

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ

Навчально-науковий інститут денної освіти

Форма навчання денна

Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту

Завідувач кафедри

_____ Олена ОЛЬХОВСЬКА

(підпис)

«_____» ____ 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

АЛГОРИТМІЗАЦІЯ ТА РОЗРОБКА НАВЧАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З РОЗДІЛУ «СТЕРЕОМЕТРІЯ» ШКІЛЬНОГО КУРСУ МАТЕМАТИКИ

**зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня магістра**

Виконавець роботи Тріфанов Віталій Олексійович

_____ «_____» _____ 202_ р.
(підпис)

Науковий керівник доцент, к.ф.-м.н. Черненко О. О.

_____ «_____» _____ 202_ р.
(підпис)

Рецензент

ПОЛТАВА 2024

РЕФЕРАТ

Записка: 73 с., 13 рис., 2 таблиці, 1 додаток, 11 джерел.

СТЕРЕОМЕТРИЯ, REACTJS, ВІЗУАЛІЗАЦІЯ, НАВЧАЛЬНЕ ПЗ

Об'єктом розробки є інтерактивне навчальне програмне забезпечення для вивчення стереометрії.

Предметом розробки є програмна реалізація веб-додатка для вивчення та закріплення знань із основ стереометрії.

Метою роботи є створення навчального веб-додатка для вивчення стереометрії, який забезпечує візуалізацію геометричних фігур, інтерактивну теорію, тестування знань і зручний інтерфейс для користувачів.

Результатом роботи стало розроблення інтерактивного навчального веб-додатка для вивчення стереометрії. Сайт поєднує в собі кілька основних компонентів, що забезпечують як теоретичне навчання, так і практичне закріплення матеріалу. У розділі теорії користувачеві доступний структурований навчальний матеріал, що містить детальні описи основних аксіом, взаємного розташування прямих та площин, а також формули для обчислення відстаней та площ. Для полегшення сприйняття інформації розроблено ілюстрації, що супроводжуються поясненнями та прикладами задач. Математичні формули на сторінках відображаються з використанням бібліотеки KaTeX, що дозволяє показувати їх у зрозумілому для користувача вигляді.

Однією з ключових особливостей веб-додатка є інтерактивний візуалізатор геометричних фігур, створений за допомогою бібліотеки Three.js. Користувачі мають можливість обирати різні фігури, такі як куб, піраміда, конус чи циліндр, а також взаємодіяти з ними: обертати об'єкти, змінювати їхній колір та перемикати режим відображення між каркасом і суцільною моделлю. Це дозволяє глибше зрозуміти властивості фігур у тривимірному просторі.

Для перевірки засвоєних знань на платформі реалізовано систему тестування. Користувачеві пропонуються питання, що стосуються обчислення об'ємів та площ поверхонь фігур, знаходження відстаней між об'єктами, а також аналізу їх

взаємного розташування. Тести супроводжуються зворотним зв'язком: у випадку правильної відповіді система відзначає результат зеленим кольором, а у разі помилки надає пояснення з правильною відповіддю.

Окрім основного функціоналу, платформа містить сторінку "Контакти", що дозволяє користувачам зв'язатися з розробниками через інтерактивну форму зворотного зв'язку. Сайт має сучасний та адаптивний дизайн, що забезпечує коректне відображення на різних пристроях, включаючи мобільні телефони, планшети та комп'ютери.

Платформа була протестована на різних пристроях та у популярних веб-браузерах. Робота програми підтверджена як стабільна та ефективна, а інтерфейс інтуїтивно зрозумілий для користувачів різного віку.

Робота має практичне значення, оскільки може використовуватися як у навчальному процесі в школах та університетах, так і для самостійного вивчення стереометрії. Сайт поєднує сучасні технології, інтуїтивний дизайн та інтерактивні інструменти, що робить вивчення стереометрії цікавішим та доступнішим для користувачів.

ЗМІСТ

ВСТУП.....	7
ПОСТАНОВКА ЗАДАЧІ.....	9
1. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	11
1.1. Сучасні підходи до створення навчального програмного забезпечення.....	11
1.2. Методи візуалізації математичних об'єктів	12
1.3. Огляд існуючих програмних рішень	13
1.4. Аналіз навчальних програм і стандартів з математики.....	17
2. ТЕОРЕТИЧНА ЧАСТИНА.....	19
2.1. Математичні основи стереометрії	19
2.2. Геометричні алгоритми побудови об'єктів	21
2.3. Алгоритми взаємодії з користувачем.....	23
2.4. Методи контролю знань учнів	26
3. ПРАКТИЧНА ЧАСТИНА	29
3.1. Проектування програмного забезпечення	29
3.2. Вибір технологій та архітектура системи	31
3.3. Реалізація основних модулів.....	32
3.4. Контроль версій та розгортання сайту в інтернеті	36
3.5. Інструкція для користувача.....	38
ВИСНОВКИ.....	45
СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.....	47
ДОДАТОК А.	48

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
3D	Тривимірне зображення або об'єкт у просторі
HTML	Мова розмітки гіпертексту для створення структури веб-сторінок
CSS	Каскадні таблиці стилів, що використовуються для оформлення веб-сторінок
JavaScript	Мова програмування для створення інтерактивності на веб-сайтах
React.js	JavaScript-бібліотека для побудови інтерфейсу користувача
KaTeX	Бібліотека для відображення математичних формул на веб-сторінках
Three.js	JavaScript-бібліотека для створення 3D-графіки у браузері
API	Інтерфейс прикладного програмування для обміну даними між програмами
SVG	Scalable Vector Graphics – формат векторної графіки для ілюстрацій
UI/UX	User Interface (інтерфейс користувача) / User Experience (досвід користувача)
npm	Node Package Manager – менеджер пакетів для JavaScript
Git	Система контролю версій для відстеження змін у програмному коді
Netlify	Платформа для хостингу веб-додатків
БД	База даних

Функціонал	Сукупність функцій, які реалізуються програмою
Інтерактивність	Здатність системи реагувати на дії користувача
Площина	Геометричне поняття – поверхня, що містить усі прямі, які проходять через дві точки
Об'єм	Простір, який займає тривимірний об'єкт
Площа поверхні	Сума площ усіх граней геометричної фігури
Координати (x, y, z)	Числові значення, що визначають положення точки у тривимірному просторі
d	Відстань між об'єктами або точками
R	Радіус кулі або кола
S	Площа поверхні
V	Об'єм геометричної фігури

ВСТУП

У сучасному світі розвиток інформаційних технологій суттєво впливає на освітній процес. Використання навчального програмного забезпечення дозволяє підвищити ефективність засвоєння знань, активізувати пізнавальну діяльність учнів та розвивати їх просторове мислення. Зокрема, вивчення математики, а саме розділу «Стереометрія», вимагає від учнів здатності візуалізувати геометричні об'єкти у тривимірному просторі.

Стереометрія є важливим розділом геометрії, що вивчає фігури у просторі, їх властивості та взаємне розташування. Проте складність абстрактного мислення учнів під час вивчення просторових фігур часто ускладнює навчальний процес. Використання інтерактивних програмних засобів може стати ефективним рішенням для візуалізації просторових об'єктів та підтримки активного навчання.

Розробка навчального програмного забезпечення з розділу «Стереометрія» сприятиме формуванню в учнів уявлення про просторові об'єкти, розвиткові логічного мислення, вмінню аналізувати та узагальнювати інформацію. Інтерактивні інструменти допоможуть створити віртуальні моделі геометричних об'єктів, проводити обчислення та контролювати знання.

Актуальність дослідження полягає у необхідності розробки інтерактивного навчального програмного забезпечення з використанням сучасних технологій для полегшення засвоєння складних понять стереометрії.

Мета роботи — алгоритмізація та розробка навчального програмного забезпечення для підтримки вивчення розділу «Стереометрія» шкільного курсу математики.

Завдання роботи:

- Проаналізувати існуючі програмні рішення та сучасні підходи до створення навчальних програм.
- Визначити математичні основи стереометрії, що підлягають візуалізації.
- Розробити алгоритми для побудови та візуалізації геометричних об'єктів.
- Спроектувати інтерфейс користувача для зручної взаємодії з програмою.

- Реалізувати функціональні модулі та забезпечити їх інтеграцію у навчальний процес.
- Перевірити ефективність програмного забезпечення на практиці.

Об'єкт дослідження — процес вивчення стереометричних об'єктів у шкільному курсі математики за допомогою програмного забезпечення.

Предмет дослідження — методи алгоритмізації, проектування та розробки навчального програмного забезпечення з використанням сучасних інформаційних технологій.

Практичне значення роботи полягає у створенні інтерактивного програмного продукту, який може бути використаний вчителями математики у процесі навчання учнів старших класів.

ПОСТАНОВКА ЗАДАЧІ

Основною метою даного проекту є розробка інтерактивного навчального програмного забезпечення для вивчення стереометрії, яке сприятиме підвищенню зацікавленості учнів у математиці, полегшенню засвоєння складних просторових понять та розвитку логічного й просторового мислення. Інтерактивний підхід дозволяє забезпечити ефективний формат навчання, що відповідає сучасним вимогам до освітніх інструментів.

Основні завдання:

1. Визначення ключових тем стереометрії: розробка теоретичних модулів, що охоплюють основні аксіоми стереометрії, рівняння площини, відстані між точками та площинами, об'єми та площі поверхонь фігур, многогранники та формулу Ейлера.
2. Розробка інтерактивних модулів:
 - Візуалізація геометричних фігур (площини, точки, прямі, многогранники) у тривимірному просторі.
 - Анімація процесів побудови та взаємодії просторових об'єктів.
 - Інтерактивні завдання з перевіркою відповідей та підказками.
3. Проектування архітектури програмного забезпечення:
 - Вибір сучасних технологій для клієнтської частини (React, TypeScript) і серверної частини (Node.js).
 - Створення модульної архітектури для можливості додавання нових функцій.
4. Розробка інтуїтивного інтерфейсу користувача:
 - Проектування навігації для швидкого доступу до розділів теорії, практичних завдань та тестів.
 - Забезпечення відповідності стандартам UX/UI-дизайну.
5. Система тестування та контролю знань:
 - Впровадження модулів автоматичної перевірки відповідей та фіксації результатів.

- Збереження результатів виконання завдань у базі даних для подальшого аналізу прогресу.

6. Забезпечення адаптивності дизайну:

- Оптимізація роботи на пристроях різних розмірів (комп'ютери, планшети, смартфони).
- Перевірка сумісності з сучасними веб-браузерами.

7. Створення супровідної документації:

- Написання інструкції для користувачів, що описує функціонал та можливості системи.
- Оформлення технічної документації для підтримки та подальшого розвитку проекту.

Головною вимогою до програмного забезпечення є його стабільність, зручність та корисність для учнів і вчителів. Воно має забезпечити інтерактивний підхід до вивчення стереометрії, сприяти закріпленню знань та підтримувати різні формати навчання – теоретичний, практичний та тестовий.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1. Сучасні підходи до створення навчального програмного забезпечення

Стрімкий розвиток інформаційних технологій сприяв появі нових методів створення навчального програмного забезпечення (НПЗ), які забезпечують інтерактивність, персоналізацію та доступність навчальних ресурсів. Сучасні системи спрямовані на адаптацію до рівня знань учнів, підтримку мультимедійних форматів, гейміфікацію та інтеграцію хмарних технологій.

Одним із ключових підходів є інтерактивне навчання, яке передбачає активну взаємодію учнів із контентом. Наприклад, віртуальні симуляції та моделі 3D дозволяють створити віртуальні лабораторії для вивчення складних концепцій. Використання інтерактивних завдань, таких як тести з миттєвим зворотним зв'язком, допомагає учням краще засвоювати матеріал.

Ще одним важливим напрямком є адаптивне навчання, яке базується на аналізі успішності учня в реальному часі. Системи з елементами штучного інтелекту можуть пропонувати завдання відповідно до рівня знань, забезпечуючи індивідуальний підхід. Наприклад, платформи Coursera та Khan Academy застосовують алгоритми адаптації для корекції освітнього маршруту кожного учня.

Мультимедійні технології також стали невід'ємною частиною навчального процесу. Використання аудіо, відео, анімацій і текстів забезпечує глибоке занурення в навчальний матеріал. Наприклад, відеоуроки з геометрії дозволяють учням візуалізувати складні просторові об'єкти, що робить їх більш зрозумілими.

Гейміфікація навчального процесу допомагає підтримувати інтерес учнів. Це може бути реалізовано через систему балів, нагород, рівнів і досягнень. Наприклад, освітні платформи Duolingo та Kahoot використовують гейміфікацію, щоб підтримувати мотивацію учнів і спонукати їх до постійного навчання. [1]

Окремо слід згадати хмарні технології, які дозволяють зберігати великі обсяги навчальних даних і забезпечувати доступ до них з будь-якого пристрою. Це робить навчальний процес гнучким і зручним. Наприклад, Google Classroom дозволяє створювати віртуальні класи, де учні можуть взаємодіяти з учителями та

отримувати навчальні матеріали онлайн.

Підтримка мобільного навчання відкриває можливості для навчання будь-де і будь-коли. Мобільні додатки з інтерактивними завданнями дозволяють учням навчатися навіть у транспорті або під час відпочинку.

Усі ці сучасні підходи до створення навчального програмного забезпечення дозволяють забезпечити індивідуальний підхід до навчання, підвищити ефективність засвоєння знань та зробити процес навчання цікавим і захопливим. У розробці програмного забезпечення з розділу «Стереометрія» ці методи будуть враховані для створення сучасного та ефективного інструменту для викладання складних геометричних концепцій.

