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

«___»____ 2023 p.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

«ЕЛЕМЕНТИ ВЕКТОРНОЇ АЛГЕБРИ: ПРОГРАМНА РЕАЛІЗАЦІЯ ДИДАКТИЧНОГО ІНСТРУМЕНТУ ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «АЛГЕБРА ТА ГЕОМЕТРІЯ»»

зі спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерні науки»

ступеня магістр

Виконавець роботи	Косогов Тарас	Леонідови	Ч	
_	(пілпис)	_ «»	2023 p.	
Науковий керівник	к.фм.н., доц.	Парфьоно	ва Тетяна Олекс	андрівна
		«»	2023 p.	
	(підпис)			
Рецензент				

ПОЛТАВА 2023

3MICT

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ СИМВОЛІВ ОЛИНИЦЬ
СКОРОЧЕНЬ. ТЕРМІНІВ
ВСТУП 4
1. ПОСТАНОВКА ЗАДАЧІ 0
2. ІНФОРМАЦІИНИИ ОГЛЯД
2.1. Необхідність та актуальність теми роботи
2.2. Огляд джерел для розробки тренажера9
2.3. Використання програмного забезпечення для вирішення
математичних завдань та задач11
2.4. Огляд розроблених тренажерів дистанційного курсу «Алгебра та
геометрія»14
3. ТЕОРЕТИЧНА ЧАСТИНА
3.1. Огляд матеріалу з теми17
3.2. Алгоритмізація задачі за темою роботи 20
3.3. Опис засобів програмного забезпечення
4. ПРАКТИЧНА ЧАСТИНА
4.1. Блок-схема алгоритму роботи тренажера
4.2. Опис програмної реалізації
4.3. Тестування розробленого тренажера54
4.4. Необхідна користувачу інструкція 59
ВИСНОВКИ
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 64
ДОДАТОК А. КОД ПРОГРАМИ Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,

Умовні позначення, символи,	Пояснення умовних позначень,
скорочення, терміни	скорочень, символів
вектор	направлений відрізок, що має певну
	довжину та певний напрямок
React	відкрита JavaScript бібліотека для
	створення інтерфейсів користувача
стан	об'єкт в React, який використовується
	для зберігання даних, які можуть
	змінюватися протягом життєвого
	циклу компонента
функція	функціональний компонент в React
	для роботи зі станом, ефектами та
	контекстом
\vec{a}, \vec{AB}	позначення вектора
,	

СКОРОЧЕНЬ, ТЕРМІНІВ

ВСТУП

Програми-тренажери є ефективними інструментами для навчання студентів, оскільки вони дозволяють студентам отримати практичний досвід в зручний для них час і місце. Вони також можуть допомогти студентам засвоїти складну теорію швидше і легше, використовуючи тести та практичні завдання.

Крім того, тренажери можуть бути корисними для викладачів, які можуть використовувати їх для оцінювання студентів та аналізу їхнього прогресу.

Отже, розробка тренажерів для дистанційних курсів є дуже актуальною і важливою задачею для підтримки ефективного дистанційного навчання.

Метою роботи є розробка алгоритму та програмного забезпечення тренажера з теми «Елементи векторної алгебри» та його впровадження в навчальний процес для студентів дистанційного навчального курсу «Алгебра та геометрія».

Об'єктом розробки є навчання студентів векторної алгебри.

Предмет розробки – саме програмне забезпечення, яке повинно забезпечувати ефективне навчання студентів.

Головне завдання – створення програмного тренажера, який надає студентам можливість вивчати і закріплювати знання з елементів векторної алгебри в рамках дистанційного курсу «Алгебра та геометрія».

Перелік використаних методів – JavaScript бібліотека React, редактор вихідного коду Visual Studio Code, пакет react-latex.

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі. У другому розділі описано необхідність та актуальність теми роботи, огляд джерел для розробки тренажера, використання програмного забезпечення для вирішення математичних завдань та задач, огляд розроблених тренажерів дистанційного курсу «Алгебра та геометрія». У третьому розділі описано огляд матеріалу з теми, алгоритмізацію задачі за темою роботи, засоби програмного забезпечення. У четвертому розділі вказано опис блок-схеми алгоритму роботи тренажера, програмної реалізації, тестування розробленого тренажера, необхідної користувачу інструкції.

