

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

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОЇ НАВЧАЛЬНОЇ ГРИ З ТЕМИ «ТРИГОНОМЕТРІЯ»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня магістра

Виконавець роботи Левченкова Альбіна Андріївна

_____«_____»____202_ р.

(підпис)

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

_____«_____»____202_ р.

(підпис)

Рецензент

ПОЛТАВА 2024

РЕФЕРАТ

Записка: 82 с., 16 рис., 2 таблиці, 1 додаток, 13 джерел.

ІНТЕРАКТИВНА ГРА, REACT, ТРИГОНОМЕТРІЯ, ГЕЙМІФІКАЦІЯ

Об'єктом розробки є інтерактивна навчальна платформа для вивчення тригонометрії.

Предметом розробки є програмна реалізація навчальної гри для закріплення знань із тригонометрії.

Метою роботи є створення інтерактивного додатка, що дозволяє учням і студентам вивчати тригонометрію в цікавій та доступній формі, використовуючи сучасні технології.

Методи дослідження: аналіз освітніх платформ і сучасних технологій, використання мови програмування JavaScript і фреймворку React для розробки клієнтської частини додатка, застосування бібліотеки KaTeX для відображення математичних формул, зберігання даних у localStorage для статистики результатів користувачів, створення стилів через styled-components для адаптації інтерфейсу.

Результатом роботи стала інтерактивна навчальна платформа, яка містить кілька модулів. У теоретичному розділі користувач може переглядати базову інформацію про тригонометричні функції, співвідношення та тотожності. У практичному розділі представлені ігри для закріплення знань. Наприклад, у модулі "Теорема косинусів" користувач розв'язує задачі за теоремою косинусів, у модулі "Теорема синусів" – задачі, пов'язані із синусами трикутників, а у модулі "Значення тригонометричних функцій" – знаходить значення синусів, косинусів та тангенсів для заданих кутів. Окрім цього, є модуль "Тригонометричні співвідношення", який спрямований на обчислення співвідношень між сторонами трикутників, і модуль "Тригонометричні вирази", що вчить спрощувати вирази та працювати із тригонометричними тотожностями.

Кожна гра відстежує правильність відповідей, виводить результат і дозволяє зберегти статистику. В кінці кожного модуля користувач отримує повний звіт про час, витрачений на проходження, і кількість правильних відповідей.

Розроблений додаток адаптований для різних пристроїв, має інтуїтивно зрозумілий інтерфейс і простий у використанні. Протестовано, що програма працює без збоїв та відповідає вимогам до навчальних ігор. Використання такої платформи дозволяє ефективніше вивчати складні теми тригонометрії завдяки ігровій формі подання матеріалу.

Робота має практичне значення і може бути використана в школах, університетах та на курсах для підвищення мотивації учнів до вивчення математики.

ЗМІСТ

ВСТУП.....	7
ПОСТАНОВКА ЗАДАЧІ	9
1. ІНФОРМАЦІЙНИЙ ОГЛЯД	11
1.1. Аналіз сучасних інтерактивних технологій у навчанні	11
1.2. Особливості використання інтерактивних ігор у викладанні математики	12
1.3. Огляд програмного забезпечення для вивчення тригонометрії.....	17
2. ТЕОРЕТИЧНА ЧАСТИНА	22
2.1. Основи тригонометрії: базові поняття, функції та їх властивості	22
2.2. Методи викладання тригонометрії з використанням інтерактивних інструментів	24
2.3. Психолого-педагогічні аспекти створення навчальних ігор	27
3. ПРАКТИЧНА ЧАСТИНА.....	30
3.1. Вибір технологій для розробки програмного продукту	30
3.2. Реалізація функціональних модулів гри.....	35
3.3. Інструкція для користувача	39
ВИСНОВКИ	47
СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	49
ДОДАТОК А.	50

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
$\sin(\alpha)$, $\cos(\alpha)$, $\operatorname{tg}(\alpha)$, $\operatorname{ctg}(\alpha)$	Основні тригонометричні функції: синус, косинус, тангенс, котангенс кута α .
UX/UI	User Experience/User Interface – зручність і дизайн інтерфейсу користувача.
React.js	JavaScript-бібліотека для створення інтерфейсів користувача.
Node.js	Платформа для виконання серверного JavaScript-коду.
Firebase	Хмарна платформа для зберігання даних, автентифікації та хостингу.
LMS	Learning Management System – система управління навчанням.
JSON	JavaScript Object Notation – формат обміну даними.
HTML	HyperText Markup Language – мова розмітки гіпертексту для створення веб-сторінок.
CSS	Cascading Style Sheets – каскадні таблиці стилів для оформлення зовнішнього вигляду веб-сторінок.
SSL	Secure Sockets Layer – протокол для захищеного обміну даними в мережі Інтернет.
CDN	Content Delivery Network – мережа доставки контенту для швидкого завантаження веб-ресурсів.
API	Application Programming Interface – інтерфейс

	для взаємодії програм.
Адаптивний дизайн	Підхід до розробки веб-сторінок, що забезпечує коректне відображення на різних пристроях.
Теорема синусів	Формула, що встановлює співвідношення між сторонами трикутника і синусами протилежних кутів.
Теорема косинусів	Формула, що визначає зв'язок між сторонами трикутника через косинус кута.
Інтерактивні модулі	Розділи сайту, які включають завдання у форматі гри з динамічним зворотним зв'язком.
Тригонометрична тотожність	Рівність, яка виконується для всіх допустимих значень аргументу тригонометричних функцій.
$\sin(\alpha)$, $\cos(\alpha)$, $\operatorname{tg}(\alpha)$, $\operatorname{ctg}(\alpha)$	Основні тригонометричні функції: синус, косинус, тангенс, котангенс кута α .

ВСТУП

У сучасних умовах цифрової трансформації освіти інтерактивні технології стають невід'ємною частиною навчального процесу. Інтерактивні ігри, зокрема, демонструють високу ефективність у підвищенні зацікавленості учнів, полегшенні засвоєння складних тем і розвитку логічного мислення. Тригонометрія, будучи однією з фундаментальних тем у математиці, часто викликає труднощі у розумінні через її абстрактність та складність формул.

Актуальність даної роботи полягає у розробці інтерактивної навчальної гри, яка допоможе учням і студентам краще засвоювати тригонометричні поняття, функції та співвідношення за допомогою гейміфікації. Використання ігор у навчальному процесі дозволяє забезпечити інтерактивний підхід, що відповідає сучасним потребам учнів і викладачів.

Об'єктом розробки є інтерактивна навчальна платформа, орієнтована на вивчення тригонометрії.

Предметом розробки є програмна реалізація інтерактивної гри для закріплення знань із тригонометрії.

Метою роботи є створення інтерактивного навчального додатка, який дозволить учням вивчати тригонометрію у зручній та цікавій формі, підвищуючи мотивацію до навчання.

Методи дослідження включають аналіз сучасних інтерактивних освітніх технологій, використання React для створення клієнтської частини додатка, проектування модульної архітектури платформи, а також тестування та оцінку ефективності розробленого продукту.

Робота складається зі вступу, де описано актуальність, мету, об'єкт і предмет дослідження; інформаційного огляду з аналізом сучасних освітніх технологій та програм для вивчення тригонометрії; теоретичної частини, присвяченої базовим поняттям тригонометрії та методам викладання; практичної частини, де описано вибір технологій, розробку функціональних модулів, проектування інтерфейсу та тестування програми; висновків і рекомендацій; списку використаних джерел; додатків, які містять код програми та приклади інтерфейсу.

Розробка інтерактивної гри сприятиме вдосконаленню навчального процесу, дозволяючи учням закріплювати знання у цікавій і доступній формі, а викладачам – підвищувати якість викладання складних математичних тем.

ПОСТАНОВКА ЗАДАЧІ

Основною метою даного проекту є розробка інтерактивної навчальної гри для вивчення тригонометрії, яка сприятиме підвищенню інтересу до вивчення математики, полегшенню засвоєння складних понять та розвитку логічного мислення у студентів і учнів. Ігровий підхід дозволяє забезпечити інтерактивний формат навчання, що відповідає сучасним вимогам до освітніх інструментів.

Для розробника було поставлено наступні завдання:

1. **Визначення ключових навчальних тем** для створення ігрових модулів: теорема косинусів, теорема синусів, тригонометричні співвідношення, тангенс і котангенс, значення тригонометричних функцій та спрощення тригонометричних виразів.
2. **Розробка інтерактивних модулів** із завданнями, які включають візуалізацію трикутників, обчислення тригонометричних функцій і застосування теоретичних знань до практичних задач.
3. **Проектування архітектури додатка** із використанням React для реалізації клієнтської частини програми, розробки модульного дизайну для легкого додавання нових функцій.
4. **Розробка інтуїтивного інтерфейсу користувача**, який дозволить легко орієнтуватися у грі та виконувати завдання. Інтерфейс повинен враховувати принципи UX/UI-дизайну, забезпечуючи комфортну взаємодію користувачів із платформою.
5. **Створення системи збереження результатів гри**, яка дозволяє відстежувати прогрес користувача, фіксувати час виконання завдань та кількість правильних відповідей для подальшого аналізу.
6. **Тестування та оптимізація ігрових модулів**, включаючи перевірку правильності роботи завдань, коректності відображення інтерфейсу та сумісності програми з різними браузерами та пристроями.
7. **Забезпечення адаптивності дизайну**, щоб гра коректно відображалася на пристроях із різними розмірами екрану, включаючи мобільні телефони,

планшети та комп'ютери.

8. Реалізація інструкції для користувачів, яка пояснює, як працювати з програмою, зберігати прогрес і використовувати різні модулі для навчання.

Головною вимогою є створення ефективного, безпечного та цікавого навчального інструменту, який буде зрозумілим і корисним для учнів та викладачів. Програма має забезпечити інтерактивний підхід до навчання, який допоможе користувачам застосовувати отримані знання в різних контекстах та ситуаціях.

Це завдання включає розробку інтерфейсу користувача, створення інтерактивних завдань, тестування модулів та забезпечення стабільної роботи програми. Розроблена навчальна гра стане важливим інструментом для підвищення ефективності викладання тригонометрії та зацікавленості учнів у вивченні математики.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1. Аналіз сучасних інтерактивних технологій у навчанні

У сучасному освітньому процесі інтерактивні технології відіграють ключову роль у підвищенні зацікавленості студентів, засвоєнні складних тем та розвитку навичок критичного мислення. Завдяки технологіям навчання стало не лише інформаційно насиченим, але й більш адаптивним до індивідуальних потреб учнів. Розглянемо основні інтерактивні інструменти, які застосовуються у навчанні.

Інтерактивні освітні платформи, такі як **Kahoot**, **Quizlet** та **Edmodo**, дозволяють створювати та проводити вікторини, інтерактивні уроки й обговорення. Наприклад, **Kahoot** надає можливість викладачам створювати ігрові тести, які студенти можуть проходити в реальному часі. Це особливо ефективно для перевірки знань з математики, де можна використовувати завдання на обчислення та вибір правильних відповідей.

Навчальні симулятори стали одним із популярних інструментів інтерактивного навчання. Наприклад, у викладанні фізики чи біології широко використовуються симулятори, такі як **PhET Interactive Simulations**, які дозволяють учням експериментувати з різними явищами без необхідності лабораторного обладнання. У контексті математики, інтерактивні симулятори допомагають вивчати геометрію, тригонометрію та алгебру.

Електронні дошки, такі як **Promethean** чи **Smart Board**, дозволяють створювати інтерактивні презентації та задачі. Учні можуть взаємодіяти із завданнями безпосередньо на дошці: малювати графіки, знаходити точки перетину функцій чи виконувати геометричні побудови. Це значно полегшує викладення матеріалу в класі.

Мобільні додатки, такі як **Photomath**, **GeoGebra** та **Brilliant**, дають можливість учням самостійно вивчати математику. Наприклад, **GeoGebra** дозволяє створювати інтерактивні моделі трикутників, аналізувати їх властивості, вивчати тригонометричні функції, а також розв'язувати задачі з теорем синусів та косинусів.

Одним із провідних трендів є впровадження ігор у навчання. Інтерактивні ігри

дозволяють учням засвоювати матеріал у ненав'язливій формі. Наприклад, гра **Mathletics** включає математичні змагання між учнями, що стимулює розвиток здорової конкуренції. Вивчення тригонометрії також можна інтегрувати у формат гри, як це реалізовано у нашому проекті.

Технології VR (віртуальна реальність) та AR (доповнена реальність) стають невід'ємною частиною інтерактивного навчання. Наприклад, додаток **Google Expeditions** дозволяє учням занурюватися у віртуальні експедиції. У викладанні тригонометрії можна використовувати AR для візуалізації трикутників і тривимірних фігур, що допомагає краще зрозуміти співвідношення між сторонами та кутами.

Наприклад, під час викладання теми «Тригонометричні функції» викладач може використовувати симулятор у **GeoGebra**, де учень побудує графік функції $y = \sin(x)$ та дослідить, як змінюються значення функції при зміні кута. У той же час, можна залучити гейміфікацію, створивши квест, де учні повинні розв'язати кілька завдань для відкриття нових рівнів.

Таким чином, інтерактивні технології відкривають нові можливості для організації навчального процесу, роблячи його більш цікавим та ефективним. Їхнє застосування у навчанні тригонометрії дозволяє не лише покращити засвоєння матеріалу, але й розвинути у учнів аналітичне мислення, необхідне для розв'язування складних завдань.

1.2. Особливості використання інтерактивних ігор у викладанні математики

Інтерактивні ігри є потужним інструментом для залучення учнів до навчального процесу, особливо у викладанні таких складних предметів, як математика. Їх використання сприяє підвищенню мотивації, розвитку логічного мислення та закріпленню теоретичних знань у практичному форматі. Завдяки гейміфікації учні не тільки краще запам'ятовують матеріал, але й отримують позитивні емоції від процесу навчання.

Переваги інтерактивних ігор у навчанні математики

1. **Підвищення мотивації.** Ігровий формат спонукає учнів активно брати участь у навчальному процесі. Наприклад, якщо для розв'язання математичної задачі потрібен прогрес у грі, учні охочіше шукають правильні відповіді, оскільки це має безпосередній вплив на їх успіх у грі.
2. **Закріплення матеріалу через практику.** Ігри дозволяють застосовувати теоретичні знання в практичних ситуаціях. Наприклад, у грі, де потрібно розрахувати довжину сторін трикутника або значення синуса кута, учень не лише вирішує завдання, але й закріплює знання про тригонометричні функції.
3. **Розвиток навичок командної роботи.** Багатокористувацькі ігри можуть сприяти розвитку соціальних навичок, оскільки учні взаємодіють, допомагаючи один одному виконувати завдання. Наприклад, у групових іграх учні разом вирішують задачі на побудову графіків чи розв'язання рівнянь.
4. **Індивідуалізація навчання.** Ігри дозволяють враховувати рівень підготовки кожного учня. Системи з адаптивною складністю, такі як інтерактивні модулі нашого проєкту, підлаштовуються під швидкість навчання користувача, надаючи простіші або складніші завдання.
5. **Візуалізація математичних концепцій.** Багато учнів мають труднощі із розумінням абстрактних понять, таких як тригонометричні функції чи графіки. Ігри з інтерактивними моделями дозволяють наочно побачити залежності між сторонами та кутами трикутника, що полегшує сприйняття матеріалу.