1.2. Методи візуалізації математичних об'єктів

Візуалізація математичних об'єктів є важливим елементом сучасного навчального процесу, оскільки вона допомагає учням краще зрозуміти абстрактні математичні поняття, зробити їх наочними та легшими для засвоєння. Різні методи візуалізації застосовуються в освітніх програмах з урахуванням специфіки навчального матеріалу та цільової аудиторії.

Графічні методи візуалізації базуються на використанні рисунків, схем та діаграм для відображення математичних об'єктів. Це дозволяє побудувати наочні образи таких понять, як графіки функцій, геометричні фігури, просторові конструкції. Наприклад, побудова площин у тривимірному просторі чи відображення перетину геометричних тіл.

Динамічна візуалізація передбачає використання анімацій для демонстрації змін об'єктів із часом або залежно від заданих параметрів. Це допомагає наочно показати, як змінюються об'єкти при виконанні певних математичних операцій, таких як обертання, масштабування чи трансформація геометричних фігур.

Інтерактивні моделі дозволяють учням безпосередньо взаємодіяти з математичними об'єктами. Наприклад, змінювати параметри функцій, будувати фігури, обчислювати об'єми або площі поверхонь. Це підвищує рівень

зацікавленості та сприяє кращому розумінню матеріалу.

Для представлення математичних формул та рівнянь часто використовуються системи комп'ютерної алгебри або спеціальні математичні редактори. Це дозволяє точно відобразити складні математичні вирази, необхідні для розв'язування задач у таких розділах математики, як алгебра чи стереометрія.

Сучасні технології дозволяють використовувати віртуальну та розширену реальність для створення математичних моделей у тривимірному просторі. Учні можуть досліджувати математичні об'єкти у VR-середовищі, змінювати їх властивості та спостерігати за результатами своїх дій у реальному часі. [2]

Методи візуалізації математичних об'єктів є важливим інструментом для покращення сприйняття складних понять. У процесі розробки програмного забезпечення для вивчення стереометрії передбачається використання графічних методів, динамічних моделей та інтерактивних завдань для забезпечення ефективного навчання.

1.3. Огляд існуючих програмних рішень

Сучасні технології значно розширили можливості навчання, дозволяючи створювати програмне забезпечення, яке робить процес навчання більш інтерактивним та ефективним. У цій частині роботи ми розглянемо існуючі програми для вивчення стереометрії, їх функціональні можливості, переваги та недоліки. Це допоможе визначити, які аспекти можуть бути покращені в нашій розробці.

На сьогоднішній день існує багато різноманітного навчального програмного забезпечення для вивчення стереометрії. Серед них можна виділити як комерційні продукти, так і безкоштовні програми. Деякі з них спеціалізуються на певних аспектах стереометрії, таких як візуалізація геометричних фігур, інші ж надають комплексний підхід до вивчення предмету (див. табл. 1.1).

Основні програми для вивчення стереометрії

1. GeoGebra

- Переваги: Інтуїтивно зрозумілий інтерфейс, висока якість візуалізації, інтерактивність.
- Недоліки: Комерційний продукт, висока вартість ліцензії, обмежена підтримка мов.

○

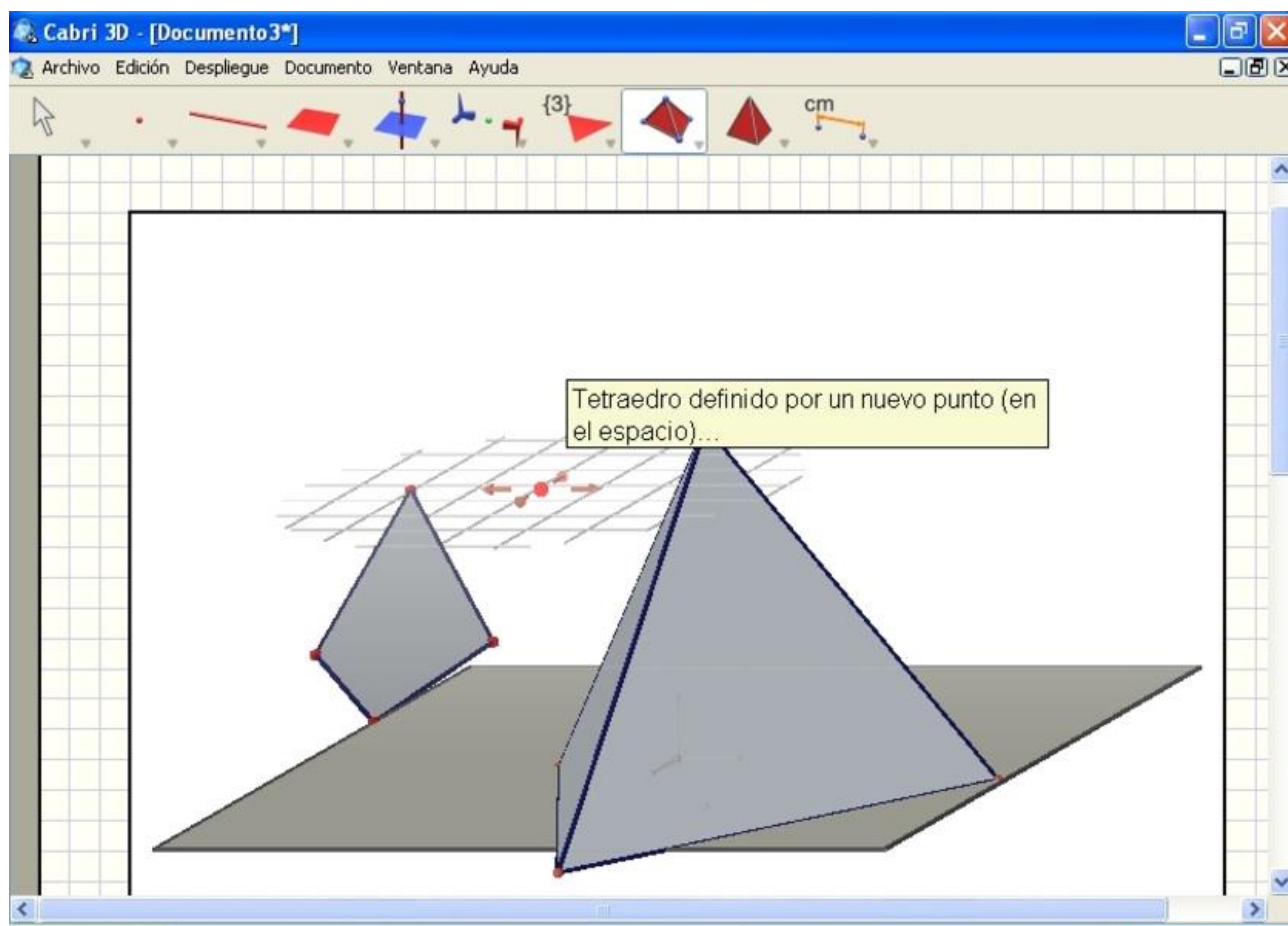


Рисунок 1.2 – Приклад роботи Cabri 3D

3. SketchUp

- Опис: SketchUp — це програма для тривимірного моделювання, яка широко використовується в архітектурі, дизайні та освіті (див. рис. 1.3). Вона дозволяє створювати детальні тривимірні моделі, що можуть бути використані для вивчення стереометрії.
- Функціональні можливості: Створення та редагування тривимірних моделей, інтерактивні інструменти для маніпуляції об'єктами, можливість експорту в різні формати.

- Переваги: Потужні інструменти для моделювання, зручний інтерфейс, безкоштовна версія з базовими функціями.
- Недоліки: Висока вартість професійної версії, складність для початківців.



Рисунок 1.3 – Приклад роботи SketchUp

Таблиця 1.1 – Порівняння програм

Програма	Вартість	Інтерфейс	Функціональні можливості	Платформа	Мова
GeoGebra	Безкоштовно	Складний	Широкий	Усі	Багатомовна
Cabri 3D	Платно	Інтуїтивний	Висока якість візуалізації	Windows, Mac	Обмежена підтримка
SketchUp	Безкоштовна / Платна	Зручний	Потужні інструменти для моделювання	Windows, Mac	Англійська

Аналіз існуючого навчального програмного забезпечення для вивчення стереометрії показує, що кожна з розглянутих програм має свої переваги та недоліки. GeoGebra пропонує широкий функціонал і є безкоштовною, але має складний інтерфейс. Cabri 3D відрізняється високою якістю візуалізації, проте є платною програмою. SketchUp пропонує потужні інструменти для моделювання,

але може бути складним для новачків. Розробка нового навчального програмного забезпечення повинна враховувати ці фактори та прагнути об'єднати кращі якості існуючих програм, забезпечуючи при цьому доступність, інтуїтивність та ефективність у навчанні стереометрії. [3]

1.4. Аналіз навчальних програм і стандартів з математики

Аналіз навчальних програм і стандартів з математики є важливим етапом розробки навчального програмного забезпечення, оскільки він дозволяє визначити ключові теми для вивчення та забезпечити відповідність проєкту офіційним вимогам освітніх стандартів.

Навчальні програми з математики базуються на затверджених освітніх стандартах, що регулюють обсяг знань, які повинні бути засвоєні учнями на кожному етапі навчання. До основних міжнародних стандартів належать національні стандарти математичної освіти (NCTM), що охоплюють фундаментальні математичні концепції, серед яких геометрія, алгебра, тригонометрія та стереометрія. Програма Міжнародного бакалаврату (IB) також передбачає поглиблений курс математики, який зосереджується на теоретичному вивченні математичних понять та їх практичному застосуванні.

У контексті української освіти, державні стандарти регламентують обов'язкове вивчення геометрії та стереометрії у старших класах загальноосвітніх шкіл. Важливість цього розділу підкреслюється ключовими темами, серед яких розташування точок, прямих і площин у просторі, геометричні тіла (призми, піраміди, циліндри, конуси та кулі), обчислення площ поверхонь і об'ємів тіл, а також побудова перерізів многогранників і тіл обертання.

Особливу увагу заслуговують підручники з геометрії, рекомендовані Міністерством освіти і науки України, оскільки вони є основним джерелом знань для учнів. Аналіз підручників дозволяє виділити кілька важливих аспектів, на які слід орієнтуватися під час створення програмного забезпечення. Теоретичний матеріал у підручниках викладається чітко та послідовно, а велика кількість

прикладів і задач дозволяє закріпити отримані знання. Крім того, практичні завдання, що стосуються побудови фігур і обчислення їх параметрів, є важливою складовою процесу навчання.

З огляду на вищезазначене, програмне забезпечення для вивчення стереометрії повинно відповідати декільком ключовим вимогам. По-перше, воно має охоплювати всі обов'язкові теми відповідно до навчальних програм і стандартів. По-друге, важливо забезпечити якісну візуалізацію геометричних об'єктів із можливістю інтерактивної взаємодії, що дозволить учням краще зрозуміти просторові концепції. По-третє, інтерактивність програми повинна сприяти активній участі користувача у процесі навчання завдяки задачам, тестам та симуляціям. Також важливим є функціонал для перевірки рівня знань і відстеження прогресу учнів.

Таким чином, аналіз навчальних програм і стандартів з математики став основою для визначення структури і функціональних можливостей розробленого програмного забезпечення. Врахування освітніх вимог і потреб сучасного навчального процесу дозволяє створити ефективний інструмент для викладання стереометрії, що поєднує доступність, наочність і інтерактивність. Це, у свою чергу, сприяє підвищенню зацікавленості учнів та покращенню якості вивчення математики у школах і ліцейх.

2. ТЕОРЕТИЧНА ЧАСТИНА

2.1. Математичні основи стереометрії

Стереометрія — це розділ геометрії, який вивчає властивості геометричних фігур у тривимірному просторі. Основні об'єкти стереометрії включають точки, прямі, площини, многогранники та тіла обертання. Математичні основи стереометрії базуються на аксіоматичній системі Евклідової геометрії, яка описує взаємне розташування цих об'єктів.

Основні поняття стереометрії

1. Точка, пряма і площина: базові елементи стереометрії, які не мають розмірів:

- Точка — визначає положення в просторі.
- Пряма — нескінченна лінія, яка визначається двома точками.
- Площина — двовимірна поверхня, що проходить через три точки, які не лежать на одній прямій.

2. Взаємне розташування об'єктів у просторі:

- Точка і пряма: Точка може належати або не належати прямій.
- Пряма і площина: Пряма може перетинати площину, лежати у площині або бути паралельною до неї.
- Дві площини: Площини можуть перетинатися по прямій або бути паралельними. [5]

Основні аксіоми стереометрії

1. Аксіома 1: Через будь-які три точки, що не лежать на одній прямій, проходить одна площина.
2. Аксіома 2: Якщо дві точки прямої належать площині, то вся пряма лежить у цій площині.
3. Аксіома 3: Якщо дві площини мають спільну точку, то вони перетинаються по прямій.

Теореми стереометрії

1. Теорема про існування площини: Через пряму і точку поза нею можна

провести площину, і до того ж тільки одну.

2. Теорема про перетин площин: Дві пересічні площини перетинаються по прямій.
3. Теорема про паралельність прямих: Дві прямі, які не перетинаються, або паралельні, або скрещуються у просторі.

Основні фігури стереометрії

1. Призма: Многогранник із двома паралельними основами, які є рівними багатокутниками.
2. Піраміда: Многогранник, основа якого — багатокутник, а решта граней — трикутники, які мають спільну вершину.
3. Циліндр: Тіло обертання з круговою основою, яке утворюється при обертанні прямокутника навколо однієї зі сторін.
4. Конус: Тіло обертання з круговою основою, утворене при обертанні прямокутного трикутника навколо одного з катетів.
5. Куля: Тіло обертання, яке утворюється при обертанні півкола навколо його діаметра.

