidebarStyles, imageStyles, textStyles), щоб задати вигляд.

Завдяки цим стилям, можемо:

-керувати розміщенням і розміром зображення;

-задавати стиль тексту (розмір, колір, відступи);

-оформити саму бічну панель (позиціювання, тінь, фон, прокрутку тощо).

Лістинг 2.11 Додавання інформації про аудиторію.

Ось так виглядає SideBar на карті, рис.2.17.



Рис.2.17. SideBar



Також невід'ємною частиною будь-якого проекту є Footer.Він містить логотип і текстову інформацію про проєкт.

Лістинг 2.12 Додавання Footer.

```
Перепелиця Діана та Місай Володимир
</div>
</footer>
);
};
export default Footer;
```

З метою підвищення зручності користування для студентів, до проєкту була додана мобільна версія. Це дозволяє комфортно взаємодіяти з картою університету зі смартфонів та планшетів. Було змінено користувацький інтерфейс, реалізовано мобільне бургер-меню для швидкого доступу до розділів. Також було додано обробку тач-подій для навігації по 3D-карті (див.рис.2.18).



Рис.2.18 Мобільна версія карти

Джерело: розроблено автором

Маємо ось такий фінальний результат віртуальної карти університету з освітленням та з налаштуваннями(див.рис. 2.19).



Рис. 2.19 Фінальний вигляд інтерактивної карти *Джерело:* розроблено автором

Висновок до розділу 2

У другому розділі було здійснено повноцінну прикладну реалізацію інтерактивної віртуальної карти Національного університету "Острозька академія", яка охоплює етапи збору інформації, ЗD-моделювання, створення середовища розробки, інтеграцію бібліотек та реалізацію функціоналу користувацької взаємодії.

На початковому етапі було зібрано необхідну інформацію про архітектуру навчального корпусу, що дало змогу точно відтворити будівлю у вигляді 3D-моделі. Моделі першого та другого поверхів були створені в середовищі Blender і експортовані у форматі .glb, що забезпечило їхню сумісність з WebGL-технологіями.

Для реалізації віртуальної карти було використано React як сучасний фреймворк для побудови інтерфейсу користувача, а бібліотека Three.js — для інтерактивного відображення 3D-сцен. Додаткові бібліотеки @react-three/fiber та @react-three/drei дозволили суттєво спростити роботу з WebGL у середовищі React.

Було реалізовано інтерактивну 3D-сцену з можливістю перемикання між поверхами, візуалізацією та підсвічуванням аудиторій, а також клікабельними

зонами. Компонент InteractiveZone дозволив додати функціонал анімації прозорості для активних аудиторій, що значно покращує візуальне сприйняття карти. Крім цього, реалізовано функціональне меню з акордеонами для зручної навігації по аудиторіях, а також можливість пошуку і фільтрації за назвою.

Результати роботи демонструють успішну реалізацію інтерактивної карти університету, яка є не лише візуально привабливою, але й функціонально зручною для кінцевих користувачів.

ВИСНОВКИ

В рамках виконання кваліфікаційної роботи було реалізовано інтерактивну віртуальну карту Національного університету "Острозька академія", яка відповідає сучасним вимогам до навігації в освітньому середовищі. Робота охоплює як теоретичне обґрунтування актуальності розробки, так і практичне розробку проєкту за допомогою сучасних технологій.

На першому етапі дослідження було сформульовано мету, об'єкт і предмет роботи, а також обґрунтовано доцільність створення віртуальної карти як важливого елементу цифрової інфраструктури університету. Зроблено висновок, що для студентів, особливо першокурсників, а також для гостей навчального закладу, зручна візуальна навігація є необхідною. Віртуальна карта дозволяє швидко орієнтуватися у просторі університету, надає додаткову інформацію про об'єкти на території закладу. Наступним етапом було розглянуто низку технологій для створення інтерактивної карти, зокрема React, Three.js, WebGL, а також інструменти для 3D-моделювання. За результатами порівняльного аналізу було обрано найоптимальніші з них: для фронтенд-розробки - React, для рендерингу 3D-сцен - Three.js у поєднанні з WebGL, а для створення моделей - Blender, який виявився не лише багатофункціональним і сучасним, а й безкоштовним, що є важливою перевагою.