Приклади інтерактивних ігор у викладанні математики

1. **Math Blaster.** Гра, яка мотивує дітей вирішувати математичні задачі, щоб просуватися далі у грі (див. рис. 1.1). Завдання включають розв'язання рівнянь, побудову графіків та обчислення геометричних параметрів.



Рисунок 1.1 – Гра «Math Blaster»

2. **DragonBox Algebra.** Це гра, яка перетворює складні рівняння на цікаві головоломки, дозволяючи учням навчитися розв'язувати алгебраїчні рівняння у ненав'язливій формі (див. рис. 1.2).



Рисунок 1.2 – Гра «DragonBox Algebra»

3. **GeoGebra Games.** Інтерактивна платформа GeoGebra дозволяє створювати ігри, в яких учні працюють із геометричними побудовами, графіками функцій та тригонометрією (див. рис. 1.3). Наприклад, завдання на знаходження синуса кута в трикутнику або побудову тригонометричних графіків.

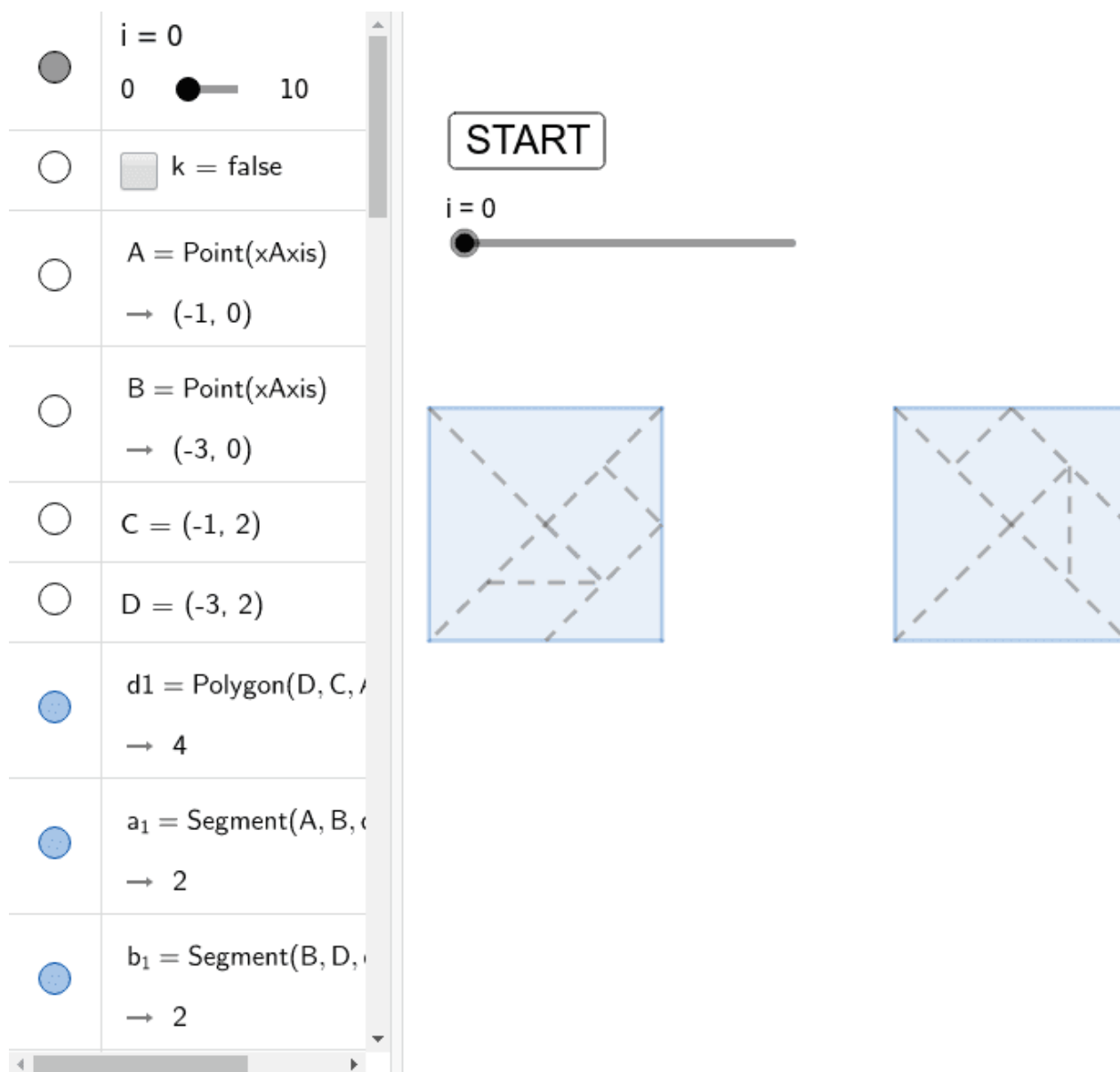


Рисунок 1.3 – Гра «GeoGebra Games»

4. **Інтерактивні модулі нашого проєкту.** У нашому проєкті ми створили серію інтерактивних ігор, таких як:

- **«Обчисли довжину сторони»** — гра, у якій потрібно застосувати теорему косинусів для обчислення сторін трикутника.
- **«Синус і косинус»** — гра для закріплення значень тригонометричних функцій.
- **«Тригонометричні тотожності»** — завдання на спрощення виразів із використанням тригонометричних формул.

Особливості реалізації інтерактивних ігор у навчанні

1. **Гейміфікація навчального процесу.** Навчання через ігри перетворює рутину на захоплюючий процес, що позитивно впливає на зацікавленість учнів.
2. **Миттєвий зворотний зв'язок.** Ігри надають учням миттєвий зворотний зв'язок про їхні дії, наприклад, підказки чи повідомлення про помилки. Це дозволяє швидко виправляти помилки та закріплювати правильний підхід.
3. **Інтерактивність.** Ігрові механіки, такі як перетягування об'єктів, побудова фігур або введення числових значень, дозволяють зробити навчання більш динамічним.
4. **Мотиваційні елементи.** Додавання системи винагород, рівнів та досягнень у гру стимулює учнів до досягнення кращих результатів.

Інтерактивні ігри у викладанні математики є ефективним засобом підвищення мотивації учнів та покращення їхніх знань. Вони дозволяють комбінувати навчання та розваги, сприяючи кращому засвоєнню складних тем, таких як тригонометрія. Реалізація ігор у нашому проєкті доводить, що навчальний процес може бути цікавим, динамічним та ефективним.

1.3. Огляд програмного забезпечення для вивчення тригонометрії

У сучасному освітньому процесі використання програмного забезпечення для вивчення математики, зокрема тригонометрії, стало невід'ємною частиною інтерактивного навчання. Це дозволяє учням не лише опановувати теоретичний матеріал, але й практикувати його застосування у візуально зрозумілому середовищі.

Популярне програмне забезпечення для вивчення тригонометрії

1. **GeoGebra** — це потужний інструмент для вивчення геометрії, алгебри та тригонометрії (див. рис. 1.3). Програма дозволяє будувати трикутники, створювати графіки тригонометричних функцій та досліджувати їхні властивості.
- **Особливості:**
 - Інтерактивні побудови трикутників із динамічним змінням

параметрів.

- Візуалізація тригонометричних функцій, таких як синус, косинус, тангенс.
- Підтримка анімацій для демонстрації змінних величин, наприклад, зміна кута у трикутнику.

- **Переваги:**

- Доступність на різних платформах, включаючи мобільні пристрої.
- Простий інтерфейс для учнів різного рівня підготовки.

2. **Desmos** — це онлайн-платформа для створення графіків функцій, включаючи тригонометричні (див. рис. 1.4). Вона активно використовується у школах для навчання основам тригонометрії та аналізу функцій.

- **Особливості:**

- Побудова та візуалізація графіків тригонометричних функцій.
- Можливість змінювати параметри функцій у реальному часі, щоб побачити, як це впливає на їхній вигляд.
- Інтуїтивний інтерфейс, що підходить як для викладачів, так і для учнів.

- **Переваги:**

- Онлайн-доступність без необхідності встановлення додаткового ПЗ.
- Інтерактивність, що дозволяє учням швидко зрозуміти зв'язок між змінними у функціях.

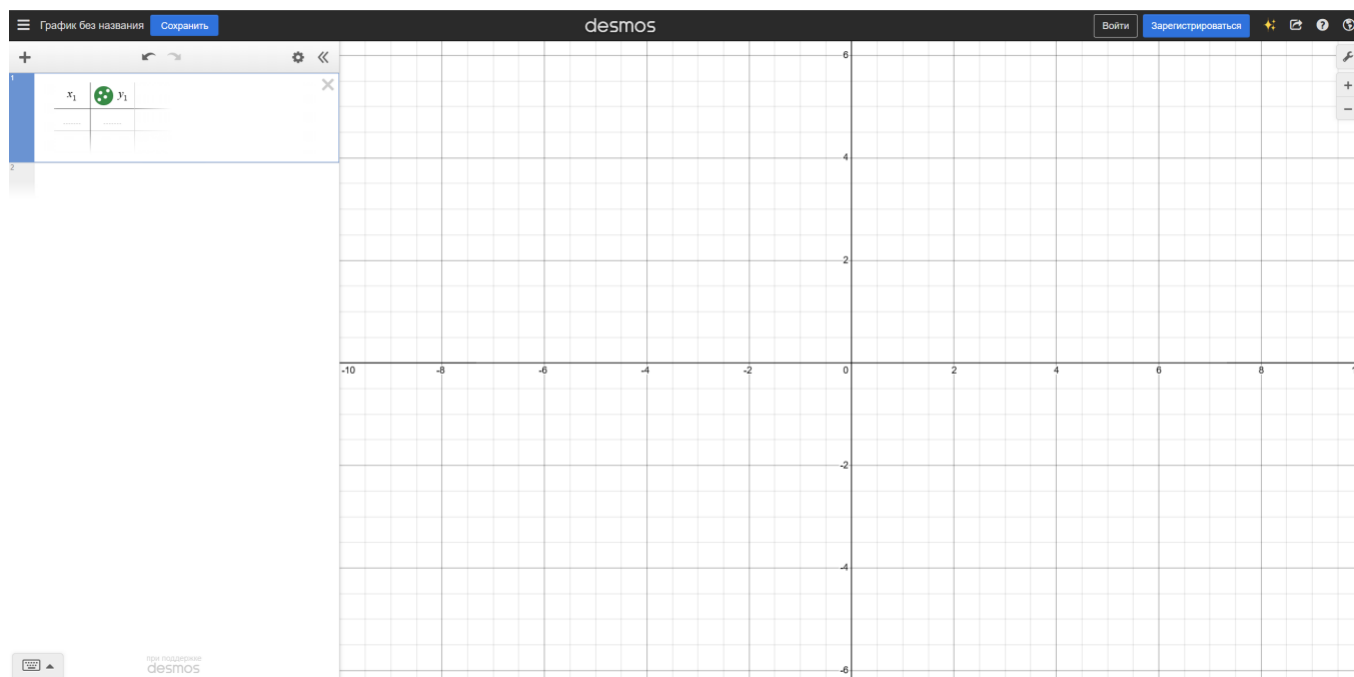


Рисунок 1.4 – Daemos

3. **Trigonometry Tutor** — це спеціалізована програма для вивчення тригонометрії, яка пропонує практичні завдання, інтерактивні уроки та тести для перевірки знань.

- **Особливості:**

- Покрокові уроки, що пояснюють базові поняття тригонометрії.
- Задачі на обчислення значень функцій, використання тригонометричних формул та розв'язання рівнянь.
- Автоматичне оцінювання правильності відповідей.

- **Переваги:**

- Орієнтованість на самостійне навчання.
- Відповідність шкільній програмі.

4. **Khan Academy.** Освітня платформа, що пропонує широкий вибір відеоуроків, вправ і тестів, у тому числі з тригонометрії.

- **Особливості:**

- Пояснення теоретичних основ тригонометрії у простій і доступній формі.
- Практичні завдання, що допомагають закріпити знання.

- Інтерактивні елементи, наприклад, графіки та діаграми.
- **Переваги:**
 - Безкоштовний доступ до матеріалів.
 - Адаптація рівня складності до потреб учня.

Порівняння інструментів зображено в таблиці (див. табл. 1.1)

Програмне забезпечення	Основна функціональність	Платформа	Рівень складності
GeoGebra	Побудова трикутників, графіки функцій	Windows, Mac, Web	Середній
Desmos	Графіки тригонометричних функцій	Web	Легкий
Trigonometry Tutor	Уроки та завдання з тригонометрії	Windows, Web	Легкий
Khan Academy	Відеоуроки, інтерактивні вправи	Web	Легкий

У проєкті інтерактивне навчання тригонометрії також реалізується через серію модулів, таких як:

- **«Косинусова теорема»** — практичні завдання для обчислення сторін трикутника.
- **«Значення тригонометричних функцій»** — ігрові завдання для закріплення значень синуса, косинуса, тангенса.
- **«Тригонометричні тотожності»** — спрощення тригонометричних виразів через інтерактивні вправи.

Програмне забезпечення для вивчення тригонометрії сприяє кращому

засвоєнню матеріалу за рахунок інтерактивності, візуалізації та доступності. У нашому проєкті використовуються найкращі підходи з цих програм, що дозволяє забезпечити ефективне та цікаве навчання для учнів різного рівня підготовки.

2. ТЕОРЕТИЧНА ЧАСТИНА

2.1. Основи тригонометрії: базові поняття, функції та їх властивості

Тригонометрія — це розділ математики, що вивчає залежності між кутами та сторонами трикутників, а також властивості тригонометричних функцій. Основи тригонометрії є фундаментом для розуміння більш складних математичних концепцій та їх застосування в науці, техніці й повсякденному житті [1].

Базові поняття

1. **Трикутник.** Основним об'єктом дослідження у тригонометрії є трикутник. Тригонометрія базується на властивостях прямокутного трикутника, у якому один з кутів дорівнює 90° .
 - **Гіпотенуза** — найдовша сторона трикутника, яка лежить навпроти прямого кута.
 - **Катети** — дві сторони, що утворюють прямий кут.
2. **Кути.** Кут у трикутнику вимірюється у градусах або радіанах. Відповідно до тригонометрії, сума кутів у трикутнику завжди дорівнює 180° .
 - Радіан — це інший спосіб вимірювання кутів, де повне коло становить 2π радіан.

Тригонометричні функції

Тригонометричні функції визначають співвідношення між сторонами прямокутного трикутника. Основними функціями є:

Синус. Синус кута в прямокутному трикутнику визначається як відношення довжини протилежного катета до довжини гіпотенузи. Якщо α — це кут, то синус кута α позначається як $\sin(\alpha)$ і визначається формулою:

$$\sin(\alpha) = \frac{\text{протилежний катет}}{\text{гіпотенуза}}$$

Синус функція має період 2π і коливається між -1 та 1 .

Косинус. Косинус кута в прямокутному трикутнику визначається як відношення довжини прилеглого катета до довжини гіпотенузи. Якщо α — це кут, то косинус кута α позначається як $\cos(\alpha)$ і визначається формулою:

$$\cos(a) = \frac{\text{прилеглий катет}}{\text{гіпотенуза}}$$

Косинус функція також має період 2π і коливається між -1 та 1 .