Формули, що використовуються у стереометрії

1. Площа поверхні:
 - Куб: $S = 6a^2$
 - Призма: $S = 2S_{\text{основи}} + P_{\text{основи}} * h$
 - Куля: $S = 4\pi R^2$
2. Об'єм:
 - Куб: $V = a^3$
 - Призма: $V = S_{\text{основи}} * h$
 - Куля: $V = \frac{4}{3}\pi R^3$

Математичні основи стереометрії формують базу для вивчення тривимірних об'єктів та їх властивостей. Розуміння аксіом, теорем і базових фігур дозволяє здійснювати складні обчислення, моделювати геометричні фігури та розв'язувати просторові задачі. Це є ключем до створення ефективного навчального програмного забезпечення для вивчення стереометрії.

2.2. Геометричні алгоритми побудови об'єктів

Геометричні алгоритми побудови об'єктів є основою для створення навчального програмного забезпечення з вивчення стереометрії. Ці алгоритми дозволяють будувати тривимірні геометричні фігури на основі математичних правил і візуалізувати їх у реальному часі. У цій частині розглянемо основні підходи до побудови об'єктів, математичні основи алгоритмів та їх реалізацію.

Основні етапи побудови геометричних об'єктів

1. Задання початкових параметрів: включає вибір початкових координат точок, визначення параметрів (довжини ребер, радіусів, висот) та обчислення допоміжних величин.
2. Розрахунок координат точок: обчислюються координати вершин фігур за допомогою формул геометрії у просторі. Наприклад, для побудови прямокутного паралелепіпеда необхідно визначити всі вісім вершин.
3. З'єднання точок для створення ребер і граней: створюються відрізки (ребра) між точками, що визначають межі фігур, та площини (грані) на основі полігональних об'єктів. [6]
4. Візуалізація об'єкта: візуалізація включає відображення об'єкта у тривимірному просторі за допомогою комп'ютерної графіки. Це може бути як статична модель, так і інтерактивна, з можливістю обертання та масштабування.

Приклади алгоритмів побудови геометричних об'єктів

1. Побудова точки у просторі

Алгоритм задає координати точки у тривимірній системі координат:

- Вхідні дані: (x, y, z) — координати точки.
- Вихідні дані: візуалізація точки у просторі.

2. Побудова прямої

Пряма задається двома точками $A(x_1, y_1, z_1)$ та $B(x_2, y_2, z_2)$

Алгоритм будує відрізок:

1. Обчислити різницю координат точок.

2. Побудувати відрізок між точками А та В.

3. Побудова площини

Площина визначається трьома точками $A(x_1, y_1, z_1)$, $B(x_2, y_2, z_2)$, $C(x_3, y_3, z_3)$:

1. Знайти нормальний вектор площини: $\vec{n} = \vec{AB} \times \vec{AC}$

2. Визначити рівняння площини: $Ax + By + Cz + D = 0$

4. Побудова многогранників (приклад — куб)

1. Визначити центр куба та довжину ребра.

2. Обчислити координати всіх вершин.

3. З'єднати відповідні вершини для побудови граней.

Алгоритми побудови об'ємних тіл обертання

1. Побудова циліндра

- Вхідні дані: радіус основи R, висота h.

- Кроки побудови:

1. Побудувати дві кругові основи.

2. З'єднати відповідні точки основ для формування бічної поверхні.

2. Побудова конуса

- Вхідні дані: радіус основи R, висота h.

- Кроки побудови:

1. Побудувати кругову основу.

2. З'єднати всі точки основи з вершиною конуса.

3. Побудова кулі

- Вхідні дані: радіус R.

- Кроки побудови:

1. Задати центр кулі.

2. Візуалізувати кулю через координати точок на її поверхні за допомогою сферичних координат:

$$x = R \cos(\theta) \sin(\varphi)$$

$$y = R \sin(\theta) \sin(\varphi)$$

$$z = R \cos(\varphi)$$

Реалізація алгоритмів у програмному забезпеченні

Для побудови геометричних об'єктів використовуються мови програмування та бібліотеки комп'ютерної графіки, такі як:

- JavaScript (Three.js): для створення інтерактивних тривимірних сцен.
- Python (Matplotlib, PyOpenGL): для побудови тривимірних моделей.
- C++ (OpenGL): для складних графічних додатків.

Геометричні алгоритми побудови об'єктів є основою для створення навчального програмного забезпечення з візуалізації стереометричних фігур. Вони дозволяють автоматизувати процес побудови, забезпечуючи інтерактивну взаємодію та візуалізацію математичних об'єктів у реальному часі. Ці алгоритми повинні враховувати точність обчислень, ефективність відображення та зручність користування.

2.3. Алгоритми взаємодії з користувачем

Алгоритми взаємодії з користувачем у навчальному програмному забезпеченні є ключовим аспектом, який визначає зручність користування, швидкість доступу до функцій та рівень залучення учнів до навчального процесу. У цьому розділі розглянемо основні алгоритми, що забезпечують ефективну взаємодію з користувачем у контексті вивчення стереометрії.

Основні принципи взаємодії з користувачем

Розробка інтерфейсу користувача базується на кількох основних принципах:

1. Інтуїтивність: Інтерфейс має бути зрозумілим без попереднього навчання.
2. Доступність: Додаток повинен працювати на різних пристроях та підтримувати адаптивний дизайн.
3. Візуальна привабливість: Використання графічних елементів має сприяти кращому сприйняттю інформації.
4. Інтерактивність: Користувач повинен мати змогу активно взаємодіяти з навчальним матеріалом.

Алгоритми взаємодії з користувачем

1. Обробка введення користувача

Обробка введення забезпечує взаємодію користувача з додатком через мишу, клавіатуру або сенсорний екран. Основні алгоритми:

- Введення тексту: Обробка текстових даних для введення відповідей або числових значень.
- Кнопковий інтерфейс: Обробка натискань на інтерактивні кнопки для виклику функцій або переходу між сторінками.

Приклад алгоритму введення координат точки:

1. Користувач вводить значення координат (x,y,z).
2. Введені дані перевіряються на відповідність числовому формату.
3. Якщо перевірка успішна — значення зберігаються. Інакше виводиться повідомлення про помилку.

2. Навігація між розділами

Навігація дозволяє користувачам легко переходити між різними розділами програми.

- Меню навігації: Відображає список доступних розділів з можливістю швидкого доступу. [7]
- Кнопки «Назад» і «Далі»: Забезпечують покрокову навігацію.

Алгоритм роботи меню навігації:

1. Визначається розташування користувача в поточному розділі.
2. За натисканням на пункт меню виконується перенаправлення на відповідний розділ.
3. Зберігається стан навігації для повернення до попереднього розділу.

3. Інтерактивні елементи навчання

Це динамічні компоненти, з якими користувач може взаємодіяти безпосередньо:

- Обертання об'єктів: Використовується для вивчення геометричних тіл з різних ракурсів.
- Переміщення фігур: Дозволяє змінювати положення об'єктів на екрані.
- Зміна параметрів: Користувач може змінювати параметри, такі як довжина сторін, радіус, висота об'єкта.

Алгоритм взаємодії з тривимірною моделлю:

1. Визначення початкового положення миші або сенсорного дотику.
2. Обробка переміщення в просторі.
3. Оновлення координат об'єкта відповідно до руху користувача.
4. Перемальовування сцени.

4. Перевірка відповідей і зворотний зв'язок

Навчальна програма повинна мати механізм перевірки правильності відповідей та надання зворотного зв'язку.

- Автоматична перевірка: Введені дані перевіряються за заданими алгоритмами. [8]
- Повідомлення про результат: Успішні відповіді відзначаються позитивними повідомленнями, а помилки — рекомендаціями щодо виправлення.

Алгоритм перевірки відповіді:

1. Користувач вводить відповідь у поле введення.
2. Програма перевіряє відповідність даних коректній відповіді.
3. Якщо відповідь правильна — відображається повідомлення «Вірно!».
4. Якщо відповідь неправильна — надається підказка або зразок правильного розв'язання.

5. Збереження та відновлення прогресу

Ця функція дозволяє зберігати прогрес користувача у навчанні та відновлювати його пізніше.

- Збереження даних: Програма записує стан користувача у локальне сховище або на сервер.
- Відновлення: Під час наступного входу програма завантажує збережені дані та відновлює попередній стан.

Алгоритм збереження прогресу:

1. Виконується перевірка наявності облікового запису користувача.
2. Записуються дані про завершені завдання, оцінки та час роботи.
3. У разі успішного збереження відображається повідомлення «Прогрес

збережено».

Алгоритми взаємодії з користувачем у навчальному програмному забезпеченні спрямовані на створення зручного, інтуїтивного та ефективного інтерфейсу. Реалізація цих алгоритмів дозволяє забезпечити інтерактивність, персоналізацію навчання та підтримку постійного зворотного зв'язку, що значно підвищує ефективність засвоєння знань.

2.4. Методи контролю знань учнів

Контроль знань учнів є важливим компонентом навчального процесу, що дозволяє оцінити рівень засвоєння матеріалу, виявити прогалини у знаннях та адаптувати освітній процес до потреб учнів. У навчальному програмному забезпеченні для стереометрії методи контролю знань базуються на сучасних цифрових технологіях, які забезпечують інтерактивність, автоматизацію перевірки та зворотний зв'язок.

Основні методи контролю знань

1. Тестування з множинним вибором

Цей метод передбачає, що учень вибирає правильні відповіді з кількох запропонованих варіантів.

Переваги:

- Автоматична перевірка відповідей.
- Швидке отримання результатів.
- Можливість включення запитань різного рівня складності.

Приклад завдання:

Яка площа основи правильної трикутної призми зі стороною основи $a=5$ см?

2. Завдання на заповнення пропусків

Учень повинен самостійно заповнити пропуски у тексті або формулі. Це допомагає перевірити знання теоретичного матеріалу та математичних формул.

Переваги:

- Перевірка знань без підказок.

- Розвиток навичок формулювання відповідей.

Приклад завдання:

Заповніть пропуски у формулі об'єму кулі: $V = \frac{4}{3}\pi R^3$ —

3. Практичні завдання на побудову

Цей метод передбачає виконання завдань з побудовою геометричних об'єктів, обчисленням параметрів або розв'язанням задач.

Переваги:

- Перевірка навичок практичного застосування знань.
- Можливість використання інтерактивних інструментів.

Приклад завдання:

Побудуйте правильну піраміду з основою квадрат зі стороною 4 см і висотою 6 см. Обчисліть її об'єм.

4. Інтерактивні тести з візуалізацією

Тести з використанням інтерактивної графіки дозволяють учням маніпулювати геометричними об'єктами, змінювати параметри та отримувати миттєвий зворотний зв'язок.

Переваги:

- Візуалізація полегшує розуміння складних понять.
- Інтерактивність підвищує мотивацію учнів.

Приклад завдання:

Обертайте піраміду, щоб знайти її правильну орієнтацію та визначити об'єм.

5. Самоперевірка

Режим самоперевірки дозволяє учням перевіряти свої знання без виставлення оцінки. Учень отримує рекомендації та підказки у разі помилок.

Переваги:

- Зниження рівня стресу перед іспитами.
- Заохочення самостійного навчання.

Приклад функції:

Розв'яжіть задачу самостійно. Натисніть «Перевірити», щоб отримати підказку або правильну відповідь.

6. Проектні завдання та практичні роботи

Проектні завдання передбачають виконання комплексних робіт, що об'єднують кілька тем. Учні створюють власні проекти, які перевіряються викладачами. [9]

Переваги:

- Розвиток аналітичних і творчих навичок.
- Можливість командної роботи.

Приклад проекту:

Створіть тривимірну модель правильного чотирикутного призми в програмі для моделювання. Представте її властивості та обчисліть об'єм і площу поверхні.

7. Підсумкові тести та контрольні роботи

Цей метод використовується для підсумкової перевірки знань з усього курсу або його окремих розділів.

Переваги:

- Повне оцінювання рівня знань учнів.
- Використання різних типів завдань.

Приклад контрольної роботи:

Обчисліть площу бічної поверхні циліндра з радіусом основи 5 см і висотою 10 см. Побудуйте його розгортку.

Зворотний зв'язок та оцінювання

Ефективний контроль знань передбачає надання зворотного зв'язку. Учні отримують докладний аналіз своїх помилок, пояснення правильних відповідей та рекомендації щодо покращення знань. Це сприяє ефективному засвоєнню матеріалу та підвищує рівень успішності.

Методи контролю знань учнів у навчальному програмному забезпеченні мають забезпечувати комплексну перевірку теоретичних і практичних навичок, зворотний зв'язок, підтримку різних типів тестів та інтерактивність. Застосування цих методів сприяє розвитку критичного мислення, вдосконаленню навичок самостійного навчання та формуванню стійких математичних компетенцій.

3. ПРАКТИЧНА ЧАСТИНА

3.1. Проектування програмного забезпечення

Проектування програмного забезпечення для навчального ресурсу з розділу «Стереометрія» передбачало створення веб-додатку з інтерактивними можливостями для вивчення геометричних об'єктів та математичних принципів. Основні етапи проектування включали визначення вимог до функціоналу, архітектурне проектування, створення інтерфейсу користувача, розробку алгоритмів обробки даних та побудову базових структур додатку.