Обсяг пояснювальної записки: 76 стор., в т.ч. основна частина - 65 стор., джерела - 9 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Проблема полягає в тому, що студенти можуть мати складнощі зі засвоєнням теорії та практичного використання елементів векторної алгебри в рамках курсу «Алгебра та геометрія». Це може бути пов'язано з тим, що матеріал є складним та абстрактним для студентів, а також з тим, що дистанційне навчання не забезпечує достатнього рівня взаємодії студентів з викладачем та іншими студентами.

Для розв'язання цієї проблеми потрібно розробити тренажер з теми «Елементи векторної алгебри», який буде доступний студентам для самостійного навчання та практики. Тренажер повинен містити вправи, що допоможуть студентам зрозуміти та запам'ятати теорію. Програмне забезпечення для тренажера повинно містити наступні елементи:

- Навчальний матеріал: текстові пояснення теорії, графіки та ілюстрації.
- Тести та вправи: тренажер повинен містити практичні завдання, які дозволяють студентам перевірити свої знання та здібності у векторній алгебрі.
- Автоматизовану систему оцінювання: програма має забезпечити автоматичну перевірку відповідей та надавати студентам звіти про їхні досягнення.

Розробка такого тренажера є важливим кроком для поліпшення ефективності дистанційного навчання та підвищення рівня розуміння теорії елементів векторної алгебри.

Ключові завдання програмної реалізації такого тренажера можуть включати:

 Розробка навчального вмісту: Розробка навчальних матеріалів та вправ з векторної алгебри, які будуть доступні для студентів на платформі курсу.

- Інтерактивні вправи: Створення інтерактивних завдань та вправ, де студенти можуть виконувати операції з векторами та перевіряти свої результати.
- Зворотний зв'язок: Надання зворотного зв'язку студентам щодо їх відповідей, пояснення помилок.
- Відстеження прогресу: Можливість відстежувати прогрес студентів, включаючи їхні оцінки.
- 5. Доступність та зручність: Забезпечення доступності для різних категорій користувачів та забезпечення зручності навчання через дистанційні технології.

Для складання тестів з теми «Елементи векторної алгебри» пропонуються такі питання [1]:

- 1. Що називається вектором?
- 2. Які вектори називаються колінеарними; компланарними?
- 3. Які операції над векторами називаються лінійними?
- 4. Що називається розкладом вектора за базисом?
- 5. Який базис називається ортонормованим?
- 6. Що називається декартовою прямокутною системою координат у просторі?
- 7. Що називається радіусом-вектором точки *M* відносно декартової прямокутної системи координат?

Вони використовуються для тестування студентів з перевірки основних понять.

Практичне засвоєння матеріалу відбувається за допомогою прикладів. Тому практичні навики здійснюються завдяки завданням:

Завдання 1. Задано дві точки A(-3,6,-5) та B(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} .

Завдання 2. Чи колінеарні вектори $\vec{c_1} = -\vec{a} + 4\vec{b}$ і $\vec{c_2} = 3\vec{a} - 2\vec{b}$, побудовані на векторах $\vec{a} = \{2, -2, 3\}$ і $\vec{b} = \{3, 1, -1\}$.[2]

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Необхідність та актуальність теми роботи

Необхідність і актуальність теми «Елементи векторної алгебри: програмна реалізація тренажера дистанційного курсу «Алгебра та геометрія»» можна розглядати в більш детальному контексті:

1. Ростучий попит на математичну освіту;

Зростання глобальної конкуренції та потреба в технологічно обізнаних спеціалістах робить математичну освіту більш актуальною і необхідною, включаючи в себе векторну алгебру.

2. Математика в цифровому світі;

Векторна алгебра використовується в численних аспектах цифрової технології, таких як комп'ютерна графіка, машинне навчання, інтернетрозробка тощо. Зрозуміння цієї математики стає ключовим фактором для розвитку кар'єри в інформаційних та технологічних галузях.

3. Віддалене навчання;

Пандемія COVID-19 та війна підкреслила важливість дистанційного навчання. Програмний тренажер може стати інструментом для надання освіти в умовах, коли доступ до учбових закладів обмежений.

4. Різнорівневе навчання;

Студенти мають різний рівень підготовки та розуміння математичних понять. Програмна реалізація може адаптувати відповідні завдання та пояснення для різних груп студентів.