Тангенс. Тангенс кута в прямокутному трикутнику визначається як відношення довжини протилежного катета до довжини прилеглого катета. Якщо α — це кут, то тангенс кута α позначається як $\operatorname{tg}(\alpha)$ і визначається формулою:

$$\operatorname{tg}(\alpha) = \frac{\text{протилежний катет}}{\text{прилеглий катет}}$$

Тангенс функція має період π і може приймати будь-які значення від $-\infty$ до ∞ .

Котангенс. Котангенс кута в прямокутному трикутнику визначається як відношення довжини прилеглого катета до довжини протилежного катета. Якщо α — це кут, то котангенс кута α позначається як $\operatorname{ctg}(\alpha)$ і визначається формулою:

$$\operatorname{ctg}(\alpha) = \frac{\text{прилеглий катет}}{\text{протилежний катет}}$$

Котангенс функція також має період π і може приймати будь-які значення від $-\infty$ до ∞ .

Графіки тригонометричних функцій. Графіки синуса, косинуса, тангенса та котангенса відіграють важливу роль у вивченні тригонометрії. Вони допомагають візуалізувати поведінку цих функцій, їх періодичність та симетрію [1].

- Графік синуса $\sin(x)$ — це хвиляста крива, яка коливається між -1 та 1 з періодом 2π .
- Графік косинуса $\cos(x)$ — також хвиляста крива, яка коливається між -1 та 1 з періодом 2π .
- Графік тангенса $\operatorname{tg}(x)$ має вертикальні асимптоти в точках $\frac{\pi}{2} + k\pi$, де k — ціле число, і період π .
- Графік котангенса $\operatorname{ctg}(x)$ має вертикальні асимптоти в точках $k\pi$, де k — ціле число, і період π .

Застосування тригонометричних функцій. Тригонометричні функції широко застосовуються у різних галузях науки та техніки. Вони використовуються для аналізу періодичних процесів, таких як коливання та хвилі, у геометрії для

визначення кутів і довжин сторін трикутників, в інженерії для розрахунку навантажень та інших задач.

Приклади застосування

1. **Розв'язання задач.** Знаходження невідомої сторони або кута у трикутнику.

Наприклад:

- У трикутнику з гіпотенузою 10 см і кутом 30° , знайти довжину протилежного катета: $\sin 30^\circ = 0.5$, катет = $\sin 30^\circ \times 10 = 5$ см.

2. **Фізика та інженерія.** Вивчення коливальних процесів, моделювання хвиль.

3. **Комп'ютерна графіка.** Використання тригонометричних функцій для побудови анімацій та відтворення об'єктів у тривимірному просторі.

Знання та розуміння основних тригонометричних функцій є важливим елементом математичної освіти, що дозволяє учням вирішувати різноманітні практичні задачі та застосовувати їх у подальшому навчанні та професійній діяльності.

2.2. Методи викладання тригонометрії з використанням інтерактивних інструментів

Вивчення тригонометрії завжди було викликом для багатьох учнів через її абстрактність і складність сприйняття. Використання інтерактивних інструментів допомагає зробити процес навчання більш захопливим, наочним і доступним. Інтерактивні методи дозволяють учням не лише вивчати тригонометрію, а й застосовувати її на практиці, що сприяє кращому засвоєнню матеріалу.

Традиційні підходи до викладання тригонометрії

Класичні методи викладання тригонометрії базуються на:

1. Лекціях із поясненням теоретичного матеріалу.
2. Розв'язуванні задач на папері.
3. Використанні таблиць значень тригонометричних функцій.

Хоча ці методи ефективні для передачі базових знань, вони можуть бути недостатньо мотивуючими для учнів, які звикли до інтерактивних технологій у повсякденному житті.

Інтерактивні інструменти у викладанні тригонометрії

Використання сучасних технологій у навчанні відкриває нові можливості для викладання тригонометрії.

1. **Онлайн-симулятори та візуалізації.** Наприклад, програми як **GeoGebra** дозволяють учням побачити, як змінюються значення тригонометричних функцій залежно від кутів. Учні можуть створювати трикутники, вимірювати кути, і одразу отримувати результати обчислень [2].
 - **Приклад:** Побудова графіків функцій $\sin x$, $\cos x$, $\tan x$, $\cot x$ та спостереження їх періодичності.
2. **Інтерактивні навчальні платформи**
 - **Khan Academy:** пропонує інтерактивні вправи, відеоуроки та тести для засвоєння тригонометрії.
 - **Brilliant.org:** дозволяє вирішувати задачі з використанням інтерактивних візуалізацій та покрокових підказок.
3. **Гейміфікація навчання.** Використання ігор для навчання дозволяє зробити процес вивчення тригонометрії цікавим і захопливим. Наприклад, інтерактивні ігри, де учні розв'язують задачі, щоб перейти на наступний рівень.
 - **Приклад:** Гра, у якій учень має знайти значення кута або довжину сторони, щоб розблокувати наступне завдання.
4. **Мобільні додатки.** Додатки, такі як **Photomath** або **Desmos**, дозволяють учням вирішувати тригонометричні рівняння або будувати графіки прямо на смартфоні.
 - **Desmos:** допомагає візуалізувати функції та експериментувати з параметрами графіків.
 - **Trig Solver:** забезпечує автоматичне розв'язування задач із детальним поясненням.
5. **Доповнена та віртуальна реальність (AR/VR).** Використання VR-додатків для вивчення тригонометрії дає змогу учням «поринути» у світ геометричних об'єктів.

- **Приклад:** Учень може віртуально побачити трикутник у тривимірному просторі, змінювати його параметри та спостерігати залежності між сторонами й кутами.

Методи інтеграції інтерактивних інструментів у навчання

1. **Проектний підхід.** Учням пропонують створити проекти, де вони використовують тригонометрію для вирішення реальних завдань, наприклад, розрахунок висоти будівлі за допомогою кута нахилу.
2. **Фліп-клас (Flipped Classroom).** Учні вивчають теорію вдома за допомогою інтерактивних платформ, а на уроках застосовують знання на практиці.
3. **Командна робота.** Учні працюють у групах над інтерактивними задачами, використовуючи мобільні додатки або онлайн-платформи для спільного розв'язування проблем.
4. **Інтеграція до традиційних уроків.** Інтерактивні інструменти використовуються як доповнення до лекцій і практичних занять, наприклад, для демонстрації графіків або візуалізації тригонометричних функцій.

Переваги інтерактивних методів

- **Мотивація учнів:** Гейміфікація та візуалізація роблять навчання захопливим.
- **Краще розуміння матеріалу:** Наочність допомагає учням зрозуміти абстрактні поняття.
- **Практичне застосування:** Учні бачать, як тригонометрія використовується у реальному світі.
- **Індивідуальний підхід:** Платформи дозволяють адаптувати завдання до рівня кожного учня.

Приклад впровадження інтерактивного уроку

На уроці з теми «Графіки тригонометричних функцій» можна використати наступний підхід:

1. Використати платформу GeoGebra для побудови графіків $\sin x$ та $\cos x$ [3].
2. Дати учням змогу змінювати параметри амплітуди та частоти для спостереження змін у графіках.
3. Завершити урок інтерактивною грою, у якій учні мають визначити значення

функції для заданих кутів.

Інтерактивні інструменти відкривають нові можливості для викладання тригонометрії. Вони роблять процес навчання цікавим, наочним і практичним, сприяють кращому розумінню матеріалу та формуванню навичок застосування знань у реальних ситуаціях. У нашій роботі інтерактивні методи реалізовані через навчальні модулі, що дозволяють закріпити знання про тригонометричні функції, тотожності та задачі.

2.3. Психолого-педагогічні аспекти створення навчальних ігор

У процесі розробки навчальних ігор важливо враховувати психолого-педагогічні аспекти, які сприяють ефективному засвоєнню знань, розвитку навичок та формуванню інтересу до навчання. Правильно спроектована гра не лише допомагає закріпити навчальний матеріал, але й сприяє розвитку когнітивних здібностей, мотивації та творчого мислення учнів.

Основні психолого-педагогічні принципи

1. **Мотивація через цікавість.** Навчальні ігри стимулюють інтерес до навчання завдяки інтерактивності, елементам гри та змагальності.
 - **Приклад:** Учень отримує бали за правильні відповіді, що мотивує його проходити нові рівні гри.
2. **Адаптація до рівня учня.** Ефективна навчальна гра враховує індивідуальні особливості учнів, пропонуючи завдання різної складності.
 - **Приклад:** Якщо учень часто помиляється, гра може запропонувати підказку або більш просте завдання.
3. **Залучення до активного навчання.** Гра передбачає активну участь учня, що дозволяє краще засвоювати матеріал порівняно з пасивними методами навчання.
 - **Приклад:** Учень повинен виконувати дії в грі (вибір правильних відповідей, розв'язання задач), а не просто переглядати матеріал [4].
4. **Формування емоційного зв'язку.** Позитивний емоційний досвід під час гри

сприяє кращому запам'ятовуванню інформації.

- **Приклад:** Анімації, нагороди за досягнення, веселі персонажі створюють емоційно приємну атмосферу.

5. **Розвиток когнітивних здібностей.** Навчальні ігри можуть сприяти розвитку логічного мислення, уваги, пам'яті та здатності вирішувати проблеми.

- **Приклад:** Завдання в грі потребують логічного аналізу та пошуку оптимального рішення.

Врахування вікових особливостей

1. **Молодший шкільний вік (6–10 років).** Учні цього віку схильні до гри як природного способу пізнання світу. Для них важливі яскраві візуальні елементи, прості завдання та нагороди.

- **Приклад:** Ігри, де потрібно підбирати відповідні значення тригонометричних функцій до малюнків або анімацій.

2. **Середній шкільний вік (11–14 років).** Учні цього віку вже мають певний рівень абстрактного мислення, тому вони здатні розв'язувати більш складні завдання.

- **Приклад:** Ігри з математичними викликами, які потребують аналізу графіків або розв'язання рівнянь.

3. **Старший шкільний вік (15–18 років).** Для цієї вікової групи важливі практичне застосування знань і змагальний аспект.

- **Приклад:** Ігри, де учні змагаються між собою у швидкості й точності розв'язання задач.

Роль зворотного зв'язку

Навчальна гра повинна забезпечувати миттєвий зворотний зв'язок, що дозволяє учневі одразу побачити, чи правильно він виконав завдання.

- **Позитивний зворотний зв'язок** мотивує і підвищує впевненість у власних силах.
- **Негативний зворотний зв'язок** повинен бути конструктивним і пояснювати помилку.

- **Приклад:** Якщо учень обрав неправильну відповідь, гра показує

правильний розв'язок і пояснює логіку.

Елементи гейміфікації

1. **Рівні складності.** Завдання в грі мають поступово ускладнюватися, що підтримує інтерес учнів.

- **Приклад:** На першому рівні учень обчислює прості значення тригонометричних функцій, а на останньому — розв'язує складні задачі з трикутниками.

2. **Нагороди та досягнення.** Учень може отримувати віртуальні нагороди за успішне виконання завдань.

- **Приклад:** Здобуття медалей за точність, швидкість або послідовність у правильних відповідях.

3. **Змагання**

Ігри з рейтинговими таблицями або можливістю змагатися з іншими учнями сприяють залученню та підвищенню інтересу.

Переваги навчальних ігор

- **Зменшення страху перед помилками:** Гра дозволяє експериментувати без страху «завалити» завдання.
- **Краще засвоєння матеріалу:** Наочність і активна участь сприяють запам'ятовуванню.
- **Формування навичок самостійного навчання:** Учні навчаються самостійно шукати рішення та виправляти свої помилки.

Навчальні ігри, розроблені з урахуванням психолого-педагогічних аспектів, сприяють ефективному засвоєнню тригонометрії, розвитку логічного мислення та формуванню позитивного ставлення до математики. У нашій роботі ці принципи реалізовані через інтерактивні модулі, які забезпечують активне залучення учнів, мотивацію та поступове оволодіння матеріалом.

3. ПРАКТИЧНА ЧАСТИНА

3.1. Вибір технологій для розробки програмного продукту

Вибір технологій для розробки програмного продукту є одним із ключових етапів, що визначає ефективність та якість кінцевого результату. У даному розділі будуть описані технології, які використовувалися для створення інтерактивної гри з тригонометрії.

Вимоги до технологій

Перед початком розробки були визначені основні вимоги до технологій:

- **Інтерактивність:** забезпечення можливості взаємодії користувача з програмою у реальному часі.
- **Кросплатформеність:** доступність програми на різних пристроях, включаючи комп'ютери, планшети та смартфони.
- **Простота використання:** зрозумілий інтерфейс як для учнів, так і для викладачів.
- **Гнучкість і масштабованість:** можливість додавання нових функцій та модулів у майбутньому.
- **Ефективність розробки:** використання сучасних інструментів для прискорення процесу створення програми [5].

Вибір технологій для реалізації

1. Мова програмування

Для розробки продукту було обрано **JavaScript**, оскільки:

- Це одна з найпопулярніших мов програмування у світі, що забезпечує велику спільноту розробників і широкий вибір бібліотек.
- Підтримує інтерактивні елементи, необхідні для створення навчальних ігор.
- Може використовуватися як для клієнтської частини (frontend), так і для серверної частини (backend).

2. Фреймворк для клієнтської частини

Основним інструментом для створення інтерфейсу користувача було обрано

React:

- Забезпечує компонентний підхід, що дозволяє повторно використовувати код і скорочує час розробки.
- Має зручні інструменти для роботи з інтерактивними елементами.
- Підтримує швидке оновлення інтерфейсу без перезавантаження сторінки.

3. Серверна частина

Для обробки запитів і збереження даних використовувався **Node.js** у поєднанні з **Express.js**:

- Node.js дозволяє створювати швидкі та масштабовані сервери, використовуючи ту саму мову програмування, що й для клієнтської частини.
- Express.js надає простий і зручний інструментарій для створення серверного API.

4. База даних

Для зберігання статистики про прогрес учнів було обрано **MongoDB**:

- Це документно-орієнтована база даних, яка забезпечує гнучкість у зберіганні структурованих даних [6].
- Підходить для динамічних ігрових проєктів, оскільки не вимагає попереднього визначення схеми даних.

5. Система управління станом

Для управління станом програми використовувалася бібліотека **Redux**:

- Вона забезпечує централізоване управління даними, що важливо для багатокомпонентного додатка.

6. Стилзація інтерфейсу

Для оформлення інтерфейсу було використано **styled-components**:

- Дозволяє писати CSS у JavaScript, що спрощує роботу з динамічними стилями.

7. Інструменти для тестування

Для тестування програми використовувалися такі інструменти:

- **Jest** — для модульного тестування логіки програми.
- **Cypress** — для тестування користувацького інтерфейсу.

Обґрунтування вибору технологій

Описані інструменти були обрані на основі їхньої надійності, продуктивності та популярності. Вони забезпечують швидку розробку, легке впровадження змін і підтримку інтерактивних елементів, що є важливими для створення ефективного навчального продукту.

Таким чином, обрані технології дозволяють створити сучасний, кросплатформений продукт, який відповідає вимогам інтерактивного навчання. Використання React у поєднанні з Node.js і MongoDB забезпечило реалізацію як клієнтської, так і серверної частини програми, а також інтеграцію всіх необхідних функцій.

На рисунку 3.1 можна проглянути логічну структуру взаємодії взаємодії на сайті (див. рис. 3.1).