Аналіз вимог до функціоналу

Основна мета розробки – створення інтерактивного веб-додатку з наступними можливостями:

- Візуалізація геометричних об'єктів у тривимірному просторі.
- Перегляд теоретичних матеріалів, розподілених по секціях.
- Виконання інтерактивних вправ і тестів.
- Зворотний зв'язок у вигляді підказок та оцінювання результатів.

Архітектурне проектування

Було обрано архітектуру клієнт-сервер із використанням сучасного фреймворку React.js для клієнтської частини. Це забезпечує динамічне оновлення вмісту сторінок без перезавантаження, зручне управління станом програми та можливість масштабування проекту в майбутньому.

Основні модулі клієнтської частини:

- Навігаційний модуль: відображає меню, забезпечує переходи між секціями.
- Теоретичний модуль: відображає інтерактивні сторінки з теоретичними матеріалами.
- Практичний модуль: надає вправи з автоматичним перевірянням.
- Модуль тестування: оцінює рівень знань користувача.

Проектування інтерфейсу користувача

Інтерфейс було спроектовано з урахуванням сучасних принципів UX/UI дизайну, включаючи:

- Інтуїтивну навігацію через бокову панель меню.
- Чітку візуальну ієрархію: заголовки, підзаголовки, текстові блоки.
- Візуалізацію математичних об'єктів: використання SVG-графіки та анімацій.
- Адаптивність: підтримка мобільних пристроїв та різних розмірів екранів.

Розробка алгоритмів обробки даних

Було створено спеціальні алгоритми для:

- Обчислення параметрів геометричних фігур (площі, об'єму, довжин сторін).
- Візуалізації фігур на основі заданих координат.
- Перевірки відповідей у тестах і надання зворотного зв'язку.

UML-діаграма системи

Для відображення загальної структури системи було створено UML-діаграму класів, яка демонструє взаємозв'язок основних компонентів додатку (див. рис. 3.1).

UML-діаграма класів системи

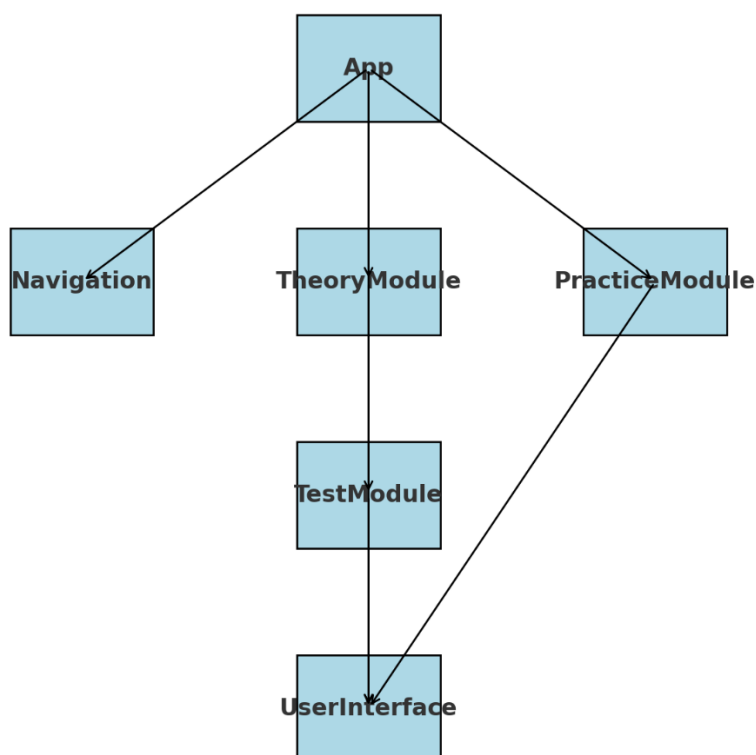


Рисунок 3.1 – UML-діаграма класів

UML-діаграма класів відображає основні компоненти системи: App,

Navigation, TheoryModule, PracticeModule, TestModule та UserInterface. Стрілки показують зв'язки між класами, підкреслюючи взаємодію між навігацією, теоретичними та практичними модулями та інтерфейсом користувача.

Діаграма демонструє структуру системи, забезпечуючи зрозумілу організацію основних функціональних елементів веб-додатку для вивчення стереометрії

3.2. Вибір технологій та архітектура системи

Розробка навчального програмного забезпечення з розділу «Стереометрія» базується на сучасних веб-технологіях, які забезпечують високу продуктивність, масштабованість та зручність у використанні. Основними обраними технологіями стали:

- React.js — бібліотека JavaScript для створення інтерфейсу користувача. Вона забезпечує компонентний підхід, швидку розробку та підтримку інтерактивних елементів.
- HTML5 і CSS3 — для створення структури та оформлення веб-сторінок. Використання адаптивних стилів та анімацій дозволило створити привабливий та сучасний інтерфейс.
- JavaScript (ES6+) — для написання основної логіки додатка, створення інтерактивних елементів та управління станом компонентів.
- React Router — для управління маршрутизацією між сторінками додатка.
- KaTeX — для відображення математичних формул, необхідних для пояснення теоретичного матеріалу.
- SVG і Canvas — для побудови графічних ілюстрацій геометричних об'єктів.

Архітектура системи

Архітектура розробленого програмного забезпечення базується на компонентній моделі React.js. Додаток поділено на окремі компоненти, які відповідають за різні функції, включаючи відображення теоретичного матеріалу, тестових завдань та інтерактивних вправ.

Основні компоненти системи:

- Header — компонент для відображення заголовку та навігаційного меню.
- Sidebar — бокова панель для навігації між темами.
- Theory — компонент, який містить теоретичні розділи з інтерактивними ілюстраціями.
- Practice — модуль для інтерактивних вправ і тестів.
- Footer — підвал сайту з додатковою інформацією.

Фізична структура проекту зображена на рисунку 3.2 (див. рис. 3.2)

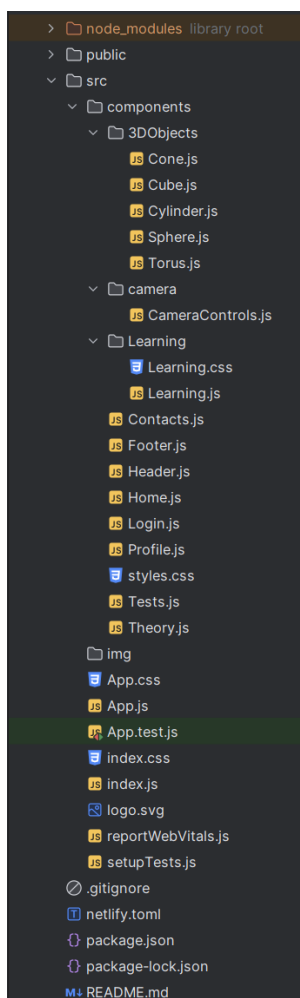


Рисунок 3.2 – Фізична структура

3.3. Реалізація основних модулів

Реалізація програмного забезпечення для вивчення стереометрії включала створення основних модулів, які забезпечують функціональність веб-додатка. У цьому розділі розглянемо ключові модулі з поясненням їх реалізації та прикладами

коду.

1. Модуль теоретичного матеріалу (Theory.js)

Функціональність: Відображення теоретичних розділів із підтримкою математичних формул та інтерактивних ілюстрацій.

Основні частини коду:

```
import React from 'react';
import { BlockMath, InlineMath } from 'react-katex';
import 'katex/dist/katex.min.css';
```

```
function Theory() {
```

```
  return (
```

```
    <main className="theory-container">
```

```
      <h1>Теорія Стереометрії</h1>
```

```
      <div id="axioms" className="topic">
```

```
        <h2>Основні аксіоми стереометрії</h2>
```

```
        <p>
```

```
          Стереометрія базується на трьох основних аксіомах:
```

```
        </p>
```

```
        <ul>
```

```
          <li><strong>Аксіома 1:</strong> Через будь-які три точки, які не лежать на одній
          прямій, проходить одна площина.</li>
```

```
          <li><strong>Аксіома 2:</strong> Якщо дві точки прямої лежать у площині, то всі
          точки цієї прямої лежать у площині.</li>
```

```
          <li><strong>Аксіома 3:</strong> Якщо дві площини перетинаються, то вони
          перетинаються по прямій.</li>
```

```
        </ul>
```

```
        <BlockMath math="Ax + By + Cz + D = 0" />
```

```
      </div>
```

```
    </main>
```

```
  );
```

```
}
```

```
export default Theory;
```

Пояснення:

- Використовується компонент Theory для відображення теоретичного матеріалу.
- Використано KaTeX для математичних формул.
- Теоретичні розділи організовані за допомогою структурованих тегів div, h2, ul, li.

2. Модуль тестування знань (Practice.js)

Функціональність: Відображення тестових завдань з інтерактивними відповідями.

Основні частини коду:

```
import React, { useState } from 'react';
```

```
function Practice() {
```

```
  const [selectedAnswer, setSelectedAnswer] = useState(null);
```

```
  const [isCorrect, setIsCorrect] = useState(false);
```

```
  const handleAnswer = (answer) => {
```

```
    setSelectedAnswer(answer);
```

```
    setIsCorrect(answer === 'A');
```

```
  };
```

```
  return (
```

```
    <div className="practice-container">
```

```
      <h2>Тест: Об'єм кулі</h2>
```

```
      <p>Знайти об'єм кулі з радіусом 5 см.</p>
```

```
      <div className="answer-options">
```

```
        <button onClick={() => handleAnswer('A')}>523.6 см3</button>
```

```
        <button onClick={() => handleAnswer('B')}>314.2 см3</button>
```

```
      </div>
```

```
      {selectedAnswer && (
```

```
        <div className={`result ${isCorrect ? 'correct' : 'incorrect'}`}>
```

```
          {isCorrect ? 'Правильно!' : 'Неправильно, спробуйте ще раз.'}
```

```
        </div>
```

```
      )}
```

```
    </div>
```

```
);
}
```

```
export default Practice;
```

Пояснення:

- Створено компонент Practice для тестування знань.
- Реалізовано обробку натискань на відповіді за допомогою onClick.
- Використовується useState для збереження обраної відповіді та відображення результату.

3. Модуль навігації (Sidebar.js)

Функціональність: Відображення бокового меню для переходу між розділами.

Основні частини коду:

```
import React from 'react';
```

```
function Sidebar() {
  return (
    <aside className="sidebar">
      <h2>Навігація</h2>
      <nav>
        <a href="#axioms">Основні аксіоми</a>
        <a href="#distance">Відстань між точками</a>
        <a href="#sphere">Об'єм та площа кулі</a>
        <a href="#tests">Тести</a>
      </nav>
    </aside>
  );
}
```

```
export default Sidebar;
```

Пояснення:

- Навігація створена за допомогою a href, які ведуть до відповідних розділів.
- Оформлення виконане за допомогою стилів CSS.

Реалізація основних модулів програми базується на принципах компонентного підходу React.js, з використанням інтерактивних елементів та

модульної структури. Це дозволяє легко масштабувати проект, додавати нові функції та забезпечувати зручну взаємодію з користувачем.

3.4. Контроль версій та розгортання сайту в інтернеті

Розробка навчального програмного забезпечення з розділу «Стереометрія» передбачала використання сучасних інструментів для контролю версій, спільної розробки та розгортання проекту в інтернеті. Для цих цілей було обрано GitHub як платформу для зберігання вихідного коду та управління версіями, а також Netlify для автоматичного розгортання веб-додатка.

Контроль версій з GitHub

GitHub — це відома система контролю версій, яка підтримує Git та забезпечує можливості для зберігання, відстеження змін і спільної роботи над проектом. Основні етапи роботи включали:

1. Ініціалізація репозиторію

На початку розробки проекту був створений новий репозиторій на GitHub.

Виконані дії:

- Ініціалізація проекту React.js командою:

```
npx create-react-app stereometry
```

- Створення репозиторію:

```
git init
```

```
git remote add origin https://github.com/user/stereometry.git
```

2. Основні команди Git

Під час розробки використовувались базові команди для контролю версій:

- Додавання файлів до коміту:

```
git add .
```

- Створення коміту з описом змін:

```
git commit -m "Додав початкову версію сайту"
```

- Відправлення змін на віддалений репозиторій:

```
git push origin main
```

3. Робота з гілками

- Створення нової гілки для розробки нової функції:

`git checkout -b feature/sidebar-navigation`

- Злиття гілок після тестування:

`git merge feature/sidebar-navigation`

Розгортання сайту з Netlify

Для забезпечення безперервного розгортання було використано платформу Netlify, яка пропонує автоматичне розгортання при кожному оновленні коду в GitHub. Основні кроки:

1. Налаштування розгортання

- Створення проекту на Netlify.
- Підключення репозиторію з GitHub:
 - Вибір потрібного репозиторію.
 - Встановлення основної гілки (main) для автоматичного розгортання.

2. Автоматичне розгортання

- Процес розгортання: Netlify автоматично відстежує нові коміти в репозиторії та виконує наступні дії:
 - Завантаження коду з GitHub.
 - Запуск команд для побудови проекту:

`npm install`

`npm run build`

- Публікація сайту на наданій Netlify URL-адресі.

Завдяки використанню GitHub та Netlify проект залишався актуальним та стабільним на всіх етапах розробки. GitHub забезпечив надійний контроль версій, дозволяючи швидко відслідковувати зміни та виправляти помилки. Netlify надав можливість автоматичного розгортання проекту в інтернеті, що дозволило оперативно тестувати сайт та ділитися результатами роботи з викладачами та одногрупниками.