5. Залучення студентів до навчання;

Інтерактивні програми та тренажери можуть зробити навчання більш цікавим і залучити студентів до активної участі та розв'язання математичних завдань. 6. Оновлення та контроль навчального процесу;

Програмна реалізація дозволяє постійно оновлювати вміст та вносити зміни у відповідності до нових методик навчання та вимог ринку праці.

7. Застосування в освіті.

Такий тренажер може бути корисним для школярів, студентів, а також для педагогів і вчителів як інструмент для навчання та викладання векторної алгебри.

Усі ці аспекти свідчать про важливість створення програмного тренажера для вивчення елементів векторної алгебри і актуальність цієї теми в сучасному освітньому та технологічному контексті. Такий тренажер сприятиме покращенню рівня математичної грамотності та підготовки фахівців для сучасного світу.

2.2. Огляд джерел для розробки тренажера

Огляд інших джерел для розробки тренажера є важливим етапом, оскільки дозволяє отримати ширший огляд інформації та використати найкращі практики та ідеї з уже наявних рішень. Це може допомогти покращити якість та ефективність створюваного тренажера, а також зекономити час та зусилля на пошук необхідної інформації та розробку аналогів. Огляд інших джерел дозволяє також оцінити поточний рівень розвитку технологій та тенденції їх подальшого розвитку в даній галузі, що допомагає вибрати оптимальні методи розробки та інструменти для досягнення поставлених цілей.

Стаття «Програмна реалізація розв'язування задач за темою «Пряма на площині» дисципліни «Алгебра і геометрія» присвячена розробці тренажера для навчання розв'язуванню задач за темою «Пряма на площині» [3]. Автори статті наводять детальний опис функцій тренажера, в якій використовуються елементи графічного інтерфейсу для візуалізації математичних понять та алгоритмів розв'язування задач. У тренажері передбачені такі функції:

- пошук координат вектора;
- пошук координат точки;
- визначення довжини сторони трикутника;
- пошук рівняння прямої;
- визначення кута та рівняння його бісектриси;
- визначення відстані між точкою і прямою;
- знаходження площі трикутника.

Для розв'язування задач в програмі-тренажері використовується метод декомпозиції задачі на підзадачі та їх послідовне розв'язування. Автори статті демонструють процес розв'язування однієї із задач за допомогою тренажера.

Тренажер може бути ефективним інструментом для навчання розв'язування задач за темою «Пряма на площині».

Загалом, стаття «Програмна реалізація розв'язування задач за темою «Пряма на площині» дисципліни «Алгебра і геометрія» є цікавим джерелом для викладачів та розробників програм-тренажерів, які цікавляться розвитком математичної освіти.

Стаття «Програмна реалізація розв'язування задач за темою «Вектори, площини, прямі у просторі» з дисципліни «Алгебра і геометрія» присвячена розробці програмного продукту для навчання студентів векторній алгебрі та геометрії на прикладі розв'язування задач на тему «Вектори, площини, прямі у просторі» [4].

У статті автори описують програмний продукт, за допомогою якого можна розв'язувати задачі в предметі «Алгебра і геометрія». Програма базується на платформі Delphi та містить в собі модульну структуру, яка дозволяє розв'язувати задачі різної складності. Крім того, програма має

інтерфейс користувача, який дозволяє студентам легко та зручно користуватись продуктом.

У статті автори також звертають увагу на необхідність розвитку інноваційних методів навчання, які мають допомогти студентам краще засвоювати матеріал. Розроблений продукт є одним з таких методів і дозволяє студентам самостійно вчитися та розвивати свої знання з векторної алгебри та геометрії.

Отже, стаття пропонує новий підхід до навчання векторної алгебри та геометрії, а саме застосування програмного забезпечення для розв'язування задач. Цей підхід може бути корисним як для студентів, які вивчають матеріал на дистанційному курсі, так і для викладачів, які шукають інноваційні методи навчання своїх студентів.

2.3. Використання програмного забезпечення для вирішення математичних завдань та задач

Основною причиною використання Mathway i GeoGebra є можливість швидкого та ефективного вирішення математичних завдань та задач.

GeoGebra - це безкоштовне програмне забезпечення, яке поєднує можливості геометричної та алгебраїчної роботи з математичними обчисленнями та статистикою (рис. 2.1). Воно призначене для використання в освітньому процесі як засіб демонстрації та дослідження математичних концепцій, а також для побудови графіків та візуалізації даних.