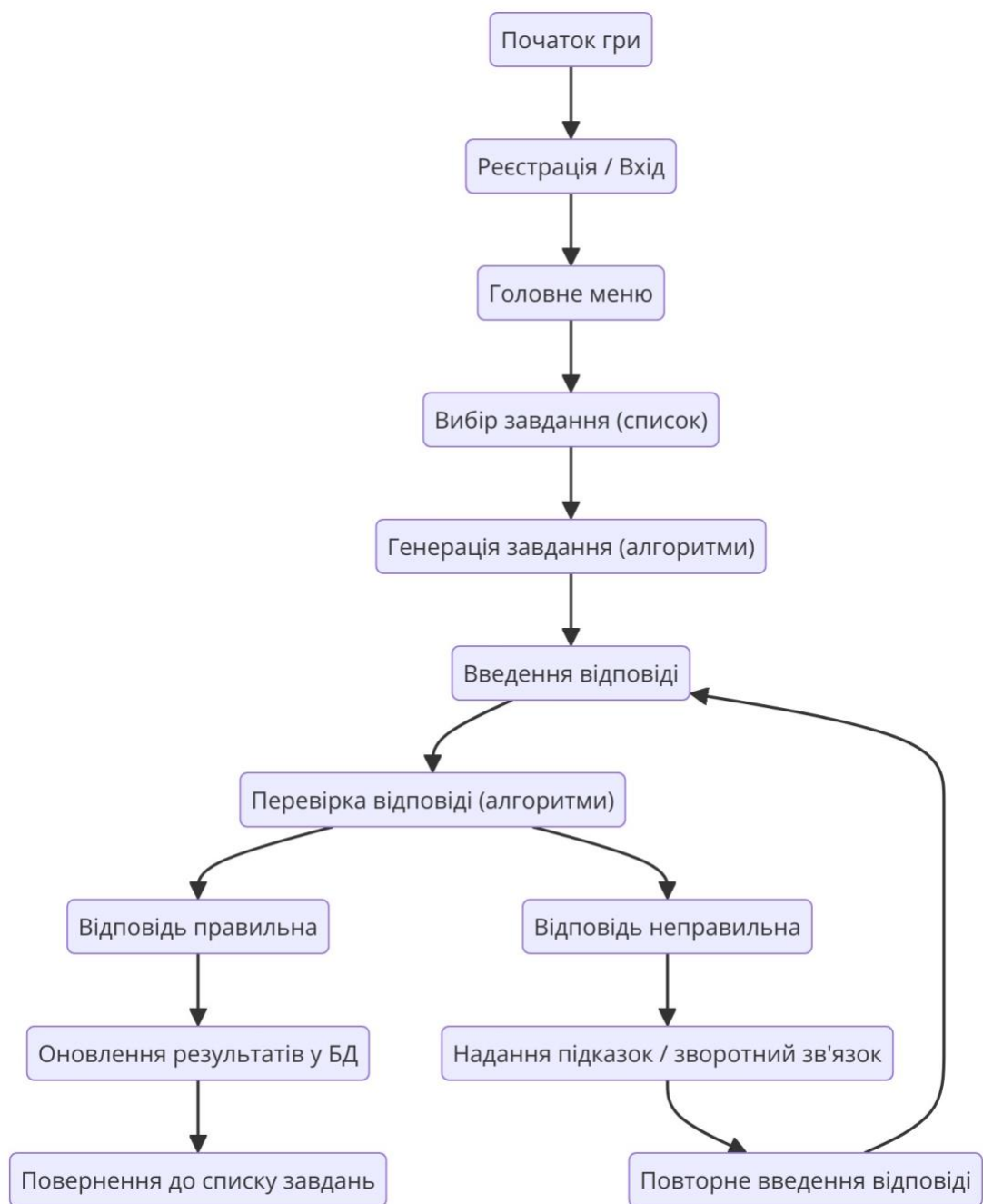


Рисунок 3.1 – Логічна структура проекту

На рисунку 3.2 можна переглянути фізичну структуру проекту (див. рис. 3.2)

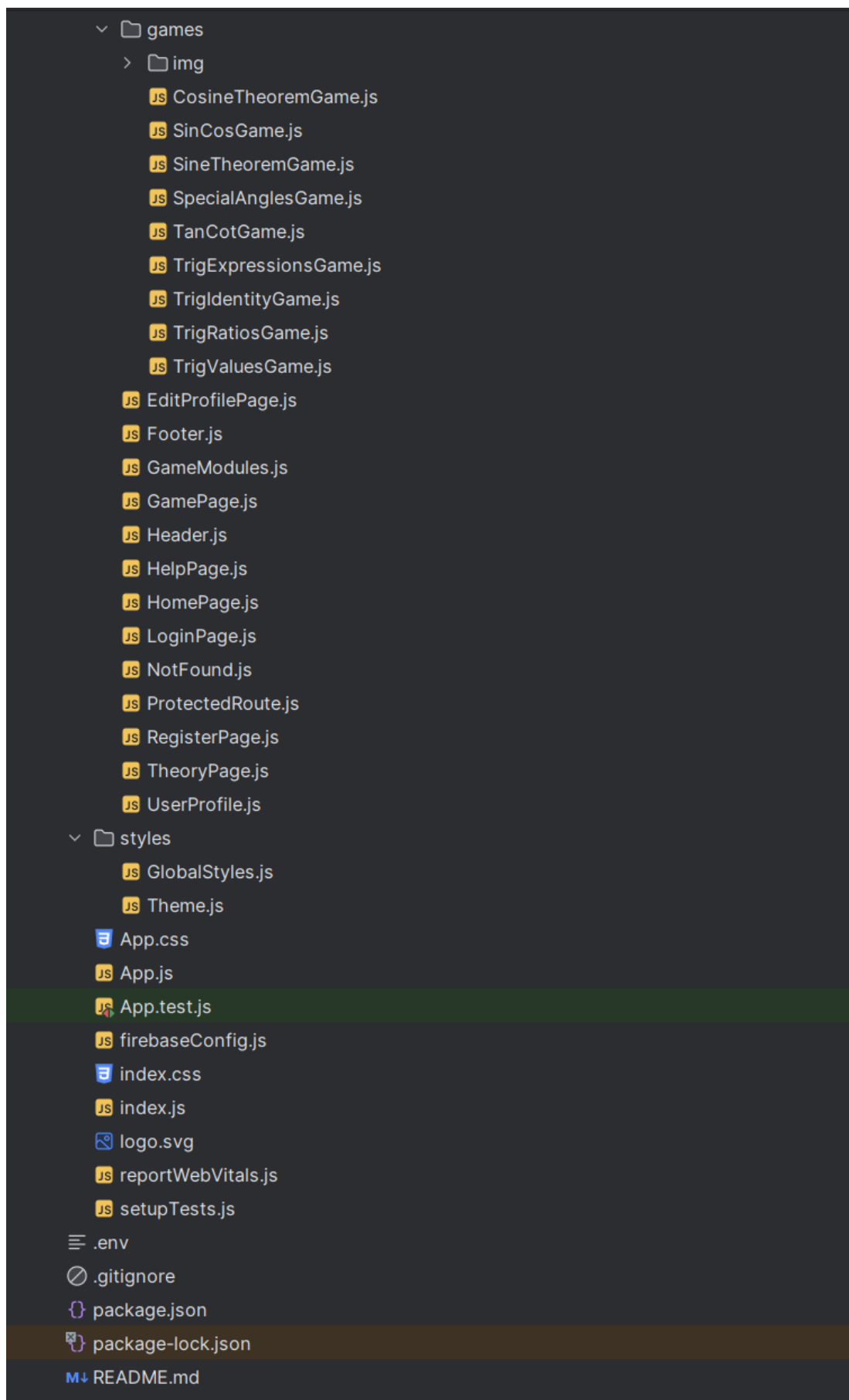


Рисунок 3.2 – Фізична структура проекту

3.2. Реалізація функціональних модулів гри

Розробка інтерактивних модулів гри була здійснена за допомогою мови програмування JavaScript, бібліотеки React для фронтенду та styled-components для стилізації. У цьому розділі розглянуто реалізацію основних функціональних модулів із поясненням коду.

Модуль "Основи тригонометрії"

Цей модуль реалізує базові завдання для ознайомлення користувачів із тригонометричними функціями. Приклад коду:

```
import React, { useState } from 'react';
import styled from 'styled-components';

const tasks = [
  { question: 'Обчисліть значення  $\sin(30^\circ)$ ', correctAnswer: '0.5' },
  { question: 'Обчисліть значення  $\cos(60^\circ)$ ', correctAnswer: '0.5' },
  { question: 'Обчисліть значення  $\operatorname{tg}(45^\circ)$ ', correctAnswer: '1' },
];

const BasicTrigonometryModule = () => {
  const [currentTaskIndex, setCurrentTaskIndex] = useState(0);
  const [userAnswer, setUserAnswer] = useState('');
  const [feedback, setFeedback] = useState('');

  const handleSubmit = () => {
    const isCorrect = userAnswer === tasks[currentTaskIndex].correctAnswer;
    setFeedback(isCorrect ? 'Правильно!' : `Неправильно. Правильна відповідь: ${tasks[currentTaskIndex].correctAnswer}`);

    if (isCorrect && currentTaskIndex < tasks.length - 1) {
      setCurrentTaskIndex(currentTaskIndex + 1);
      setUserAnswer('');
    }
  };

  return (
    <ModuleContainer>
```

```

    <Question>{tasks[currentTaskIndex].question}</Question>
    <Input
      type="text"
      value={userAnswer}
      onChange={(e) => setUserAnswer(e.target.value)}
      placeholder="Введіть відповідь"
    />
    <SubmitButton onClick={handleSubmit}>Готово</SubmitButton>
    {feedback && <FeedbackMessage>{feedback}</FeedbackMessage>}
  </ModuleContainer>
);
};

export default BasicTrigonometryModule;

const ModuleContainer = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 20px;
`;

const Question = styled.h2`
  margin-bottom: 20px;
`;

const Input = styled.input`
  padding: 10px;
  margin-bottom: 20px;
`;

const SubmitButton = styled.button`
  padding: 10px 20px;
`;

const FeedbackMessage = styled.p`
  margin-top: 20px;
  font-size: 16px;

```

```
color: green;
`;
```

Модуль "Розв'язування трикутників"

Цей модуль дозволяє обчислювати сторони та кути трикутника за допомогою теорем синусів і косинусів. Приклад коду:

```
import React, { useState } from 'react';
import styled from 'styled-components';

const TriangleSolver = () => {
  const [sideA, setSideA] = useState("");
  const [sideB, setSideB] = useState("");
  const [angleC, setAngleC] = useState("");
  const [result, setResult] = useState(null);

  const calculateSideC = () => {
    const angleRad = (Math.PI / 180) * angleC;
    const sideC = Math.sqrt(
      Math.pow(Number(sideA), 2) +
      Math.pow(Number(sideB), 2) -
      2 * Number(sideA) * Number(sideB) * Math.cos(angleRad)
    );
    setResult(sideC.toFixed(2));
  };

  return (
    <SolverContainer>
      <InputContainer>
        <label>Сторона A:</label>
        <Input type="number" value={sideA} onChange={(e) => setSideA(e.target.value)} />
      </InputContainer>
      <InputContainer>
        <label>Сторона B:</label>
        <Input type="number" value={sideB} onChange={(e) => setSideB(e.target.value)} />
      </InputContainer>
      <InputContainer>
        <label>Кут C (градуси):</label>
        <Input type="number" value={angleC} onChange={(e) => setAngleC(e.target.value)} />
      </InputContainer>
    </SolverContainer>
  );
};
```

```

    </InputContainer>
    <Button onClick={calculateSideC}>Обчислити сторону C</Button>
    {result && <Result>Сторона C: {result}</Result>}
  </SolverContainer>
);
};

```

```
export default TriangleSolver;
```

```

const SolverContainer = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
`;

```

```

const InputContainer = styled.div`
  margin-bottom: 15px;
`;

```

```

const Input = styled.input`
  margin-left: 10px;
  padding: 5px;
`;

```

```

const Button = styled.button`
  padding: 10px 20px;
  margin-top: 20px;
`;

```

```

const Result = styled.div`
  margin-top: 20px;
  font-size: 18px;
  color: blue;
`;

```

Модуль "Графіки тригонометричних функцій"

Цей модуль дозволяє візуалізувати графіки синуса, косинуса та тангенса.

```

import React from 'react';
import { Line } from 'react-chartjs-2';

```

```

const TrigGraph = () => {
  const data = {
    labels: Array.from({ length: 361 }, (_, i) => i - 180), // Градуси від -180 до 180
    datasets: [
      {
        label: 'sin(x)',
        data: Array.from({ length: 361 }, (_, i) => Math.sin(((i - 180) * Math.PI) / 180)),
        borderColor: 'blue',
        fill: false,
      },
      {
        label: 'cos(x)',
        data: Array.from({ length: 361 }, (_, i) => Math.cos(((i - 180) * Math.PI) / 180)),
        borderColor: 'red',
        fill: false,
      },
    ],
  };

  return (
    <div>
      <h2>Графіки тригонометричних функцій</h2>
      <Line data={data} />
    </div>
  );
};

export default TrigGraph;

```

Описані вище модулі інтегруються в єдиний додаток через роутинг. Використання сучасних бібліотек і підходів до розробки забезпечує високу продуктивність і зручність програми. Кожен модуль може бути розширений для додавання нового функціоналу в майбутньому.

3.3. Інструкція для користувача

1. Перегляд основної інформації на сайті (див. рис. 3.3)

1.1. Головна сторінка

- Після переходу на сайт (наприклад, <https://trigonometrygame.com>) ви потрапляєте на головну сторінку.
- Головна сторінка включає:
 - Логотип платформи "Trigonometry Game".
 - Привітальне повідомлення: "Ласкаво просимо до навчальної платформи! Вивчайте тригонометрію за допомогою інтерактивних ігор та теорії!".
 - Чотири основні кнопки: "Теорія", "Ігрові модулі", "Мій профіль", "Допомога".
 - У верхньому меню відображаються розділи: "Головна", "Теорія", "Ігри", "Довідка", "Профіль", а також кнопка "Логін"/"Вийти".