3.5. Інструкція для користувача

1. Головна сторінка сайту (див. рис. 3.3)

1.1. Основний інтерфейс

- Після переходу на сайт ви потрапляєте на головну сторінку.
- Головна сторінка включає:
 - Логотип платформи “Стереометрія”.
 - Привітальне повідомлення: *"Стереометрія: легко про складне"*.
 - Основну кнопку “Почати навчання”, яка веде до теоретичного матеріалу.
 - Три основні блоки з функціоналом:
 - Теорія — детальні пояснення складних тем з ілюстраціями.
 - Візуалізація — об’ємні фігури для інтерактивного навчання.
 - Тести — перевірка знань за допомогою адаптивного тестування.

Навігація

- Верхнє меню дозволяє швидко переходити між основними сторінками: "Навчання", "Теорія", "Тести", "Контакти".

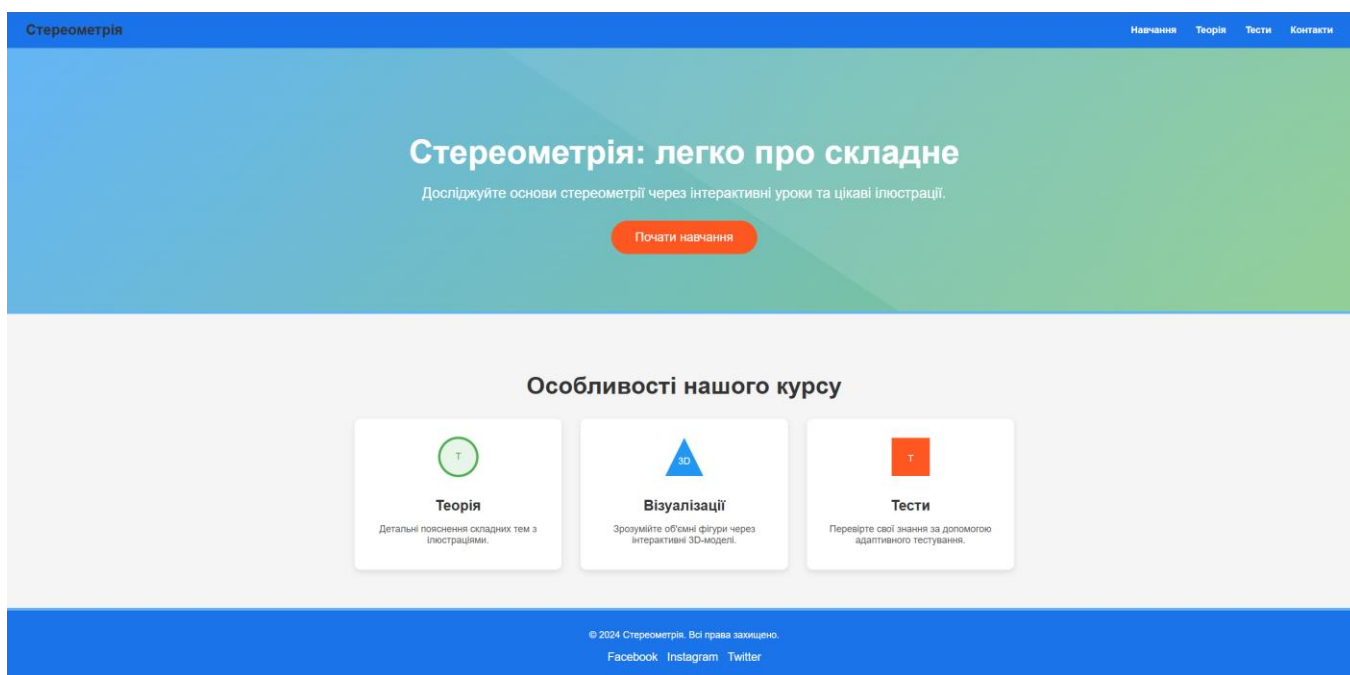


Рисунок 3.3 – Головна сторінка сайту

2. Розділ "Теорія" (див. рис. 3.4)

2.1. Доступ до теоретичних матеріалів

- Для переходу до теорії натисніть кнопку “Теорія” у верхньому меню.
- На сторінці теорії представлені тематичні блоки:
 - Основні аксіоми стереометрії.
 - Відстань між точками у просторі.
 - Об'єм та площа поверхні кулі.
 - Інші важливі теми.

2.2. Структура сторінки теорії

- Зліва розташована бічна панель навігації, що дозволяє переходити між розділами теорії.
- Справа — блок з детальним описом теми, що включає текст, формули та інтерактивні ілюстрації.

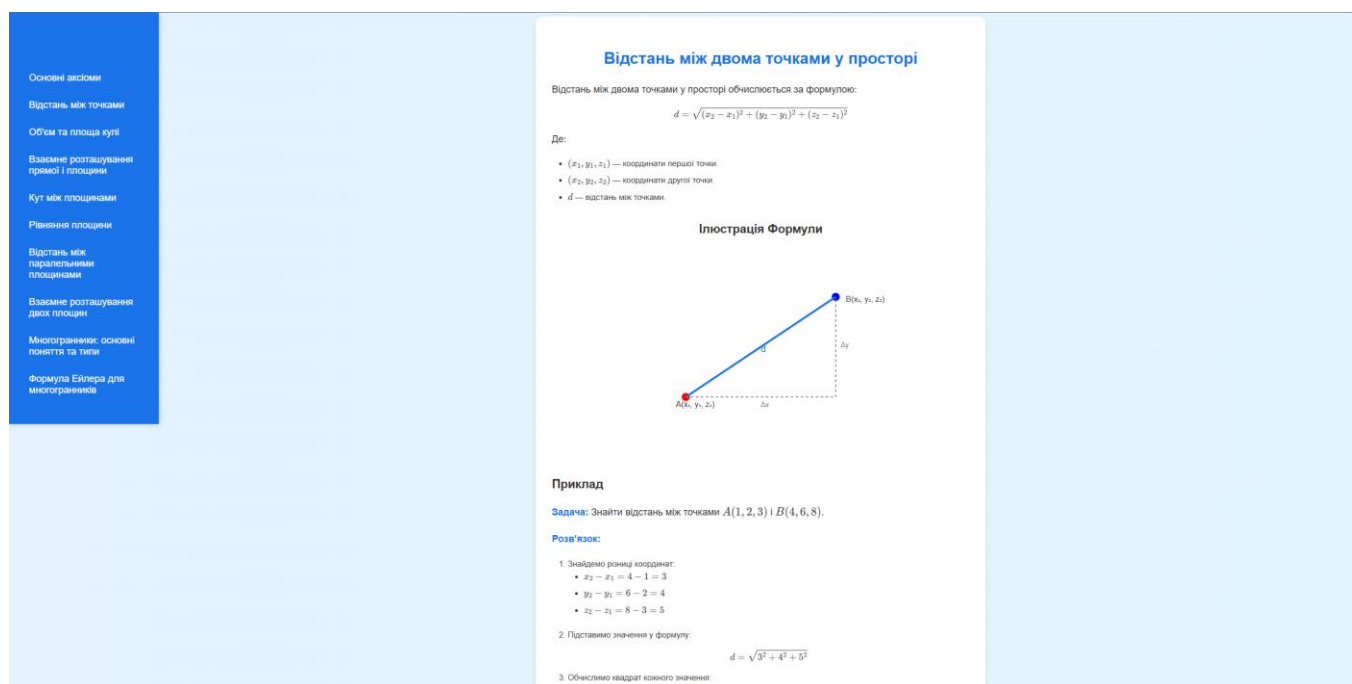


Рисунок 3.4 – Розділ "Теорія"

3. Розділ "Тести"

3.1. Процес тестування (див. рис. 3.5)

- Система пропонує питання у форматі тестових завдань.

- Питання містять:
 - Текст задачі.
 - Кілька варіантів відповідей.
- Після вибору відповіді результат відображається миттєво:
 - Зелений колір — правильна відповідь.
 - Червоний колір — неправильна відповідь.

3.2. Завершення тестування (див. рис. 3.6)

- Після завершення тестування користувач отримує результат:
 - Кількість правильних відповідей.
 - Кнопку для повторного проходження тесту.

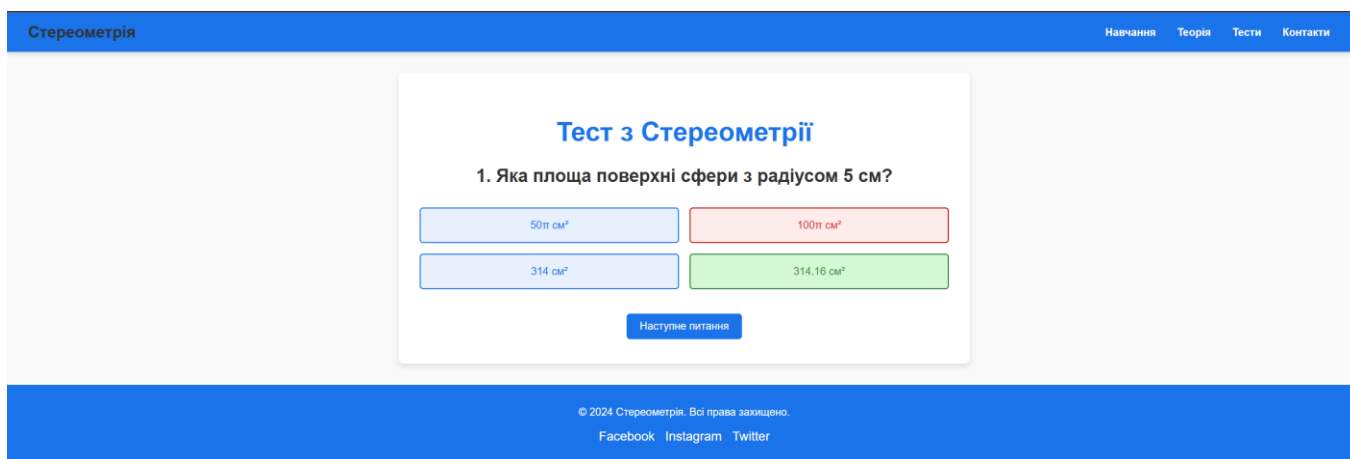


Рисунок 3.5 – Процес тестування

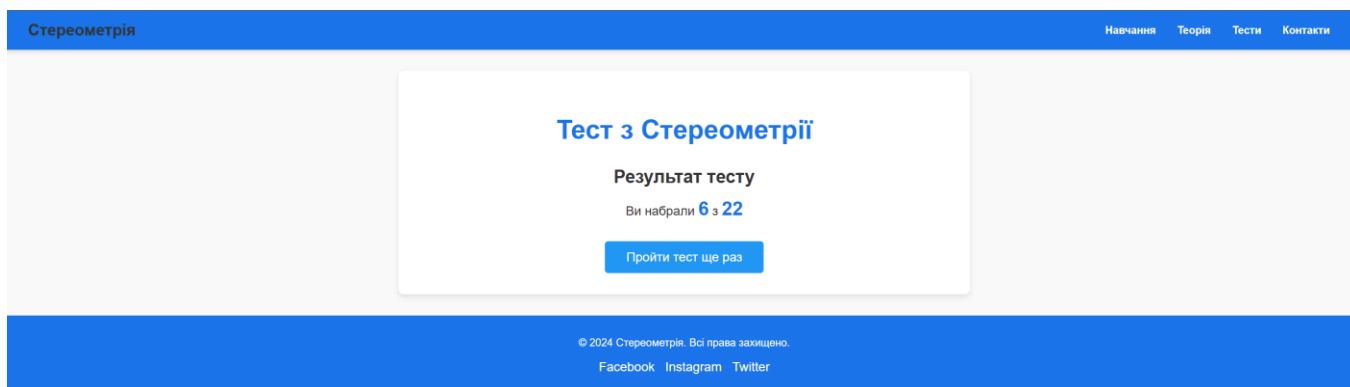


Рисунок 3.6 – Результат тестування

4. Розділ "Контакти" (див. рис. 3.7)

4.1. Загальна інформація

- Розділ “Контакти” надає можливість зв'язатися з адміністраторами платформи:
 - Email: support@example.com
 - Телефон: +38 (067) 123-45-67
 - Адреса: вул. Центральна, 1, Київ, Україна.

4.2. Форма зворотного зв'язку

- Користувач може надіслати повідомлення через форму:
 - Ім'я.
 - Email.
 - Повідомлення.
- Після заповнення натисніть кнопку “Надіслати”.

Стереометрія

Надання Теорія Тести Контакти

Контакти

Масте питання? Ми завжди раді допомогти!

Наші контакти

Телефон: +38 (067) 123-45-67
 Email: support@example.com
 Адреса: вул. Центральна, 1, Київ, Україна

Зв'яжіться з нами

Ім'я
 Ваше ім'я

Email
 Ваш email

Повідомлення
 Ваше повідомлення

Надіслати

Наша команда

150 x 150
Ім'я Прізвище
Посада: Розробник

150 x 150
Ім'я Прізвище
Посада: Дизайнер

150 x 150
Ім'я Прізвище
Посада: Менеджер

© 2024 Стереометрія. Всі права захищено.
 Facebook Instagram Twitter

Рисунок 3.7 – Розділ "Контакти"

5. Розділ "Візуалізація"

5.1. Вибір геометричного об'єкта (див. рис. 3.8)

- На сторінці візуалізації користувач може обрати тривимірний об'єкт для

перегляду:

- Куб.
- Конус.
- Інші фігури.