На прикладному етапі було реалізовано повноцінний веб-застосунок, що містить інтерактивну 3D-карту навчального корпусу університету з підтримкою перемикання між поверхами, підсвічування аудиторій, інтерактивних зон і функціонального меню з пошуком. Моделі було створено в Blender, а бібліотеки @react-three/fiber та @react-three/drei значно спростили роботу з WebGL у середовищі React. Особливу увагу приділено деталізації користувацької взаємодії: реалізовано анімацію прозорості для активних аудиторій, систему фільтрації за назвами та інтерактивну навігацію через бічне меню. Це суттєво покращує інтерфейс для користувачів.Також до системи було додано фотографії аудиторій та інформаційні блоки, що робить карту ще більш інформативною. Результати роботи продемонстрували, що інтерактивна 3D-карта є не лише сучасним та візуально привабливим рішенням, а й практично важливим інструментом для університету. Вона підвищує рівень цифрової інтеграції в освітнє середовище, сприяє зручності та комфорту студентів, працівників і відвідувачів. Крім того, наявність такого інноваційного інструменту позитивно впливає на імідж навчального закладу, в якому добре розвинутий рівень інформаційних технологій.

Можемо підсумувати, що усі поставлені завдання були виконані, цілі досягнуто, а розроблена система має значний потенціал для подальшого вдосконалення й інтеграції в інші освітні та інформаційні сервіси університету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.React [Електронний ресурс].-Режим доступу: <u>https://uk.legacy.reactis.org/</u>(дата звернення: 10.01.2025р.) 2.React Component [Електронний ресурс].-Режим доступу: https://react.dev/reference/react/Component (дата звернення: 10.01.2025р.) 3. Vue.js Documentation [Електронний ресурс].-Режим доступу: https://vueis.org/guide/introduction (дата звернення: 11.01.2025р.) 4. Vue.js як javascript - фреймворк [Електронний ресурс].-Режим доступу: https://foxminded.ua/vue-js/ (дата звернення: 11.01.2025р.) 5. Babylon.js Documentation [Електронний ресурс].-Режим доступу https://doc.babylonis.com/ (дата звернення: 13.01.2025р.) 6. Three.js Documentation [Електронний ресурс].-Режим доступу https://threejs.org/docs/(дата звернення: 14.01.2025р.) 7. What is WebGL and why use Three.js [Електронний ресурс].-Режим доступу https://threejs-journey.com/lessons/what-is-webgl-and-why-use-three-js#introduction (дата звернення: 14.01.2025р.) 8. WebGL Documentation [Електронний ресурс]. - Режим доступу https://developer.mozilla.org/en-US/docs/Web/API/WebGL API (дата звернення: 15.01.2025р.) 9. Основи WebGL [Електронний ресурс]. - Режим доступу https://webglfundamentals.org/webgl/lessons/uk/ (дата звернення: 15.01.2025р.) 10.React +Three.js [Електронний ресурс].-Режим доступу https://mediusware.com/blog/details/integrating-threejs-with-react (дата звернення: 16.01.2025p.)

11.Blender [Електронний ресурс].-Режим доступу

https://docs.blender.org/manual/en/latest/ (дата звернення: 17.01.2025р.)

12.3ds Max [Електронний ресурс].-Режим доступу

https://uk.wikipedia.org/wiki/Autodesk_3ds_MAX (дата звернення: 17.01.2025р.)

13.Cinema 4D [Електронний ресурс].-Режим доступу

https://uk.wikipedia.org/wiki/Cinema_4D (дата звернення: 17.01.2025р.)

14.Мапа ТРЦ «Оазис» [Електронний ресурс].-Режим доступу

https://www.oazis.km.ua/map (дата звернення: 18.01.2025р.)

15. Карта «Злата Плаза» [Електронний ресурс].-Режим доступу

<u>https://zlata-plaza.com.ua/floor-maps#floor1</u> (дата звернення: 18.01.2025р.)

16.Мапа Бостонського університету [Електронний ресурс].-Режим доступу

https://boston.university-tour.com/map.php

17.Bringing 3D to Life: Integrating Three.js with React, Redux and MUI [Електронний ресурс].-Режим доступу

<u>https://chamallakshika09.medium.com/bringing-3d-to-life-integrating-three-js-with-react-r</u> edux-and-mui-6471bd41e475(дата звернення: 19.01.2025р.)

18.React-Three/Fiber [Електронний ресурс].-Режим доступу

<u>https://r3f.docs.pmnd.rs/getting-started/introduction?trk=public_post_comment-text(дата</u> звернення: 20.01.2025р.)

19.React-Three/Drei [Електронний ресурс].-Режим доступу

https://www.npmjs.com/package/@react-three/drei/v/9.0.1(дата звернення: 20.01.2025р.)

20.OrbitControls[Електронний ресурс].-Режим доступу

https://threejs.org/docs/#examples/en/controls/OrbitControls(дата звернення:

20.01.2025p.)

21. UseGLTF <u>https://sbcode.net/react-three-fiber/use-gltf/(д</u>ата звернення: 20.01.2025р.)

21.How to Build Interactive 3D Apps <u>https://formidable.dev/blog/2021/react-three-fiber/</u> (дата звернення: 21.01.2025р.)