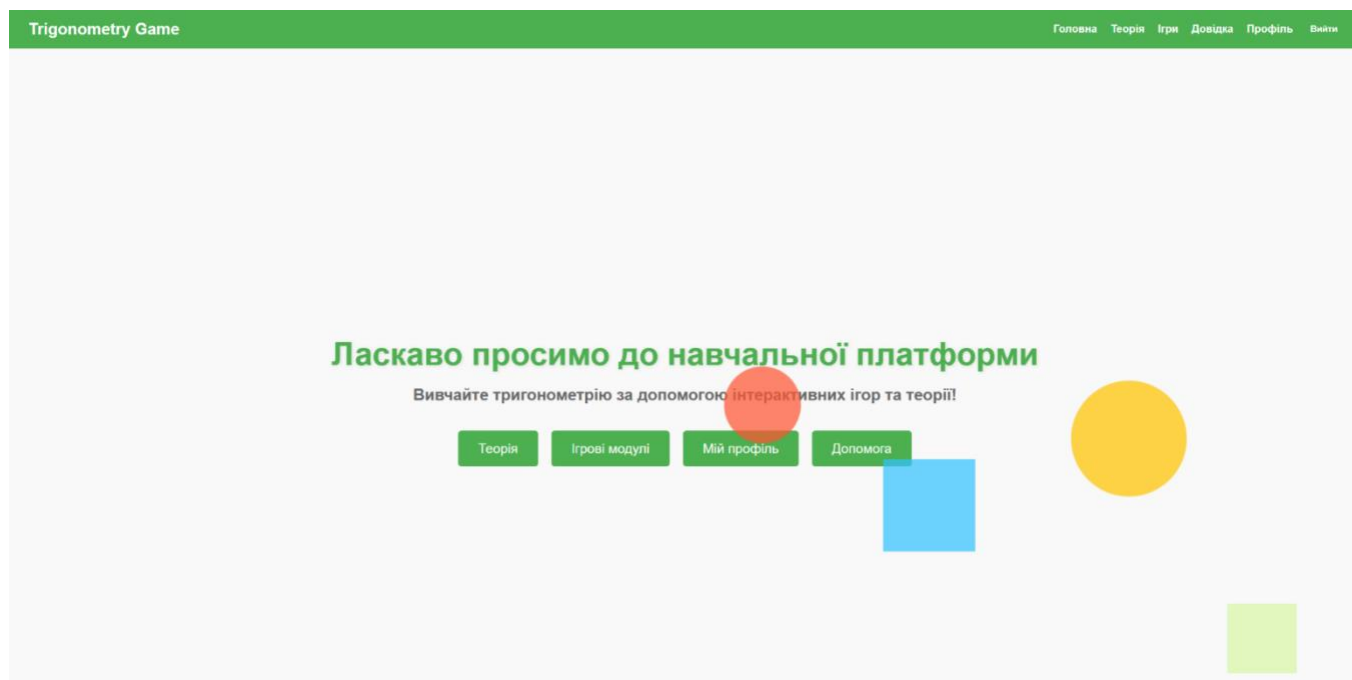


Рисунок 3.3 – Головна сторінка

1.2. Навігація

- Кнопки у верхньому меню дозволяють швидко перемикатися між основними розділами сайту:
 - **"Головна"** – повернення до стартової сторінки.

- **"Теорія"** – доступ до навчальних матеріалів.
- **"Ігри"** – вибір навчальних ігор.
- **"Довідка"** – відповіді на поширені запитання.
- **"Профіль"** – перегляд і редагування особистих даних.

2. Реєстрація та вхід до системи (див. рис. 3.2)

2.1. Вхід

- Натисніть кнопку "Логін" у верхньому меню.
- У формі авторизації введіть:
 - Електронну пошту.
 - Пароль.
- Натисніть "Увійти", щоб увійти до свого акаунта.

Рисунок 3.4 – Форма логіну

2.2. Реєстрація

- Якщо у вас немає акаунта, натисніть "Зареєструватися" внизу форми входу.
- Заповніть форму реєстрації:
 - Ім'я та прізвище.
 - Електронну пошту.
 - Придумайте пароль.
- Після реєстрації ви автоматично отримаєте доступ до особистого кабінету.

3. Розділ "Теорія" (див. рис. 3.5)

3.1. Доступ до матеріалів

- Перейдіть до розділу "Теорія", натиснувши на відповідний пункт у верхньому меню.
- У цьому розділі представлені теоретичні матеріали з основ тригонометрії, що включають:
 - Основні тригонометричні функції (синус, косинус, тангенс, котангенс).
 - Формули тригонометричних співвідношень.
 - Теореми синусів і косинусів.

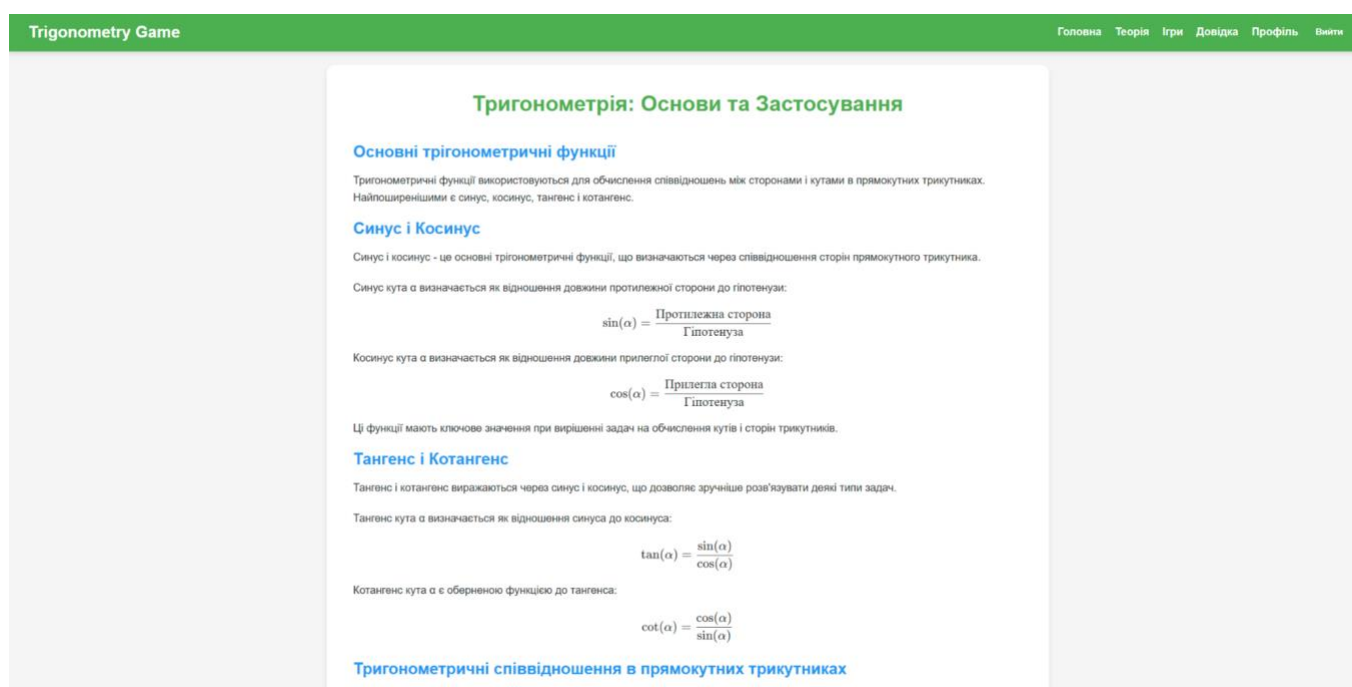


Рисунок 3.5 – Сторінка з теорією

3.2. Структура сторінки

- Матеріали поділені на тематичні блоки з заголовками.
- Формули відображаються у зручному форматі, що дозволяє легко їх зрозуміти.
- Додані приклади з розв'язанням задач.

4. Вибір та проходження ігрових модулів

4.1. Вибір модуля

- У розділі "Ігри" ви знайдете список модулів (див. рис. 3.6), таких як:
 - "Синус та косинус гострого кута"

- "Тангенс та котангенс гострого кута"
- "Основна тригонометрична тотожність"
- "Значення тригонометричних функцій"
- "Теорема синусів"
- "Теорема косинусів".

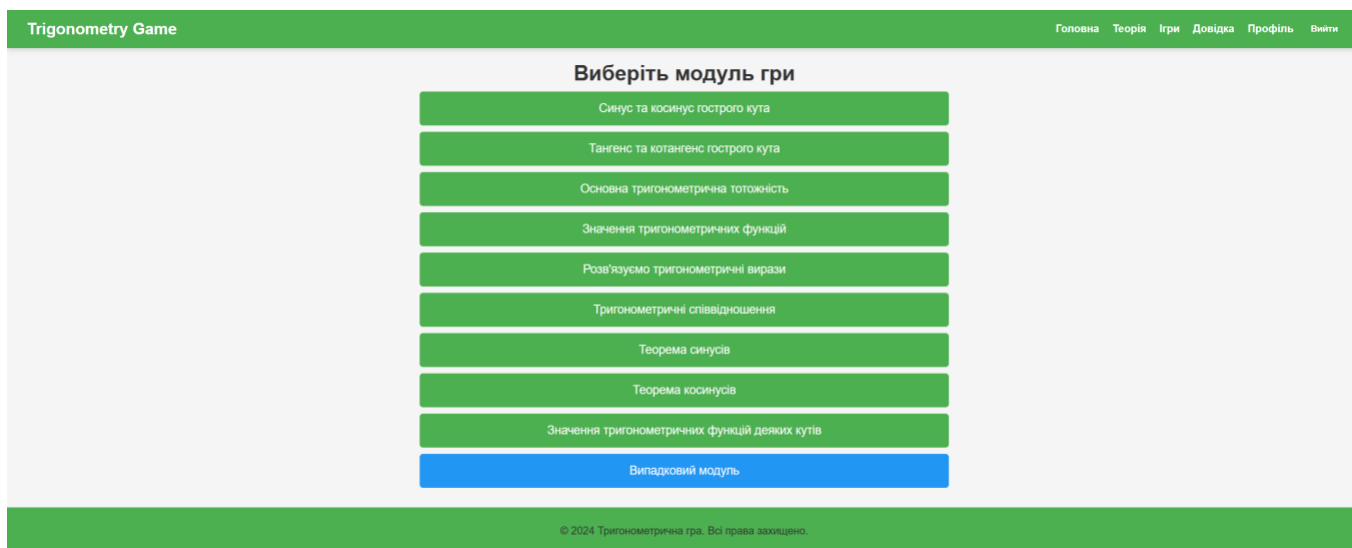


Рисунок 3.6 – Модулі гри

4.2. Процес гри

- Обравши модуль, ви побачите завдання, наприклад:
 - "Обчисліть синус кута A ".
- Інтерфейс гри включає:
 - Зображення задачі (наприклад, трикутника з позначеними сторонами).
 - Поле для введення відповіді.
 - Кнопку "Готово".

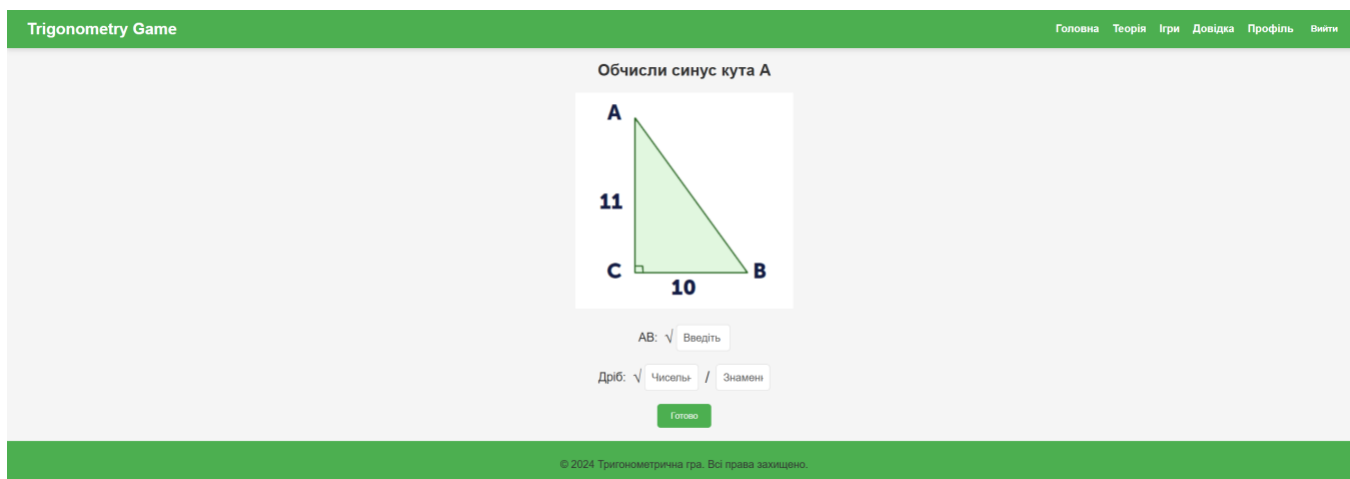


Рисунок 3.7 – Приклад гри

4.3. Оцінювання

- Після введення відповіді система повідомить:
 - "Правильно!" – якщо відповідь вірна (див. рис. 3.8).

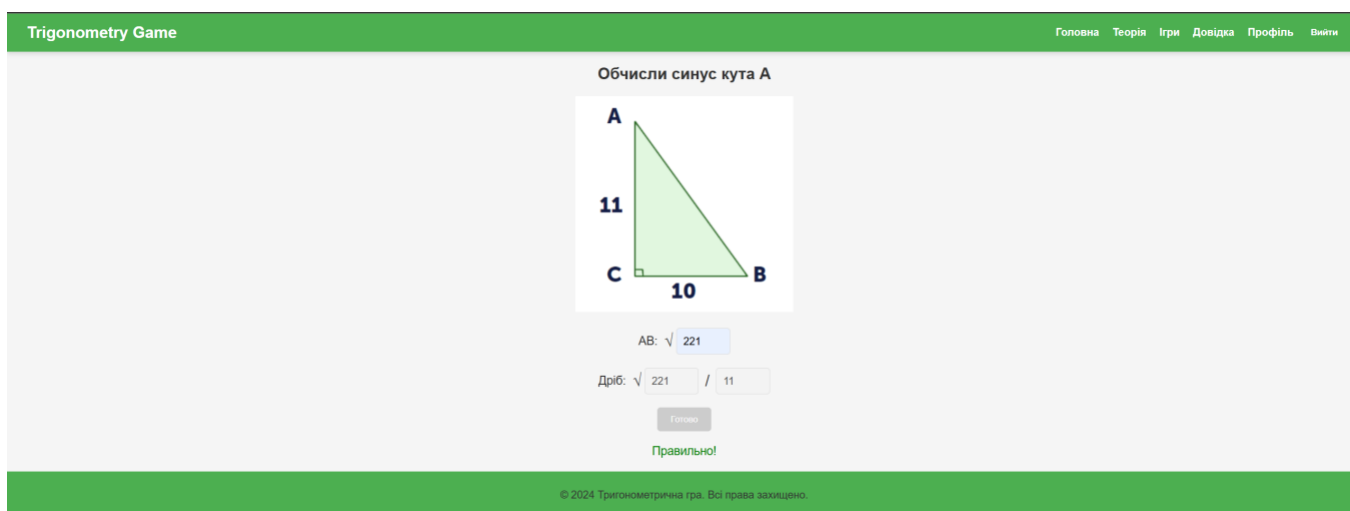


Рисунок 3.8 – Демонстрація правильної відповіді

- "Неправильно. Правильна відповідь: X" – якщо відповідь невірна (див. рис. 3.9).