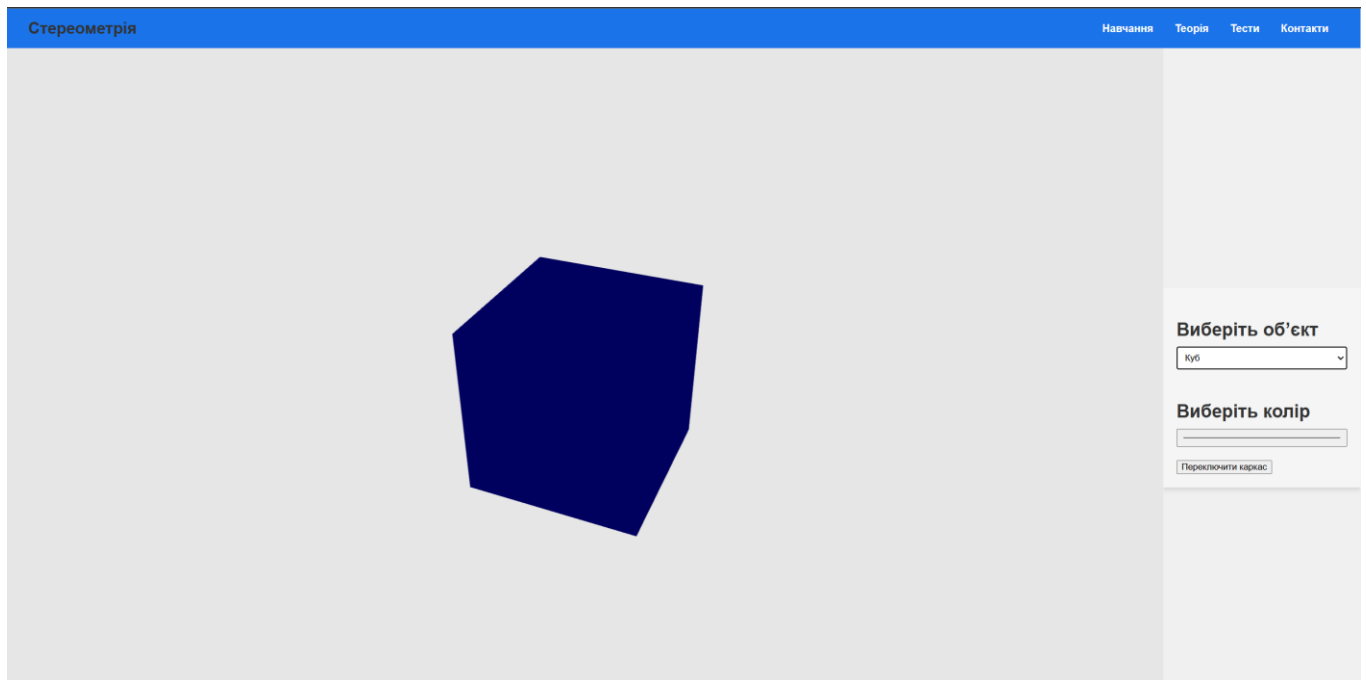


Рисунок 3.8 – Вибір об'єкта

5.2. Налаштування коліру (див. рис. 3.9)

- Доступні функції:
 - Вибір кольору фігури через колірний інтерфейс.

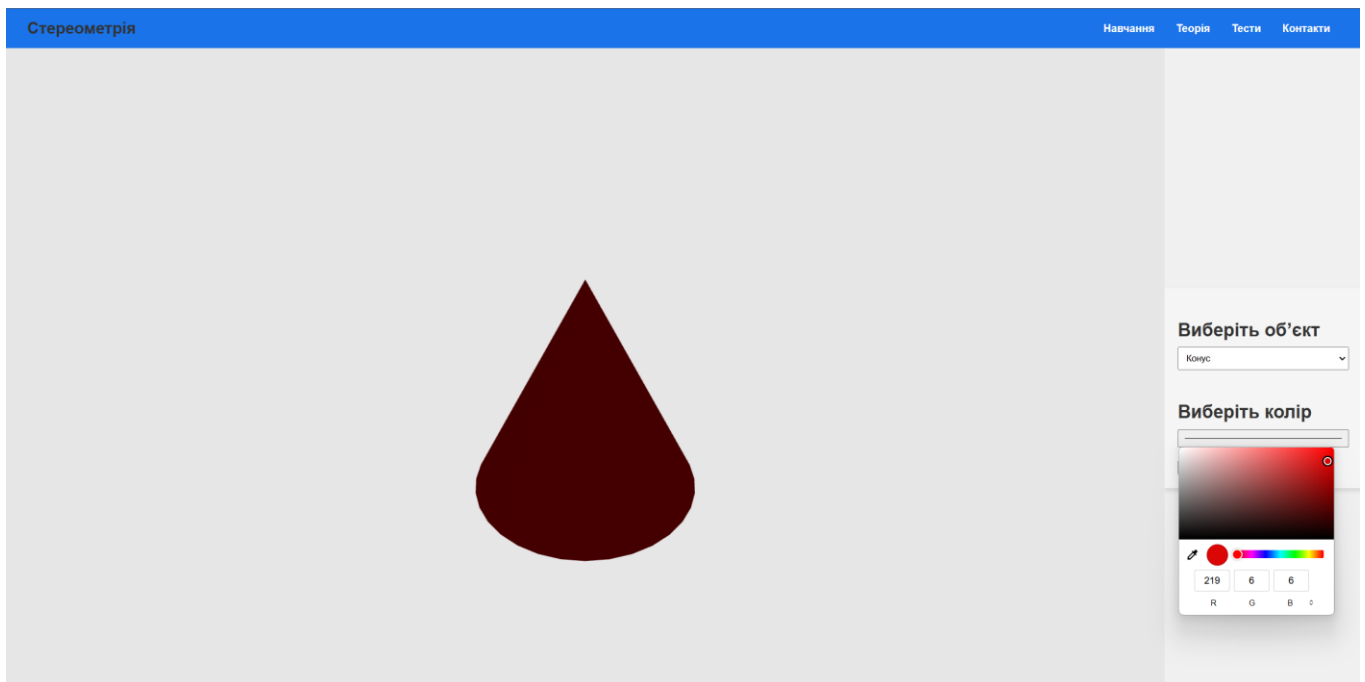


Рисунок 3.9 – Вибір коліру об'єкта

5.3. Взаємодія

- Об'єкт можна обертати та досліджувати у тривимірному просторі.

5.4. Демонстрація переключення каркасу

- Після активації опції "Переключити каркас", фігура змінює свій вигляд:
 - Грані стають прозорими, а ребра відображаються у вигляді ліній (див. рис. 3.10).
 - Це дозволяє краще зрозуміти структуру об'єкта, що є корисним для навчання та візуалізації геометричних властивостей фігур.

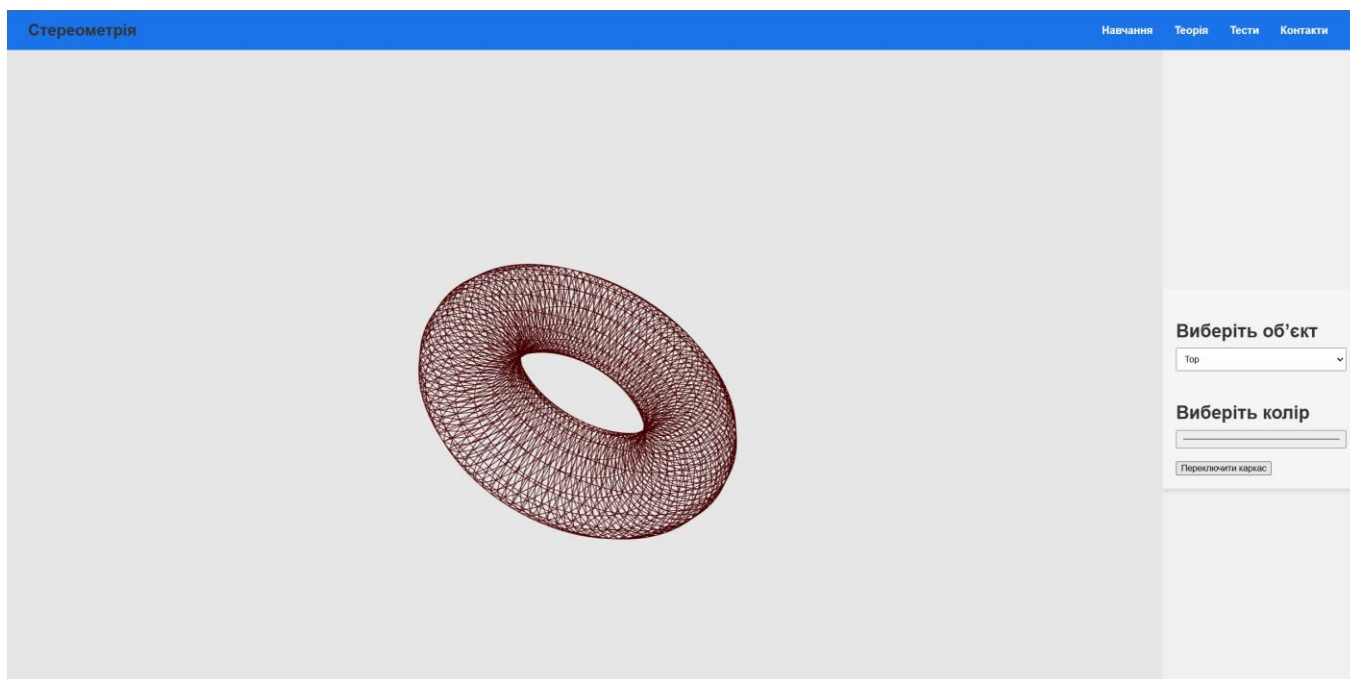


Рисунок 3.10 – Переключення каркасу

Таким чином, платформа “Стереометрія” є інтуїтивно зрозумілим і зручним навчальним інструментом для вивчення геометрії. Вона дозволяє легко отримати доступ до навчальних матеріалів, проходити тести для контролю знань і візуалізувати геометричні об'єкти.

ВИСНОВКИ

У ході виконання дипломної роботи було успішно розроблено інтерактивне навчальне програмне забезпечення з розділу "Стереометрія" шкільного курсу математики, яке поєднує в собі структуровані теоретичні матеріали, візуалізацію тривимірних фігур та можливості тестування знань. Проєкт спрямований на підвищення інтересу учнів до стереометрії та полегшення розуміння складних геометричних концепцій завдяки інтерактивному підходу.

Основні результати роботи включають:

1. Аналіз сучасних технологій та вибір інструментів для розробки:
 - Проведено аналіз існуючих програмних рішень для вивчення стереометрії.
 - Для реалізації фронтенду використано React.js, що забезпечило гнучкість та зручність розробки.
 - Застосовано Netlify для розгортання проєкту в інтернеті, що спростило процес публікації та забезпечило стабільність роботи.
 - Використання GitHub дозволило організувати контроль версій та спільну роботу над проєктом.
2. Проєктування архітектури та інтерфейсу:
 - Розроблено зручний та інтуїтивно зрозумілий інтерфейс, що включає головне меню, бічну панель для навігації по теорії та інтерактивні блоки.
 - Створено адаптивний дизайн, який забезпечує коректне відображення на різних пристроях — від комп'ютерів до мобільних телефонів.
 - Реалізовано логічну структуру сайту з розділами: "Теорія", "Навчання", "Тести" та "Контакти".
3. Реалізація функціональних модулів:
 - Розділ "Теорія" містить структуровані навчальні матеріали з темами стереометрії: аксіоми, рівняння площини, об'єм та площа фігур тощо.
 - Візуалізація 3D-фігур включає можливість вибору об'єкта, зміни кольору та перемикання на каркасний режим для кращого розуміння структури геометричних об'єктів.

- Реалізовано розділ "Тести" для перевірки знань, що забезпечує інтерактивну взаємодію з користувачем та надає зворотний зв'язок (правильна/неправильна відповідь).
- Створено функціонал для обчислень за геометричними формулами (приклади задач із розв'язками та поясненнями).

4. Розгортання платформи:

- Використано Netlify для публікації платформи в інтернеті з автоматичним деплоєм після оновлень на GitHub.
- Забезпечено легкий доступ до проєкту для учнів та вчителів за допомогою наданого посилання.

Розроблене програмне забезпечення дозволяє учням:

- Засвоювати теоретичні знання за допомогою структурованих ілюстрацій та прикладів.
- Досліджувати тривимірні фігури через інтерактивну візуалізацію.
- Тестувати свої знання у вигляді адаптивних завдань з миттєвим зворотним зв'язком.

У майбутньому проєкт може бути розширений завдяки:

- Додаванню нових тем з інших розділів математики, таких як планіметрія чи алгебра.
- Інтеграції відеоуроків та симуляцій для глибшого пояснення складних тем.
- Створенню статистики прогресу учнів для вчителів.
- Додатковій гейміфікації: впровадженню нагород, досягнень та рейтингової системи.

Розробка підтвердила ефективність використання інтерактивних технологій для навчання стереометрії, відкривши нові можливості для покращення якості викладання та мотивації учнів до вивчення математики.