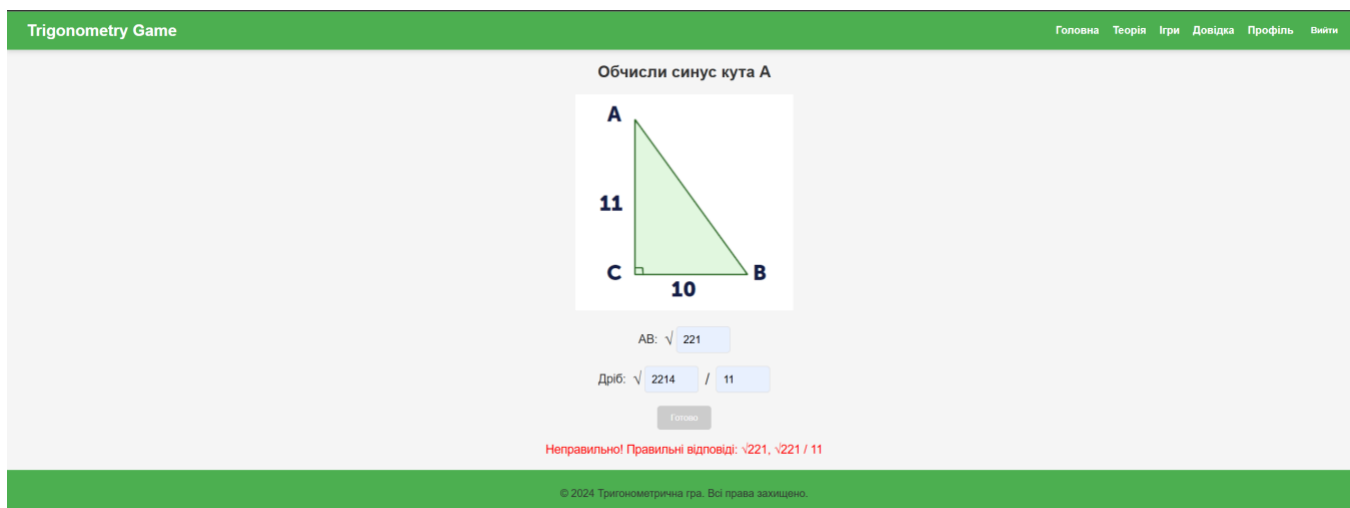


Рисунок 3.9 – Демонстрація неправильної відповіді

5. Профіль користувача (див. рис. 3.10)

5.1. Перегляд інформації

- У розділі "Профіль" відображаються:
 - Ваше ім'я та прізвище.
 - Група чи статус (наприклад, "Студент").
 - Інформація про себе.

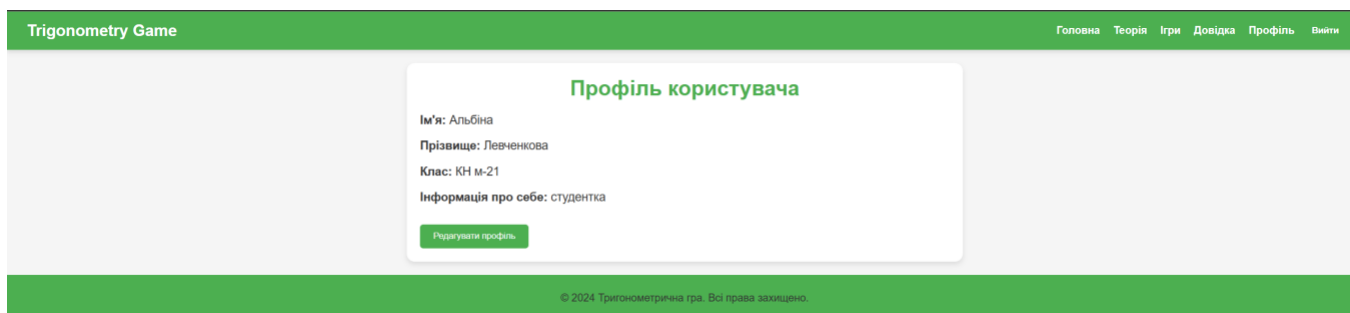


Рисунок 3.10 – Профіль користувача

5.2. Редагування профілю

- Натисніть "Редагувати профіль", щоб змінити дані (див. рис. 3.11).

Рисунок 3.11 – Редагування профілю користувача

6. Довідка та підтримка (див. рис. 3.12)

6.1. Розділ "Довідка"

- Відповіді на найчастіші запитання:
 - Як зареєструватися?
 - Як увійти до системи?
 - Як проходити модулі?

6.2. Контакти підтримки

- Якщо виникли технічні питання, звертайтеся за адресою: support@trigonometrygame.com.

Рисунок 3.12 – Сторінка довідки

ВИСНОВКИ

У ході виконання дипломної роботи було успішно розроблено інтерактивну навчальну платформу "Trigonometry Game", яка дозволяє студентам поглиблювати свої знання з тригонометрії за допомогою інтерактивних теоретичних матеріалів та ігрових модулів. Робота охоплювала всі етапи розробки, починаючи від аналізу методів викладання математики, проєктування архітектури системи до програмної реалізації платформи та її тестування.

Основні результати роботи включають:

1. Аналіз сучасних технологій та вибір інструментів для розробки:

- Проведено аналіз існуючих платформ для інтерактивного навчання.
- Вибрано React.js для фронтенду, Node.js для бекенду, а також Firebase для зберігання даних і автентифікації користувачів, що забезпечило сучасний підхід до розробки веб-додатків.

2. Проєктування та реалізація архітектури платформи:

- Розроблено структуру сайту, яка включає головну сторінку, розділи "Теорія", "Ігри", "Профіль" і "Довідка".
- Створено адаптивний дизайн, який забезпечує коректне відображення платформи на різних пристроях, включаючи мобільні телефони, планшети та комп'ютери.
- Забезпечено інтуїтивно зрозумілу навігацію для користувачів завдяки верхньому меню та структурованій організації контенту.

3. Реалізація функціональних модулів гри:

- Створено кілька інтерактивних ігрових модулів, таких як "Синус та косинус гострого кута", "Основна тригонометрична тотожність", "Теорема синусів" та інші.
- Реалізовано перевірку правильності відповідей із динамічним наданням зворотного зв'язку (похвала за правильні відповіді або підказки при помилках).
-

4. Теоретичний розділ для самостійного навчання:

- Розроблено структуровані навчальні матеріали з основ тригонометрії, які включають базові формули, властивості тригонометричних функцій та приклади їх застосування.

5. Розділ профілю користувача та статистики:

- Забезпечено можливість реєстрації та входу на платформу за допомогою електронної пошти.
- Реалізовано відображення статистики проходження ігрових модулів, що дозволяє користувачам відслідковувати свій прогрес у навчанні.

6. Тестування та забезпечення стабільності роботи платформи:

- Проведено тестування роботи платформи на різних пристроях і в популярних браузерах, таких як Chrome, Firefox та Safari.
- Реалізовано захист даних користувачів, включаючи шифрування паролів та безпечне зберігання особистої інформації.

Розроблена платформа "Trigonometry Game" є сучасним інструментом, який підвищує мотивацію учнів до вивчення тригонометрії завдяки використанню інтерактивних ігрових методів та доступних теоретичних матеріалів. Платформа може бути використана як у класному навчанні під керівництвом викладача, так і для самостійного вивчення теми.

У майбутньому платформа може бути розширена шляхом:

- Додавання нових ігрових модулів з інших розділів математики (алгебра, геометрія тощо).
- Інтеграції з системами управління навчанням (LMS), такими як Moodle.
- Впровадження рейтингової системи для стимулювання конкурентного духу серед учнів.
- Розширення функціональності теоретичних матеріалів за рахунок інтерактивних симуляцій і відеоуроків.

Розроблений проєкт підтвердив ефективність використання інтерактивних технологій у навчанні, відкривши нові можливості для покращення викладання математики та мотивації студентів до вивчення складних тем.

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Бевз Г. П., Бевз В. Г. Геометрія: Підручник для 10-11 класів. Київ: Генеза, 2021.
2. Васильєв О. М. Програмування. Bohdan Books, 2022.
3. Литвиненко О. А. Використання ІКТ у викладанні математики. Львів: Видавництво ЛНУ, 2020.
4. Schildt, Herbert: A Beginner's Guide. McGraw-Hill Education, 2022.
5. Downey, Allen B., and Chris Mayfield: How to Think Like a Computer Scientist. O'Reilly Media, 2019.
6. Лобачевський М. І. Тригонометрія: Теорія та практичні задачі. Харків: Видавництво ХНУ, 2020.
7. Придатко О. В. Основи програмування: Курс лекцій / О. В. Придатко, О. В. Хлевной, Н. Є. Бурак. Львів: ЛДУ БЖД, 2019.
8. Іваненко Т. В. Інтерактивні методи навчання в математиці. Київ: Освіта, 2021.
9. Матвієнко А. П., Коваленко В. А. Візуалізація математичних понять: Інтерактивні підходи. Харків: Видавництво ХНУ, 2020.
10. Буряк О. П. Алгебра і початки аналізу: Підручник для 10-11 класів. Освіта, 2021.
11. Ханна Дж. Алгоритми і структури даних: Практичне керівництво. Переклад з англійської. Київ: КМ-Букс, 2021.
12. Нейман С. В. Методи викладання математики в середній школі. Львів: Видавництво ЛНУ, 2020.
13. 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.

CosineTheoremGame.js

```
import React, { useState } from 'react';
import styled from 'styled-components';
import { useNavigate } from 'react-router-dom';

import image1 from './img/image15.png';
import image2 from './img/image16.png';
import image3 from './img/image17.png';
import image4 from './img/image18.png';
import image5 from './img/image19.png';

const tasks = [
  {
    image: image1,
    question: 'Обчисли довжину сторони AC (відповідь округли до цілих)',
    correctAnswer: '7'
  },
  {
    image: image2,
    question: 'Обчисли довжину сторони PR (відповідь округли до цілих)',
    correctAnswer: '14'
  },
  {
    image: image3,
    question: 'Обчисли довжину сторони SO (відповідь округли до цілих)',
    correctAnswer: '29'
  },
  {
    image: image4,
    question: 'Обчисли довжину сторони BC (відповідь округли до цілих)',
    correctAnswer: '10'
  },
  {
    image: image5,
    question: 'Обчисли довжину сторони PQ (відповідь округли до цілих)',
    correctAnswer: '8'
  }
]
```

```

    }
  ];

const CosineTheoremGame = () => {
  const [currentTaskIndex, setCurrentTaskIndex] = useState(0);
  const [userAnswers, setUserAnswers] = useState([]);
  const [timeSpent, setTimeSpent] = useState(0);
  const [startTime] = useState(Date.now());
  const [feedback, setFeedback] = useState("");
  const [feedbackColor, setFeedbackColor] = useState(""); // Стан для кольору тексту
  const navigate = useNavigate();

  const handleAnswerSubmit = (userAnswer) => {
    const isCorrect = userAnswer === tasks[currentTaskIndex].correctAnswer;
    const updatedAnswers = [...userAnswers, isCorrect];
    setUserAnswers(updatedAnswers);

    if (isCorrect) {
      setFeedback('Правильна відповідь!');
      setFeedbackColor('green');
    } else {
      setFeedback(`Неправильна відповідь! Правильна відповідь:
        ${tasks[currentTaskIndex].correctAnswer}`);
      setFeedbackColor('red');
    }
  }

  setTimeout(() => {
    setFeedback("");
    setFeedbackColor(""); // Скидаємо колір тексту
    document.querySelector('input').value = ""; // Очищення поля введення

    if (currentTaskIndex < tasks.length - 1) {
      setCurrentTaskIndex(currentTaskIndex + 1);
    } else {
      const endTime = Date.now();
      const totalTimeSpent = Math.floor((endTime - startTime) / 1000);
      setTimeSpent(totalTimeSpent);
    }
  }, 1000);
};

```

```

const correctAnswers = updatedAnswers.reduce((acc, answer) => {
  return acc + (answer ? 1 : 0);
}, 0);

const statistics = JSON.parse(localStorage.getItem('statistics')) || [];
statistics.push({ module: 'CosineTheoremGame', timeSpent: totalTimeSpent, correctAnswers });
localStorage.setItem('statistics', JSON.stringify(statistics));

alert(`Модуль завершено!\nЧас:   ${totalTimeSpent}   секунд\nПравильних   відповідей:
${correctAnswers}/${tasks.length}`);

  navigate('/games');
}
}, 1500);
};

return (
  <GameContainer>
    <Question>{tasks[currentTaskIndex].question}</Question>
    <ImageContainer>
      <img src={tasks[currentTaskIndex].image} alt={`Зображення ${currentTaskIndex + 1}`} />
    </ImageContainer>
    <AnswerInput
      type="text"
      placeholder="Введіть вашу відповідь"
      onPress={(e) => {
        if (e.key === 'Enter') {
          handleAnswerSubmit(e.target.value.trim());
        }
      }}
    />
    <SubmitButton                                onClick={()                                =>
handleAnswerSubmit(document.querySelector('input').value.trim())}>
      Готово
    </SubmitButton>
    {feedback} && <FormattedMessage                                style={{                                color:                                feedbackColor
}}>{feedback}</FormattedMessage>
  </GameContainer>

```

```

);
};

export default CosineTheoremGame;

const GameContainer = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 20px;
  text-align: center;
`;

const Question = styled.h2`
  margin-bottom: 20px;
  color: #333;
`;

const ImageContainer = styled.div`
  margin-bottom: 20px;
`;

const AnswerInput = styled.input`
  padding: 10px;
  font-size: 16px;
  border: 2px solid #ddd;
  border-radius: 5px;
  margin-bottom: 20px;
`;

const SubmitButton = styled.button`
  padding: 10px 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  &:hover {

```

```

    background-color: #45a049;
  }
`;

```

```

const FeedbackMessage = styled.div`
  margin-top: 20px;
  font-size: 18px;
  color: #333;
`;

```

SinCosGame.js

```

import React, { useState } from 'react';
import styled from 'styled-components';
import { useNavigate } from 'react-router-dom';

import image1 from './img/image.png';
import image2 from './img/image1.png';
import image3 from './img/image2.png';
import image4 from './img/image3.png';
import image5 from './img/image4.png';

const tasks = [
  { image: image1, question: 'Обчисли синус кута A', correctAnswer: '0.73' },
  { image: image2, question: 'Обчисли косинус кута P', correctAnswer: '0.87' },
  { image: image3, question: 'Обчисли синус кута M', correctAnswer: '0.77' },
  { image: image4, question: 'Обчисли синус кута B', correctAnswer: '0.64' },
  { image: image5, question: 'Обчисли косинус кута P', correctAnswer: '0.92' }
];

const SinCosGame = () => {
  const [currentTaskIndex, setCurrentTaskIndex] = useState(0);
  const [userAnswers, setUserAnswers] = useState([]);
  const [feedback, setFeedback] = useState("");
  const [feedbackColor, setFeedbackColor] = useState("");
  const [startTime] = useState(Date.now());
  const navigate = useNavigate();

  const handleAnswerSubmit = (userAnswer) => {
    const correctAnswer = tasks[currentTaskIndex].correctAnswer;

```

```

const isCorrect = userAnswer === correctAnswer;

setFeedbackColor(isCorrect ? 'green' : 'red');
setFeedback(isCorrect ? 'Правильно!' : `Неправильно! Правильна відповідь: ${correctAnswer}`);

const updatedAnswers = [...userAnswers, isCorrect];
setUserAnswers(updatedAnswers);

setTimeout(() => {
  setFeedback("");
  setFeedbackColor("");

  if (currentTaskIndex < tasks.length - 1) {
    setCurrentTaskIndex(currentTaskIndex + 1);
  } else {
    const endTime = Date.now();
    const totalTimeSpent = Math.floor((endTime - startTime) / 1000);
    const correctAnswers = updatedAnswers.reduce((acc, answer) => acc + (answer ? 1 : 0), 0);

    alert(`Модуль завершено!\nЧас:   ${totalTimeSpent}   секунд\nПравильних   відповідей:
    ${correctAnswers}/${tasks.length}`);

    navigate('/games');
  }
}, 1500);
};

return (
  <GameContainer>
    <Question>{tasks[currentTaskIndex].question}</Question>
    <ImageContainer>
      <img src={tasks[currentTaskIndex].image} alt={`Зображення ${currentTaskIndex + 1}`} />
    </ImageContainer>
    <AnswerInput
      type="text"
      placeholder="Введіть вашу відповідь"
      onKeyPress={(e) => {
        if (e.key === 'Enter') {