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Бевз Г. П., Бевз В. Г. Геометрія: Підручник для 10-11 класів. — Київ: Генеза, 2021.
2. Матвієнко А. П., Коваленко В. А. Візуалізація математичних понять: Інтерактивні підходи. — Харків: Видавництво ХНУ, 2020.
3. Литвиненко О. А. Використання ІКТ у викладанні математики. — Львів: Видавництво ЛНУ, 2020.
4. Іваненко Т. В. Інтерактивні методи навчання в математиці. — Київ: Освіта, 2021.
5. Васильєв О. М. Програмування. Bohdan Books, 2022.
6. React.js Documentation. React — A JavaScript library for building user interfaces. Інтернет-доступ: <https://legacy.reactjs.org/>
7. Mozilla Developer Network (MDN). Introduction to HTML and CSS. Інтернет-доступ: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML
8. Netlify Documentation. Deploy Modern Web Projects. Інтернет-доступ: <https://docs.netlify.com/>
9. Буряк О. П. Алгебра і початки аналізу: Підручник для 10-11 класів. — Київ: Освіта, 2021.
10. Лобачевський М. І. Тригонометрія: Теорія та практичні задачі. — Харків: Видавництво ХНУ, 2020.
11. Ольховська О. В. Методичні рекомендації до виконання кваліфікаційної роботи для студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра / О. В. Ольховська, О. О. Черненко. — Полтава : ПУЕТ, 2022. — 67 с. — 1 електрон. опт. диск (CVD-ROM).

ДОДАТОК А.

Home.js

```
import React from 'react';
import { useNavigate } from 'react-router-dom';
import './styles.css';

function Home() {
  const navigate = useNavigate();

  const handleStartLearning = () => {
    navigate('/theory'); // Перехід одразу на сторінку теорії
  };

  return (
    <>
      <section className="hero">
        <div className="hero-content">
          <h1 className="hero-title">Стереометрія: легко про складне</h1>
          <p className="hero-subtitle">
            Досліджуйте основи стереометрії через інтерактивні уроки та цікаві ілюстрації.
          </p>
          <button className="primary-btn" onClick={handleStartLearning}>
            Почати навчання
          </button>
        </div>
      </section>

      <section className="features">
        <h2>Особливості нашого курсу</h2>
        <div className="features-grid">
          <div className="feature-card">
            <div className="feature-icon">
              <svg width="80" height="80" viewBox="0 0 100 100">
                <circle cx="50" cy="50" r="40" stroke="#4caf50" strokeWidth="5" fill="#e8f5e9" />
                <text x="50" y="55" textAnchor="middle" fontSize="20" fill="#4caf50">T</text>
              </svg>
            </div>
          </div>
        </div>
      </section>
    </>
  );
}
```

```

    <h3>Теорія</h3>
    <p>Детальні пояснення складних тем з ілюстраціями.</p>
  </div>

  <div className="feature-card">
    <div className="feature-icon">
      <svg width="80" height="80" viewBox="0 0 100 100">
        <polygon points="50,10 90,90 10,90" fill="#2196f3" />
        <text x="50" y="65" textAnchor="middle" fontSize="20" fill="ffffff">3D</text>
      </svg>
    </div>
    <h3>Візуалізації</h3>
    <p>Зрозумійте об'ємні фігури через інтерактивні 3D-моделі.</p>
  </div>

  <div className="feature-card">
    <div className="feature-icon">
      <svg width="80" height="80" viewBox="0 0 100 100">
        <rect x="10" y="10" width="80" height="80" fill="#ff5722" />
        <text x="50" y="60" textAnchor="middle" fontSize="20" fill="ffffff">T</text>
      </svg>
    </div>
    <h3>Тести</h3>
    <p>Перевірте свої знання за допомогою адаптивного тестування.</p>
  </div>
</div>
</section>
</>
);
}

```

```
export default Home;
```

Learning.js

```

import React, { useState } from "react";
import { Canvas } from "@react-three/fiber";
import Cube from "../3DObjects/Cube";
import Sphere from "../3DObjects/Sphere";
import Cylinder from "../3DObjects/Cylinder";

```

```

import Cone from "../3DObjects/Cone";
import Torus from "../3DObjects/Torus";
import { OrbitControls } from "@react-three/drei";
import "../Learning.css";

const Learning = () => {
  const [selectedObject, setSelectedObject] = useState("Cube");
  const [isWireframe, setWireframe] = useState(false);
  const [color, setColor] = useState("#0000ff");

  const renderObject = () => {
    switch (selectedObject) {
      case "Cube":
        return <Cube args={[3, 3, 3]} color={color} isWireframe={isWireframe} />;
      case "Sphere":
        return <Sphere args={[2, 32, 32]} color={color} isWireframe={isWireframe} />;
      case "Cylinder":
        return <Cylinder args={[2, 2, 4, 32]} color={color} isWireframe={isWireframe} />;
      case "Cone":
        return <Cone args={[2, 4, 32]} color={color} isWireframe={isWireframe} />;
      case "Torus":
        return <Torus args={[2, 0.8, 16, 100]} color={color} isWireframe={isWireframe} />;
      default:
        return null;
    }
  };

  return (
    <div className="learning-container">
      <Canvas shadows camera={{ position: [5, 5, 10], fov: 50 }}>
        <ambientLight intensity={0.4} />
        <pointLight position={[10, 10, 10]} intensity={1} castShadow />
        <OrbitControls enablePan enableZoom enableRotate />
        {renderObject()}
      </Canvas>

      <div className="controls-panel">
        <h3>Выберите объект</h3>

```

```

    <select
      value={selectedObject}
      onChange={(e) => setSelectedObject(e.target.value)}
    >
      <option value="Cube">Куб</option>
      <option value="Sphere">Сфера</option>
      <option value="Cylinder">Циліндр</option>
      <option value="Cone">Конус</option>
      <option value="Torus">Тор</option>
    </select>

    <h3>Виберіть колір</h3>
    <input
      type="color"
      value={color}
      onChange={(e) => setColor(e.target.value)}
    />

    <button onClick={() => setWireframe(!isWireframe)}>
      Переключити каркас
    </button>
  </div>
</div>
);
};

```

export default Learning;

Contacts.js

```

import React, { useState } from 'react';
import './styles.css';

function Contacts() {
  const [formData, setFormData] = useState({
    name: "",
    email: "",
    message: "",
  });
}

```

```
const [submitted, setSubmitted] = useState(false);
```

```
const handleInputChange = (e) => {
  const { name, value } = e.target;
  setFormData({
    ...formData,
    [name]: value,
  });
};
```

```
const handleSubmit = (e) => {
  e.preventDefault();
  // Тут ви можете додати інтеграцію з сервером для відправки даних
  console.log('Submitted Data:', formData);
  setSubmitted(true);
};
```

```
return (
  <main className="contacts-page">
    <section className="contact-header">
      <h1>Контакти</h1>
      <p>Маєте питання? Ми завжди раді допомогти!</p>
    </section>

    <section className="contact-info">
      <h2>Наші контакти</h2>
      <ul>
        <li>
          <strong>Телефон:</strong> +38 (067) 123-45-67
        </li>
        <li>
          <strong>Email:</strong> support@example.com
        </li>
        <li>
          <strong>Адреса:</strong> вул. Центральна, 1, Київ, Україна
        </li>
      </ul>
    </section>
  )
```

```

<section className="contact-form">
  <h2>Зв'яжіться з нами</h2>
  {submitted ? (
    <div className="success-message">
      <p>Дякуємо за ваш запит! Ми зв'яжемося з вами найближчим часом.</p>
    </div>
  ) : (
    <form onSubmit={handleSubmit}>
      <div className="form-group">
        <label htmlFor="name">Ім'я</label>
        <input
          type="text"
          id="name"
          name="name"
          value={formData.name}
          onChange={handleInputChange}
          required
          placeholder="Ваше ім'я"
        />
      </div>
      <div className="form-group">
        <label htmlFor="email">Email</label>
        <input
          type="email"
          id="email"
          name="email"
          value={formData.email}
          onChange={handleInputChange}
          required
          placeholder="Ваш email"
        />
      </div>
      <div className="form-group">
        <label htmlFor="message">Повідомлення</label>
        <textarea
          id="message"
          name="message"

```

```

        value={formData.message}
        onChange={handleInputChange}
        required
        placeholder="Ваше повідомлення"
      />
    </div>
    <button type="submit" className="submit-btn">
      Надіслати
    </button>
  </form>
)}
</section>

```

```

<section className="team">
  <h2>Наша команда</h2>
  <div className="team-grid">
    <div className="team-member">
      
      <h3>Ім'я Прізвище</h3>
      <p>Посада: Розробник</p>
    </div>
    <div className="team-member">
      
      <h3>Ім'я Прізвище</h3>
      <p>Посада: Дизайнер</p>
    </div>
    <div className="team-member">
      
      <h3>Ім'я Прізвище</h3>
      <p>Посада: Менеджер</p>
    </div>
  </div>
</section>
</main>
);
}

```

export default Contacts;

Tests.js

```

import React, { useState } from "react";
import "./styles.css";

function Test() {
  const questions = [
    {
      id: 1,
      question: "Яка площа поверхні сфери з радіусом 5 см?",
      options: [
        { id: "a", text: "50π см²", correct: false },
        { id: "b", text: "100π см²", correct: false },
        { id: "c", text: "314 см²", correct: false },
        { id: "d", text: "314.16 см²", correct: true }
      ]
    },
    {
      id: 2,
      question: "Скільки вершин має куб?",
      options: [
        { id: "a", text: "6", correct: false },
        { id: "b", text: "8", correct: true },
        { id: "c", text: "10", correct: false },
        { id: "d", text: "12", correct: false }
      ]
    }
  ],

```

```

{
  id: 3,
  question: "Яке рівняння площини, що проходить через точку (1, 2, 3) з нормальним вектором (2, -1, 4)?",
  options: [
    { id: "a", text: " $2x - y + 4z - 12 = 0$ ", correct: true },
    { id: "b", text: " $x + y - z = 0$ ", correct: false },
    { id: "c", text: " $2x + y + 3z = 0$ ", correct: false },
    { id: "d", text: " $x - y + 2z = 0$ ", correct: false }
  ]
},
{
  id: 4,
  question: "Як визначити відстань між точками (2, -1, 3) та (4, 6, 8)?",
  options: [
    { id: "a", text: " $\sqrt{50}$ ", correct: false },
    { id: "b", text: " $\sqrt{86}$ ", correct: false },
    { id: "c", text: " $\sqrt{89}$ ", correct: true },
    { id: "d", text: " $\sqrt{100}$ ", correct: false }
  ]
},
{
  id: 5,
  question: "Що є нормальним вектором для площини  $x + 2y - z + 4 = 0$ ?",
  options: [
    { id: "a", text: "(1, 2, -1)", correct: true },
    { id: "b", text: "(2, -1, 4)", correct: false },
    { id: "c", text: "(1, -2, 4)", correct: false },
    { id: "d", text: "(2, 1, -4)", correct: false }
  ]
},
{
  id: 6,
  question: "Яка формула обчислення об'єму кулі з радіусом R?",
  options: [
    { id: "a", text: " $V = 4/3 \pi R^2$ ", correct: false },
    { id: "b", text: " $V = \pi R^2$ ", correct: false },
    { id: "c", text: " $V = 4/3 \pi R^3$ ", correct: true },

```

```

    { id: "d", text: " $V = 2\pi R^3$ ", correct: false }
  ]
},
{
  id: 7,
  question: "Скільки граней має паралелепіпед?",
  options: [
    { id: "a", text: "4", correct: false },
    { id: "b", text: "6", correct: true },
    { id: "c", text: "8", correct: false },
    { id: "d", text: "12", correct: false }
  ]
},
{
  id: 8,
  question: "Який кут між векторами (1, 0, 0) та (0, 1, 0)?",
  options: [
    { id: "a", text: "0°", correct: false },
    { id: "b", text: "45°", correct: false },
    { id: "c", text: "90°", correct: true },
    { id: "d", text: "180°", correct: false }
  ]
},
{
  id: 9,
  question: "Що описує формула  $d = |Ax_1 + By_1 + Cz_1 + D| / \sqrt{A^2 + B^2 + C^2}$ ?",
  options: [
    { id: "a", text: "Відстань між двома точками", correct: false },
    { id: "b", text: "Відстань від точки до площини", correct: true },
    { id: "c", text: "Довжина вектора", correct: false },
    { id: "d", text: "Кут між площинами", correct: false }
  ]
},
{
  id: 10,
  question: "Як виглядає рівняння площини в просторі?",
  options: [
    { id: "a", text: " $Ax + By + Cz + D = 0$ ", correct: true },

```

```

    { id: "b", text: " $Ax^2 + By^2 + Cz^2 + D = 0$ ", correct: false },
    { id: "c", text: " $Ax + By + Cz = 0$ ", correct: false },
    { id: "d", text: " $Ax - By - Cz = 0$ ", correct: false }
  ]
},
{
  id: 11,
  question: "Яка площа поверхні сфери з радіусом 5 см?",
  options: [
    { id: "a", text: " $50\pi \text{ см}^2$ ", correct: false },
    { id: "b", text: " $100\pi \text{ см}^2$ ", correct: false },
    { id: "c", text: " $314 \text{ см}^2$ ", correct: false },
    { id: "d", text: " $314.16 \text{ см}^2$ ", correct: true }
  ]
},
{
  id: 12,
  question: "Скільки вершин має куб?",
  options: [
    { id: "a", text: "6", correct: false },
    { id: "b", text: "8", correct: true },
    { id: "c", text: "10", correct: false },
    { id: "d", text: "12", correct: false }
  ]
},
{
  id: 13,
  question: "Яке рівняння площини через точку (1, 2, 3) з нормальним вектором (2, -1, 4)?",
  options: [
    { id: "a", text: " $2x - y + 4z - 12 = 0$ ", correct: true },
    { id: "b", text: " $x + y - z = 0$ ", correct: false },
    { id: "c", text: " $2x + y + 3z = 0$ ", correct: false },
    { id: "d", text: " $x - y + 2z = 0$ ", correct: false }
  ]
},
{
  id: 14,
  question: "Як обчислити відстань між точками (2, -1, 3) та (4, 6, 8)?",