```

```

        handleAnswerSubmit(e.target.value.trim());
        e.target.value = "";
    }
}
/>
<SubmitButton onClick={() => {
    const input = document.querySelector('input');
    handleAnswerSubmit(input.value.trim());
    input.value = "";
}}>
    Гомого
</SubmitButton>
    {feedback && <FeedbackMessage color={feedbackColor}>{feedback}</FeedbackMessage>}
</GameContainer>
);
};

export default SinCosGame;

const GameContainer = styled.div`
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: 20px;
    text-align: center;
`;

const Question = styled.h2`
    margin-bottom: 20px;
    color: #333;
`;

const ImageContainer = styled.div`
    margin-bottom: 20px;
`;

const AnswerInput = styled.input`
    padding: 10px;

```

```
font-size: 16px;
border: 1px solid #ddd;
border-radius: 5px;
margin-bottom: 20px;
`;
```

```
const SubmitButton = styled.button`
padding: 10px 20px;
background-color: #4CAF50;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
&:hover {
background-color: #45a049;
}
`;
```

```
const FeedbackMessage = styled.div`
margin-top: 20px;
font-size: 18px;
color: ${props => props.color};
`;
```

SineTheoremGame.js

```
import React, { useState } from 'react';
import styled from 'styled-components';
import { useNavigate } from 'react-router-dom';
```

```
import image1 from './img/image10.png';
import image2 from './img/image11.png';
import image3 from './img/image12.png';
import image4 from './img/image13.png';
import image5 from './img/image14.png';
```

```
const tasks = [
{
image: image1,
question: 'Обчисли градусну міру кута A (відповідь округли до цілих)',
```

```

    correctAnswer: '45'
  },
  {
    image: image2,
    question: 'Обчисли довжину сторони TR (відповідь округли до цілих)',
    correctAnswer: '8'
  },
  {
    image: image3,
    question: 'Обчисли градусну міру кута L (відповідь округли до цілих)',
    correctAnswer: '52'
  },
  {
    image: image4,
    question: 'Обчисли довжину сторони FS (відповідь округли до цілих)',
    correctAnswer: '24'
  },
  {
    image: image5,
    question: 'Обчисли градусну міру кута B (відповідь округли до цілих)',
    correctAnswer: '37'
  }
];

```

```

const SineTheoremGame = () => {
  const [currentTaskIndex, setCurrentTaskIndex] = useState(0);
  const [userAnswers, setUserAnswers] = useState([]);
  const [timeSpent, setTimeSpent] = useState(0);
  const [startTime] = useState(Date.now());
  const [feedback, setFeedback] = useState("");
  const [feedbackColor, setFeedbackColor] = useState(""); // Додаємо стан для кольору тексту
  const navigate = useNavigate();

  const handleAnswerSubmit = (userAnswer) => {
    const isCorrect = userAnswer === tasks[currentTaskIndex].correctAnswer;
    const updatedAnswers = [...userAnswers, isCorrect];
    setUserAnswers(updatedAnswers);
  };

```

```

    if (isCorrect) {
      setFeedback('Правильна відповідь!');
      setFeedbackColor('green');
    } else {
      setFeedback(`Неправильна відповідь! Правильна відповідь:
      ${tasks[currentTaskIndex].correctAnswer}`);
      setFeedbackColor('red');
    }

    setTimeout(() => {
      setFeedback("");
      setFeedbackColor("");
      document.querySelector('input').value = ""; // Очищення поля введення

      if (currentTaskIndex < tasks.length - 1) {
        setCurrentTaskIndex(currentTaskIndex + 1);
      } else {
        const endTime = Date.now();
        const totalTimeSpent = Math.floor((endTime - startTime) / 1000);
        setTimeSpent(totalTimeSpent);

        const correctAnswers = updatedAnswers.reduce((acc, answer) => {
          return acc + (answer ? 1 : 0);
        }, 0);

        // Збереження результатів у статистику (місцева імплементація)
        const statistics = JSON.parse(localStorage.getItem('statistics')) || [];
        statistics.push({ module: 'SineTheoremGame', timeSpent: totalTimeSpent, correctAnswers });
        localStorage.setItem('statistics', JSON.stringify(statistics));

        // Відображення результатів на екрані
        alert(`Модуль завершено!\nЧас:  ${totalTimeSpent} секунд\nПравильних відповідей:
        ${correctAnswers}/${tasks.length}`);

        // Після показу результатів перенаправляємо користувача на сторінку з модулями
        navigate('/games');
      }
    }, 1500);

```

```

};

return (
  <GameContainer>
    <Question>{tasks[currentTaskIndex].question}</Question>
    <ImageContainer>
      <img src={tasks[currentTaskIndex].image} alt={`Зображення ${currentTaskIndex + 1}`} />
    </ImageContainer>
    <AnswerInput
      type="text"
      placeholder="Введіть вашу відповідь"
      onPress={
        (e) => {
          if (e.key === 'Enter') {
            handleAnswerSubmit(e.target.value.trim());
          }
        }
      }
    />
    <SubmitButton
      style={styles.submitButton}
      onClick={
        () => {
          handleAnswerSubmit(document.querySelector('input').value.trim())
        }
      }
    />
    <FeedbackMessage
      style={
        {
          color: feedbackColor
        }
      }
    />
  </GameContainer>
);
};

export default SineTheoremGame;

const GameContainer = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 20px;
  text-align: center;
`;

const Question = styled.h2`

```

```
margin-bottom: 20px;
color: #333;
`;
```

```
const ImageContainer = styled.div`
  margin-bottom: 20px;
`;
```

```
const AnswerInput = styled.input`
  padding: 10px;
  font-size: 16px;
  border: 2px solid #ddd;
  border-radius: 5px;
  margin-bottom: 20px;
`;
```

```
const SubmitButton = styled.button`
  padding: 10px 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  &:hover {
    background-color: #45a049;
  }
`;
```

```
const FeedbackMessage = styled.div`
  margin-top: 20px;
  font-size: 18px;
  color: #333;
`;
```

TanCotGame.js

```
import React, { useState } from 'react';
import styled from 'styled-components';
import { useNavigate } from 'react-router-dom';
```

```
import image1 from './img/image20.png'; // Заміни на правильний шлях до зображення
import image2 from './img/image21.png'; // Заміни на правильний шлях до зображення
import image3 from './img/image22.png'; // Заміни на правильний шлях до зображення
```

```
const tasks = [
  {
    image: image1,
    question: 'Вибери значення тангенса кута A',
    options: ['0.67', '1.5', '0.56', '0.83'],
    correctAnswer: '1.5'
  },
  {
    image: image2,
    question: 'Вибери значення котангенса кута N',
    options: ['1.25', '0.8', '0.62', '0.78'],
    correctAnswer: '0.8'
  },
  {
    image: image3,
    question: 'Вибери значення тангенса кута P',
    options: ['2.4', '0.42', '0.92', '0.38'],
    correctAnswer: '0.92'
  }
];
```

```
const TangentCotangentGame = () => {
  const [currentTaskIndex, setCurrentTaskIndex] = useState(0);
  const [userAnswers, setUserAnswers] = useState([]);
  const [feedback, setFeedback] = useState("");
  const navigate = useNavigate();

  const handleAnswerSubmit = (userAnswer) => {
    const isCorrect = userAnswer === tasks[currentTaskIndex].correctAnswer;
    const updatedAnswers = [...userAnswers, isCorrect];
    setUserAnswers(updatedAnswers);

    if (isCorrect) {
      setFeedback({ message: 'Правильна відповідь!', color: 'green' });
    }
  };
};
```

```

    } else {
      setFeedback({ message: `Неправильна відповідь! Правильна відповідь:
${tasks[currentTaskIndex].correctAnswer}`, color: 'red' });
    }

    setTimeout(() => {
      setFeedback("");
      if (currentTaskIndex < tasks.length - 1) {
        setCurrentTaskIndex(currentTaskIndex + 1);
      } else {
        const correctAnswers = updatedAnswers.reduce((acc, answer) => acc + (answer ? 1 : 0), 0);
        alert(`Модуль завершено!\nПравильних відповідей: ${correctAnswers}/${tasks.length}`);
        navigate('/games');
      }
    }, 1500);
  };

  return (
    <GameContainer>
      <Question>{tasks[currentTaskIndex].question}</Question>
      <ImageContainer>
        <img src={tasks[currentTaskIndex].image} alt={`Зображення ${currentTaskIndex + 1}`} />
      </ImageContainer>
      <OptionsContainer>
        {tasks[currentTaskIndex].options.map((option, index) => (
          <OptionButton key={index} onClick={() => handleAnswerSubmit(option)}>
            {option}
          </OptionButton>
        ))}
      </OptionsContainer>
      {feedback} && <FormattedMessage
color={feedback.color}>{feedback.message}</FormattedMessage>
    </GameContainer>
  );
};

export default TangentCotangentGame;

```

```
const GameContainer = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 20px;
  text-align: center;
`;
```

```
const Question = styled.h2`
  margin-bottom: 20px;
  color: #333;
`;
```

```
const ImageContainer = styled.div`
  margin-bottom: 20px;
`;
```

```
const OptionsContainer = styled.div`
  display: flex;
  justify-content: center;
  gap: 10px;
`;
```

```
const OptionButton = styled.button`
  padding: 10px 20px;
  background-color: #f0f0f0;
  border: 2px solid #ddd;
  border-radius: 5px;
  cursor: pointer;
  &:hover {
    background-color: #e0e0e0;
  }
`;
```

```
const FeedbackMessage = styled.div`
  margin-top: 20px;
  font-size: 18px;
  color: ${({ color }) => color};
```

```
`;
```

TrigExpressionsGame.js

```
import React, { useState } from 'react';
import styled from 'styled-components';
import { useNavigate } from 'react-router-dom';
import { BlockMath, InlineMath } from 'react-katex';
import 'katex/dist/katex.min.css';

const tasks = [
  {
    question: 'Сношти вираз',
    expression: '\frac{\sin \alpha \cos \alpha}{\cos^2 \alpha}',
    options: ['\cot \alpha', '\cos \alpha', '\tan \alpha', '\sin \alpha'],
    correctAnswer: '\tan \alpha'
  },
  {
    question: 'Сношти вираз',
    expression: '\frac{1 - \cos^2 \beta}{\sin \beta}',
    options: ['\tan \beta', '\sin \beta', '\cos \beta', '\cot \beta'],
    correctAnswer: '\sin \beta'
  },
  {
    question: 'Знайди значення виразу',
    expression: '\cos^2 45^\circ + \sin^2 45^\circ',
    options: ['1', '2', '\frac{1}{2}', '0'],
    correctAnswer: '1'
  },
  {
    question: 'Сношти вираз',
    expression: '\frac{\sin^2 x}{1 - \cos^2 x}',
    options: ['1', '\tan x', '\cot x', '\sec x'],
    correctAnswer: '\tan x'
  },
  {
    question: 'Знайди значення виразу',
    expression: '\frac{1}{\sin^2 30^\circ} - \cot^2 30^\circ',
    options: ['3', '4', '2', '1'],
    correctAnswer: '3'
  }
];
```

```

    }
  ];

const TrigExpressionsGame = () => {
  const [currentTaskIndex, setCurrentTaskIndex] = useState(0);
  const [userAnswers, setUserAnswers] = useState([]);
  const [timeSpent, setTimeSpent] = useState(0);
  const [startTime] = useState(Date.now());
  const [feedback, setFeedback] = useState(null);
  const [answerDisabled, setAnswerDisabled] = useState(false); // Додаємо стан для блокування
кнопок
  const navigate = useNavigate();

  const handleAnswerSubmit = (selectedOption) => {
    if (answerDisabled) return; // Якщо кнопки заблоковані, не дозволяємо подавати відповіді

    const isCorrect = selectedOption === tasks[currentTaskIndex].correctAnswer;

    const updatedAnswers = [...userAnswers, { selectedOption, isCorrect }];
    setUserAnswers(updatedAnswers);

    setAnswerDisabled(true); // Блокуємо кнопки після вибору відповіді

    if (isCorrect) {
      setFeedback({ message: 'Правильно!', correct: true });
    } else {
      setFeedback({
        message: `Неправильно. Правильна відповідь: `,
        correct: false,
        correctAnswer: tasks[currentTaskIndex].correctAnswer
      });
    }
  }

  setTimeout(() => {
    setFeedback(null);
    setAnswerDisabled(false); // Розблокуємо кнопки для наступного завдання

    if (currentTaskIndex < tasks.length - 1) {

```

```

    setCurrentTaskIndex(currentTaskIndex + 1);
  } else {
    const endTime = Date.now();
    const totalTimeSpent = Math.floor((endTime - startTime) / 1000);
    setTimeSpent(totalTimeSpent);

    const correctAnswersCount = updatedAnswers.reduce((acc, answer) => {
      return acc + (answer.isCorrect ? 1 : 0);
    }, 0);

    // Збереження результатів у статистику
    const statistics = JSON.parse(localStorage.getItem('statistics')) || [];
    statistics.push({ module: 'TrigExpressionsGame', timeSpent: totalTimeSpent, correctAnswers:
correctAnswersCount });
    localStorage.setItem('statistics', JSON.stringify(statistics));

    // Відображення результатів на екрані
    alert(`Модуль завершено!\nЧас:   ${totalTimeSpent}   секунд\nПравильних   відповідей:
${correctAnswersCount}/${tasks.length}`);

    // Перенаправлення користувача на сторінку з модулями
    navigate('/games');
  }
}, 3000); // Затримка на 3 секунди перед переходом до наступного завдання або результатів
};

return (
  <GameContainer>
    <Question>{tasks[currentTaskIndex].question}</Question>
    <ExpressionContainer>
      <BlockMath>{tasks[currentTaskIndex].expression}</BlockMath>
    </ExpressionContainer>
    <OptionsContainer>
      {tasks[currentTaskIndex].options.map((option, index) => (
        <OptionButton
          key={index}
          onClick={() => handleAnswerSubmit(option)}
          disabled={answerDisabled} // Встановлюємо властивість disabled для кнопки

```

```

    >
    <InlineMath>{option}</InlineMath>
  </IconButton>
)}
</OptionsContainer>
{feedback && (
  <Feedback correct={feedback.correct}>
    {feedback.correct ? (
      feedback.message
    ) : (
      <>
        {feedback.message}
        <InlineMath>{feedback.correctAnswer}</InlineMath>
      </>
    )}
  </Feedback>
)}
</GameContainer>
);
};

```

```
export default TrigExpressionsGame;
```

```

const GameContainer = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 30px;
  text-align: center;
  background-color: #f9f9f9;
  border-radius: 12px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
  max-width: 800px;
  margin: 40px auto;
`;

```

```

const Question = styled.h2`
  margin-bottom: 20px;