```

```

options: [
  { id: "a", text: " $\sqrt{50}$ ", correct: false },
  { id: "b", text: " $\sqrt{86}$ ", correct: false },
  { id: "c", text: " $\sqrt{89}$ ", correct: true },
  { id: "d", text: " $\sqrt{100}$ ", correct: false }
]
},
{
  id: 15,
  question: "Що є нормальним вектором для площини  $x + 2y - z + 4 = 0$ ?",
  options: [
    { id: "a", text: "(1, 2, -1)", correct: true },
    { id: "b", text: "(2, -1, 4)", correct: false },
    { id: "c", text: "(1, -2, 4)", correct: false },
    { id: "d", text: "(2, 1, -4)", correct: false }
  ]
},
{
  id: 16,
  question: "Яка формула об'єму кулі з радіусом  $R$ ?",
  options: [
    { id: "a", text: " $V = 4/3 \pi R^2$ ", correct: false },
    { id: "b", text: " $V = \pi R^2$ ", correct: false },
    { id: "c", text: " $V = 4/3 \pi R^3$ ", correct: true },
    { id: "d", text: " $V = 2\pi R^3$ ", correct: false }
  ]
},
{
  id: 17,
  question: "Скільки граней має паралелепіпед?",
  options: [
    { id: "a", text: "4", correct: false },
    { id: "b", text: "6", correct: true },
    { id: "c", text: "8", correct: false },
    { id: "d", text: "12", correct: false }
  ]
},
{

```

id: 18,

question: "Який кут між векторами $(1, 0, 0)$ та $(0, 1, 0)$?",

options: [

- { id: "a", text: "0°", correct: false },
- { id: "b", text: "45°", correct: false },
- { id: "c", text: "90°", correct: true },
- { id: "d", text: "180°", correct: false }

]

},

{

id: 19,

question: "Формула $d = |Ax_1 + By_1 + Cz_1 + D| / \sqrt{A^2 + B^2 + C^2}$ описує:",

options: [

- { id: "a", text: "Відстань між двома точками", correct: false },
- { id: "b", text: "Відстань від точки до площини", correct: true },
- { id: "c", text: "Довжину вектора", correct: false },
- { id: "d", text: "Кут між площинами", correct: false }

]

},

{

id: 20,

question: "Як виглядає рівняння площини в просторі?",

options: [

- { id: "a", text: " $Ax + By + Cz + D = 0$ ", correct: true },
- { id: "b", text: " $Ax^2 + By^2 + Cz^2 + D = 0$ ", correct: false },
- { id: "c", text: " $Ax + By + Cz = 0$ ", correct: false },
- { id: "d", text: " $Ax - By - Cz = 0$ ", correct: false }

]

},

{

id: 21,

question: "Як знайти кут між площинами з нормальними векторами $(2, -1, 3)$ та $(1, 2, 2)$?",

options: [

- { id: "a", text: " $\arccos(11 / \sqrt{42})$ ", correct: true },
- { id: "b", text: " $\arcsin(3 / \sqrt{14})$ ", correct: false },
- { id: "c", text: " $\tan^{-1}(2 / 3)$ ", correct: false },
- { id: "d", text: "90°", correct: false }

]

```

    },
    {
      id: 22,
      question: "Скільки ребер має правильна чотирикутна піраміда?",
      options: [
        { id: "a", text: "5", correct: false },
        { id: "b", text: "6", correct: false },
        { id: "c", text: "8", correct: true },
        { id: "d", text: "10", correct: false }
      ]
    }
  ];

```

```

const [currentQuestion, setCurrentQuestion] = useState(0);
const [score, setScore] = useState(0);
const [selectedAnswer, setSelectedAnswer] = useState(null);
const [showResult, setShowResult] = useState(false);
const [answered, setAnswered] = useState(false);

```

```

const handleAnswerSelection = (option) => {
  setSelectedAnswer(option);
};

```

```

const handleNextQuestion = () => {
  if (selectedAnswer?.correct) {
    setScore(score + 1);
  }
  setAnswered(true);
};

```

```

const handleContinue = () => {
  setSelectedAnswer(null);
  setAnswered(false);
  const nextQuestion = currentQuestion + 1;
  if (nextQuestion < questions.length) {
    setCurrentQuestion(nextQuestion);
  } else {
    setShowResult(true);
  }
};

```



```

        }
        disabled={answered}
      >
        {option.text}
      </button>
    )}
  </div>
</div>

<div className="action-buttons">
  {answered ? (
    <button
      className="next-btn"
      onClick={handleContinue}
    >
      Наступне питання
    </button>
  ) : (
    <button
      className="submit-btn"
      onClick={handleNextQuestion}
      disabled={!selectedAnswer}
    >
      Перевірити
    </button>
  )}
</div>
</>
):(
  <div className="test-result">
    <h3>Результат тесту</h3>
    <p>
      Ви набрали <span className="score">{score}</span> з{" "}
      <span className="score">{questions.length}</span>
    </p>
    <button className="retry-btn" onClick={handleRetry}>
      Пройти тест ще раз
    </button>
  </div>
):

```

```

        </div>
      )}
    </div>
  );
}

```

export default Test;

Theory.js

```

import React from "react";
import { BlockMath, InlineMath } from "react-katex";
import "katex/dist/katex.min.css";

function Theory() {
  return (
    <div className="theory-page">
      <style>{`
        * {
          box-sizing: border-box;
        }

        body {
          font-family: 'Arial', sans-serif;
          margin: 0;
          padding: 0;
          background: linear-gradient(to right, #e3f2fd, #e1f5fe);
        }

        .theory-page {
          display: flex;
          flex-direction: column;
          min-height: 100vh;
        }

        .header {
          background: #1a73e8;
          color: white;
          padding: 1rem 2rem;
          text-align: center;

```

```
    font-size: 1.5rem;  
    font-weight: bold;  
}
```

```
.footer {  
    background: #1a73e8;  
    color: white;  
    padding: 1rem;  
    text-align: center;  
    margin-top: auto;  
}
```

```
.main-content {  
    display: flex;  
    flex: 1;  
}
```

```
.sidebar {  
    width: 300px;  
    background: #1a73e8;  
    color: white;  
    position: sticky;  
    top: 0;  
    height: fit-content;  
    padding: 2rem;  
    display: flex;  
    flex-direction: column;  
    box-shadow: 2px 0 8px rgba(0, 0, 0, 0.2);  
}
```

```
.sidebar h2 {  
    text-align: center;  
    font-size: 1.5rem;  
    margin-bottom: 2rem;  
}
```

```
.sidebar a {  
    color: white;
```

```

    text-decoration: none;
    margin-bottom: 1rem;
    font-size: 1.2rem;
    padding: 0.5rem;
    transition: background 0.3s ease;
}

.sidebar a:hover {
    background: #0f5bb5;
    border-radius: 8px;
}

.content {
    flex: 1;
    padding: 2rem;
}

.topic {
    margin-bottom: 3rem;
    padding: 2rem;
    background: white;
    border-radius: 15px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    animation: fadeIn 1.5s ease-in-out;
}

h1, h2 {
    color: #1a73e8;
    margin-bottom: 1rem;
}

p {
    font-size: 1.2rem;
    line-height: 1.8;
    color: #333;
}

ul {

```

```

padding-left: 2rem;
margin-bottom: 1.5rem;
}

li {
margin-bottom: 0.8rem;
}

.illustration {
text-align: center;
margin: 2rem 0;
}

.illustration svg {
max-width: 600px;
height: auto;
}

@keyframes fadeIn {
from {
opacity: 0;
transform: translateY(30px);
}
to {
opacity: 1;
transform: translateY(0);
}
}
`}</style>

{/* Header */}
<div className="header">Теорія Стратегічного мислення</div>

{/* Main content */}
<div className="main-content">
  {/* Sidebar Navigation */}
  <div className="sidebar">
    <h2>Навігація</h2>

```

```

<a href="#axioms">Основні аксіоми</a>
<a href="#distance">Відстань між точками</a>
<a href="#sphere">Об'єм та площа кулі</a>
<a href="#line-plane">Взаємне розташування прямої і площини</a>
<a href="#plane-intersection">Кут між площинами</a>
<a href="#plane-equation">Рівняння площини</a>
<a href="#parallel-planes">Відстань між паралельними площинами</a>
<a href="#plane-relations">Взаємне розташування двох площин</a>
<a href="#polyhedra">Многогранники: основні поняття та типи</a>
<a href="#euler-formula">Формула Ейлера для многогранників</a>
</div>

```

```

{ /* Main Content */

```

```

<div className="content">

```

```

  <div id="axioms" className="topic">

```

```

    <h2>Основні аксіоми стереометрії</h2>

```

```

    <p>Стереометрія базується на трьох основних аксіомах:</p>

```

```

    <ul>

```

```

      <li>

```

```

        <strong>Аксіома 1:</strong> Через будь-які три точки, які не лежать на одній
        прямій,
        проходить одна площина.

```

```

      </li>

```

```

      <li>

```

```

        <strong>Аксіома 2:</strong> Якщо дві точки прямої лежать у площині, то всі
        точки цієї
        прямої лежать у площині.

```

```

      </li>

```

```

      <li>

```

```

        <strong>Аксіома 3:</strong> Якщо дві площини перетинаються, то вони
        перетинаються по
        прямій.

```

```

      </li>

```

```

    </ul>

```

```

  <div className="illustration">

```

```

    <h3>Ілюстрація Аксіом</h3>

```

```

    <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 600 400">

```

```

<polygon
  points="150,300 450,300 400,100 200,100"
  fill="rgba(135, 206, 235, 0.5)"
  stroke="blue"
  strokeWidth="2"
/>
<circle cx="250" cy="250" r="8" fill="red"/>
<circle cx="350" cy="200" r="8" fill="red"/>
<circle cx="300" cy="150" r="8" fill="red"/>
<line x1="250" y1="250" x2="350" y2="200" stroke="black" strokeWidth="2"/>
<line x1="350" y1="200" x2="300" y2="150" stroke="black" strokeWidth="2"/>
</svg>
</div>

```

<h3>Приклад</h3>

```

<p>
  <strong>Задача:</strong> Дано точки <BlockMath math="A(1, 2, 3)"/>, {" "}
  <BlockMath math="B(4, 5, 6)"/> i <BlockMath math="C(7, 8, 9)"/>. Знайти площину,
  яка
  проходить через ці точки.
</p>

```

```

<p><strong>Розв'язок:</strong></p>

```

```

<ol>
  <li>
    Обчислюємо нормальний вектор за допомогою векторного добутку: {" "}
    <BlockMath math="\vec{AB} \times \vec{AC}"/>.
  </li>
  <li>
    Підставляємо у рівняння площини: <BlockMath math="Ax + By + Cz + D = 0"/>.
  </li>
</ol>

```

```

<p>
  <strong>Відповідь:</strong> Рівняння площини: <BlockMath math="x + y + z = 1"/>.
</p>

```

```

</div>

```

```

<div id="distance" className="topic">

```

<h2>Відстань між двома точками у просторі</h2>

<p>Відстань між двома точками у просторі обчислюється за формулою:</p>

<BlockMath math="d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}">

<p>Де:</p>

<InlineMath math="(x_1, y_1, z_1)"> — координати першої точки.

<InlineMath math="(x_2, y_2, z_2)"> — координати другої точки.

<InlineMath math="d"> — відстань між точками.

<div className="illustration">

<h3>Ілюстрація Формули</h3>

<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 600 400">

<circle cx="150" cy="300" r="8" fill="red"/>

<text x="130" y="320" fill="#333" fontSize="16">

$A(x_1, y_1, z_1)$

</text>

<circle cx="450" cy="100" r="8" fill="blue"/>

<text x="470" y="110" fill="#333" fontSize="16">

$B(x_2, y_2, z_2)$

</text>

<line x1="150" y1="300" x2="450" y2="100" stroke="#1a73e8" strokeWidth="4"/>

<text x="300" y="210" fill="#1a73e8" fontSize="18">

d

</text>

<line x1="150" y1="300" x2="450" y2="300" stroke="gray" strokeWidth="2"

strokeDasharray="5,5"/>

<line x1="450" y1="300" x2="450" y2="100" stroke="gray" strokeWidth="2"

strokeDasharray="5,5"/>

<text x="300" y="320" fill="gray" fontSize="14">

Δx

</text>

```

<text x="460" y="200" fill="gray" fontSize="14">
  Δy
</text>
</svg>
</div>

```

Приклад

<p>

Задача: Знайти відстань між точками{" "}

$A(1, 2, 3)$ і $B(4, 6, 8)$.

</p>

Розв'язок:

Знайдемо різниці координат:

$x_2 - x_1 = 4 - 1 = 3$

$y_2 - y_1 = 6 - 2 = 4$

$z_2 - z_1 = 8 - 3 = 5$

Підставимо значення у формулу:

$d = \sqrt{3^2 + 4^2 + 5^2}$

Обчислимо квадрат кожного значення:

$d = \sqrt{9 + 16 + 25}$

Знайдемо корінь:

$d = \sqrt{50} \approx 7.07$

<p>

Відповідь: Відстань між точками дорівнює{" "}

7.07 одиниць.

</p>

```
    </div>  
  </div>  
</div>
```

```
  {/* Footer */  
    <div className="footer">© 2024 Теорія Стереометрії</div>  
  </div>  
);  
}
```

```
export default Theory;
```