```

```

    color: #333;
    font-weight: 600;
  `;

const ExpressionContainer = styled.div`
  background-color: #ffffff;
  padding: 20px;
  border-radius: 12px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  margin-bottom: 30px;
`;

const OptionsContainer = styled.div`
  display: flex;
  flex-wrap: wrap;
  justify-content: space-around;
  margin-bottom: 30px;
`;

const OptionButton = styled.button`
  padding: 15px 25px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 8px;
  font-size: 18px;
  cursor: pointer;
  margin: 10px;
  transition: background-color 0.3s ease;

  &:hover {
    background-color: #45a049;
  }

  &:active {
    background-color: #3e8e41;
  }

```

```

&:disabled {
  background-color: #d3d3d3;
  cursor: not-allowed;
}
`;

```

```

const Feedback = styled.div`
  margin-top: 20px;
  font-size: 18px;
  color: ${props => (props.correct ? 'green' : 'red')};
  font-weight: bold;
`;

```

TrigIdentityGame.js

```

import React, { useState } from 'react';
import styled from 'styled-components';
import { useNavigate } from 'react-router-dom';

const tasks = [
  { question: 'Обчисліть значення  $\sin^2(x) + \cos^2(x)$  для кута  $x = 30^\circ$ , correctAnswer: '1' },
  { question: 'Дано:  $\cos(x) = 0.6$ , знайдіть значення  $\sin(x)$ ', correctAnswer: '0.8' },
  { question: 'Якщо  $\sin(x) = 0.8$ , знайдіть  $\cos(x)$  для кута  $x$ ', correctAnswer: '0.6' },
  { question: 'Дано  $\sin(x) = 0.5$  і  $\cos(x) = 0.866$ . Перевірте, чи виконується основна тригонометрична тотожність.', correctAnswer: 'так' },
  { question: 'Обчисліть  $1 - \cos^2(x)$  для кута  $x = 45^\circ$ , correctAnswer: '0.5' }
];

const TrigIdentityGame = () => {
  const [currentTaskIndex, setCurrentTaskIndex] = useState(0);
  const [userAnswers, setUserAnswers] = useState([]);
  const [feedback, setFeedback] = useState("");
  const [feedbackColor, setFeedbackColor] = useState("");
  const [isAnswered, setIsAnswered] = useState(false); // Додаємо стан для контролю відповіді
  const [startTime] = useState(Date.now());
  const navigate = useNavigate();

  const handleAnswerSubmit = (userAnswer) => {
    if (isAnswered) return; // Якщо вже відповіли, не дозволяємо знову відповісти

```

```

const correctAnswer = tasks[currentTaskIndex].correctAnswer;
const isCorrect = userAnswer === correctAnswer;

setFeedbackColor(isCorrect ? 'green' : 'red');
setFeedback(isCorrect ? 'Правильно!' : `Неправильно! Правильна відповідь: ${correctAnswer}`);
setIsAnswered(true); // Встановлюємо стан, що відповідь на це питання вже дана

const updatedAnswers = [...userAnswers, isCorrect];
setUserAnswers(updatedAnswers);

setTimeout(() => {
  setFeedback("");
  setFeedbackColor("");
  setIsAnswered(false); // Дозволяємо відповідати на наступне питання

  if (currentTaskIndex < tasks.length - 1) {
    setCurrentTaskIndex(currentTaskIndex + 1);
  } else {
    const endTime = Date.now();
    const totalTimeSpent = Math.floor((endTime - startTime) / 1000);
    const correctAnswers = updatedAnswers.reduce((acc, answer) => acc + (answer ? 1 : 0), 0);

    alert(`Модуль завершено!\nЧас:   ${totalTimeSpent}   секунд\nПравильних   відповідей:
    ${correctAnswers}/${tasks.length}`);

    navigate('/games');
  }
}, 1500);
};

return (
  <GameContainer>
    <Card>
      <Question>{tasks[currentTaskIndex].question}</Question>
      <AnswerInput
        type="text"
        placeholder="Введіть вашу відповідь"
        disabled={isAnswered} // Вимикаємо поле введення після відповіді

```

```

    onKeyPress={e) => {
      if (e.key === 'Enter') {
        handleAnswerSubmit(e.target.value.trim());
        e.target.value = "";
      }
    }}
  />

  <SubmitButton
    onClick={() => {
      const input = document.querySelector('input');
      handleAnswerSubmit(input.value.trim());
      input.value = "";
    }}
    disabled={isAnswered} // Вимикаємо кнопку після відповіді
  >
    Готово
  </SubmitButton>
  {feedback && <FeedbackMessage color={feedbackColor}>{feedback}</FeedbackMessage> }
</Card>
</GameContainer>
);
};

```

```
export default TrigIdentityGame;
```

```

const GameContainer = styled.div`
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 40px;
  background-color: #f0f4f8;
  min-height: 100vh;
`;

```

```

const Card = styled.div`
  background-color: #fff;
  border-radius: 12px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

```

```
padding: 30px;
width: 100%;
max-width: 600px;
text-align: center;
`;
```

```
const Question = styled.h2`
  margin-bottom: 20px;
  color: #333;
`;
```

```
const AnswerInput = styled.input`
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ddd;
  border-radius: 5px;
  margin-bottom: 20px;
  width: 100%;
  box-sizing: border-box;
`;
```

```
const SubmitButton = styled.button`
  padding: 10px 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  margin-bottom: 20px;
  &:hover {
    background-color: #45a049;
  }
  &:disabled {
    background-color: #ddd;
    cursor: not-allowed;
  }
`;
```

```
const FeedbackMessage = styled.div`
  margin-top: 20px;
  font-size: 18px;
  color: ${(props) => props.color};
`;
```

TrigRatiosGame.js

```
import React, { useState, useEffect } from 'react';
import styled from 'styled-components';
import { useNavigate } from 'react-router-dom';

import image1 from './img/image5.png';
import image2 from './img/image6.png';
import image3 from './img/image7.png';
import image4 from './img/image8.png';
import image5 from './img/image9.png';

const tasks = [
  { image: image1, question: 'Обчисли довжину сторони AB (округли до одиниць)', correctAnswer: '7'
},
  { image: image2, question: 'Обчисли довжину сторони FK (округли до одиниць)', correctAnswer: '5'
},
  { image: image3, question: 'Обчисли довжину сторони LN (округли до одиниць)', correctAnswer: '7'
},
  { image: image4, question: 'Обчисли довжину сторони AC (округли до одиниць)', correctAnswer: '6'
},
  { image: image5, question: 'Обчисли довжину сторони FO (округли до одиниць)', correctAnswer: '3'
}
];

const TrigonometricRelationsGame = () => {
  const [currentTaskIndex, setCurrentTaskIndex] = useState(0);
  const [userAnswer, setUserAnswer] = useState("");
  const [timeSpent, setTimeSpent] = useState(0);
  const [startTime] = useState(Date.now());
  const [showResult, setShowResult] = useState(false);
  const [isCorrect, setIsCorrect] = useState(false);
  const [correctAnswers, setCorrectAnswers] = useState(0);
  const navigate = useNavigate();
```

```

useEffect(() => {
  if (showResult) {
    const timeout = setTimeout(() => {
      if (currentTaskIndex < tasks.length - 1) {
        setCurrentTaskIndex(currentTaskIndex + 1);
        setUserAnswer("");
        setShowResult(false);
      } else {
        const endTime = Date.now();
        const totalTimeSpent = Math.floor((endTime - startTime) / 1000);
        setTimeSpent(totalTimeSpent);

        const statistics = JSON.parse(localStorage.getItem('statistics')) || [];
        statistics.push({
          module: 'TrigonometricRelationsGame',
          timeSpent: totalTimeSpent,
          correctAnswers: correctAnswers,
          totalTasks: tasks.length
        });
        localStorage.setItem('statistics', JSON.stringify(statistics));

        alert(`Модуль завершено!\nЧас:   ${totalTimeSpent}   секунд\nПравильних   відповідей:
        ${correctAnswers}/${tasks.length}`);
        navigate('/games');
      }
    }, 2000);
    return () => clearTimeout(timeout);
  }
}, [showResult, currentTaskIndex, correctAnswers, navigate, startTime]);

const handleAnswerSubmit = () => {
  const correct = userAnswer === tasks[currentTaskIndex].correctAnswer;
  setIsCorrect(correct);
  setShowResult(true);

  if (correct) {
    setCorrectAnswers(prev => prev + 1);
  }
}

```

```

    }
  };

  return (
    <GameContainer>
      <Question>{tasks[currentTaskIndex].question}</Question>
      <ImageContainer>
        <img src={tasks[currentTaskIndex].image} alt={`Зображення ${currentTaskIndex + 1}`} />
      </ImageContainer>
      <AnswerInput
        type="text"
        value={userAnswer}
        placeholder="Введіть вашу відповідь"
        onChange={(e) => setUserAnswer(e.target.value.trim())}
        disabled={showResult}
      />
      <SubmitButton onClick={handleAnswerSubmit} disabled={showResult || userAnswer === ""}>
        Готово
      </SubmitButton>
      {showResult && (
        <ResultMessage correct={isCorrect}>
          {isCorrect ? 'Правильно!' : 'Неправильно. Правильна відповідь: '}
          ${tasks[currentTaskIndex].correctAnswer}`}
        </ResultMessage>
      )}
    </GameContainer>
  );
};

export default TrigonometricRelationsGame;

const GameContainer = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 20px;
  text-align: center;
`;

```

```
const Question = styled.h2`
  margin-bottom: 20px;
  color: #333;
`;
```

```
const ImageContainer = styled.div`
  margin-bottom: 20px;
`;
```

```
const AnswerInput = styled.input`
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ddd;
  border-radius: 5px;
  margin-bottom: 20px;
`;
```

```
const SubmitButton = styled.button`
  padding: 10px 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  &:hover {
    background-color: #45a049;
  }
`;
```

```
const ResultMessage = styled.div`
  margin-top: 20px;
  color: ${props => (props.correct ? 'green' : 'red')};
  font-size: 18px;
`;
```

TrigValuesGame.js

```
import React, { useState } from 'react';
import styled from 'styled-components';
```

```

import { useNavigate } from 'react-router-dom';

const tasks = [
  { question: 'Знайди гострі куту  $\alpha$  та  $\beta$ , якщо  $\cos \alpha = 0.809$ ', correctAnswer: { alpha: '36', beta: '54' }
},
  { question: 'Знайди гострі куту  $\alpha$  та  $\beta$ , якщо  $\sin \alpha = 0.242$ ', correctAnswer: { alpha: '14', beta: '76' }
},
  { question: 'Знайди гострі куту  $\alpha$  та  $\beta$ , якщо  $\cos \beta = 0.985$ ', correctAnswer: { alpha: '9', beta: '81' }
},
  { question: 'Знайди гострі куту  $\alpha$  та  $\beta$ , якщо  $\sin \beta = 0.292$ ', correctAnswer: { alpha: '17', beta: '73' }
},
  { question: 'Знайди гострі куту  $\alpha$  та  $\beta$ , якщо  $\operatorname{tg} \alpha = 0.577$ ', correctAnswer: { alpha: '30', beta: '60' }
},
];

const TrigValuesGame = () => {
  const [currentTaskIndex, setCurrentTaskIndex] = useState(0);
  const [userAnswers, setUserAnswers] = useState([]);
  const [feedback, setFeedback] = useState("");
  const [feedbackColor, setFeedbackColor] = useState("");
  const [isAnswered, setIsAnswered] = useState(false);
  const [startTime] = useState(Date.now());
  const navigate = useNavigate();

  const handleAnswerSubmit = (alphaAnswer, betaAnswer) => {
    if (isAnswered) return;

    const isAlphaCorrect = alphaAnswer === tasks[currentTaskIndex].correctAnswer.alpha;
    const isBetaCorrect = betaAnswer === tasks[currentTaskIndex].correctAnswer.beta;

    setIsAnswered(true);

    const updatedAnswers = [...userAnswers, { alphaAnswer, betaAnswer, isAlphaCorrect, isBetaCorrect }];
    setUserAnswers(updatedAnswers);

    if (isAlphaCorrect && isBetaCorrect) {
      setFeedback('Правильно!');
    }
  }
};

```

```

    setFeedbackColor('green');
  } else {
    setFeedback(`Неправильно. Правильні відповіді:  $\alpha$  =
    ${tasks[currentTaskIndex].correctAnswer.alpha},  $\beta$  = ${tasks[currentTaskIndex].correctAnswer.beta}`);
    setFeedbackColor('red');
  }

  setTimeout(() => {
    setFeedback("");
    setFeedbackColor("");
    setIsAnswered(false);

    if (currentTaskIndex < tasks.length - 1) {
      setCurrentTaskIndex(currentTaskIndex + 1);
    } else {
      const endTime = Date.now();
      const totalTimeSpent = Math.floor((endTime - startTime) / 1000);
      const correctAnswersCount = updatedAnswers.reduce((acc, answer) => {
        return acc + (answer.isAlphaCorrect && answer.isBetaCorrect ? 1 : 0);
      }, 0);

      alert(`Модуль завершено!\nЧас: ${totalTimeSpent} секунд\nПравильних відповідей:
      ${correctAnswersCount}/${tasks.length}`);

      navigate('/games');
    }
  }, 2000);
};

return (
  <GameContainer>
    <Card>
      <Question>{tasks[currentTaskIndex].question}</Question>
      <AnswerContainer>
        <Label> $\alpha$  = </Label>
        <AnswerInput
          type="text"
          placeholder="Введіть значення  $\alpha$ "

```

```

    id="alphaAnswer"
    disabled={isAnswered} // Забороняє змінювати відповідь після введення
  />
  <Label>°</Label>
  <Label>β = </Label>
  <AnswerInput
    type="text"
    placeholder="Введіть значення β"
    id="betaAnswer"
    disabled={isAnswered} // Забороняє змінювати відповідь після введення
  />
  <Label>°</Label>
</AnswerContainer>
<SubmitButton
  onClick={() => {
    const alphaAnswer = document.getElementById('alphaAnswer').value.trim();
    const betaAnswer = document.getElementById('betaAnswer').value.trim();
    handleAnswerSubmit(alphaAnswer, betaAnswer);
  }}
  disabled={isAnswered} // Забороняє повторний клік після відправки відповіді
>
  Готово
</SubmitButton>
{feedback && <Feedback color={feedbackColor}>{feedback}</Feedback> }
</Card>
</GameContainer>
);
};

```

```
export default TrigValuesGame;
```

```

const GameContainer = styled.div`
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 40px;
  background-color: #f0f4f8;
  min-height: 100vh;

```

```
`;
```

```
const Card = styled.div`
  background-color: #fff;
  border-radius: 12px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  padding: 30px;
  width: 100%;
  max-width: 600px;
  text-align: center;
`;
```

```
const Question = styled.h2`
  margin-bottom: 20px;
  color: #333;
`;
```

```
const AnswerContainer = styled.div`
  display: flex;
  align-items: center;
  justify-content: center;
  margin-bottom: 20px;
`;
```

```
const Label = styled.span`
  font-size: 24px;
  margin-right: 10px;
`;
```

```
const AnswerInput = styled.input`
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ddd;
  border-radius: 5px;
  margin-right: 10px;
  width: 60px;
`;
```

```
const SubmitButton = styled.button`
  padding: 10px 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  margin-bottom: 20px;
  &:hover {
    background-color: #45a049;
  }
  &:disabled {
    background-color: #ddd;
    cursor: not-allowed;
  }
`;
```

```
const Feedback = styled.div`
  margin-top: 20px;
  font-size: 18px;
  color: ${props => props.color};
`;
```