GeoGebra дозволяє створювати графіки функцій, побудовувати геометричні фігури, проводити обчислення з числовими даними, а також виконувати складні математичні операції, такі як інтегрування та диференціювання. Всі ці можливості дозволяють викладачам та студентам легко досліджувати математичні концепції та створювати демонстраційні матеріали для уроків.



Рисунок 2.1 – Сайт GeoGebra

GeoGebra є дуже популярним серед викладачів та студентів, оскільки воно дозволяє легко та швидко створювати математичні демонстрації та відтворювати різні математичні ситуації. Крім того, GeoGebra має велику спільноту користувачів та розширень, які дозволяють розширювати можливості програмного забезпечення.

GeoGebra також має можливість підключення до веб-серверів, що дозволяє створювати віддалені тестування та практичні завдання для студентів, які можуть виконувати ці завдання з будь-якого місця, де є доступ до Інтернету. Це дозволяє викладачам дистанційно контролювати знання студентів та оцінювати їх роботу.

Mathway - це онлайн-сервіс, який дозволяє вирішувати математичні задачі різного рівня складності (рис. 2.2). Сервіс можна використовувати як на комп'ютері, так і на мобільному пристрої, завдяки мобільному додатку.

	bra				🛚 Mat	hway					• :
Но	w can I help	you?									
Enter	a prob	lem									6
	f((x)			1	y			z	2	
()	Ι	[]	\checkmark	₹	≥	[##]	#	f(x)	e
x	7	8	9	-			≤	ln	B-10^m	{	i
y	4	5	6	/	^	×	>	log	log	Π	U
z	1	2	3	-	+	÷	<	-	(11.10)	π	~
	_	_	_	_							

Рисунок 2.2 – Сайт Mathway

Маthway пропонує розв'язувати задачі з різних галузей математики, таких як алгебра, геометрія, тригонометрія, статистика, математичний аналіз та інші. Крім того, сервіс може вирішувати як стандартні, так і складні задачі, які можуть включати числові значення, символьні вирази, системи рівнянь та нерівностей.

Mathway має зручний інтерфейс та просту навігацію, що дозволяє користувачам швидко і легко вирішувати свої задачі. Також, сервіс пропонує детальні кроки розв'язування, що дозволяє користувачам зрозуміти процес розв'язування та знайти помилки.

Маthway доступний у платній та безкоштовній версіях. Безкоштовна версія дозволяє користувачам вирішувати простіші задачі та переглядати детальні кроки розв'язування. Платна версія Mathway пропонує більш широкий спектр функцій та можливостей, таких як розв'язання складніших задач, використання графіків та таблиць, доступ до більшої кількості тематичних розділів та інше.

2.4. Огляд розроблених тренажерів дистанційного курсу «Алгебра та геометрія»

Серед матеріалів дистанційного курсу «Алгебра та геометрія» розглянуто тренажер з теми «Матриці і визначники» (рис. 2.3).

🗊 Тренажер: визначники і матриці	
Задачі для ро	эзв`язування:
Визначники	Матриці
1. Задано матрицю 2-ого порядку. Знайти визначник	1. Множенння матриць
Переи́ти до прикладу	Переи́ти до прикладу
2. Задано матрицю 3-ого порядку. Знаи́ти визначник за правилом трикутника	2. Додавання матриць
Перейти до прикладу	Перейти до прикладу
 Задано матрицю 3-ого порядку. Знайти визначник розкладанням по рядку 	3. Множення матриці на число
Перейти до прикладу	Перейти до прикладу
4. Задано матрицю 4-ого порядку. Розкласти визначник 4-го порядку по стовпцю	 Знаходження оберненої матриці методом приєднаної матриці
Переи́ти до прикладу	Перейти до прикладу
Полтавський Автор-розроб Керівник: к.ф. Тема: Матриці	університет економікиі торгівлі Зник: Алієв Ф.П. -м.н., доц. Парфоньова Т.В. і і визначники, (с) 2014

Рисунок 2.3 – Тренажер «Матриці і визначники»

До операцій з визначниками відносяться:

- знаходження визначника;
- знаходження визначника за правилом трикутника;
- знаходження визначника розкладанням по рядку;
- розкладання визначника 4-го порядку по стовпцю.

До операцій з матрицями відносяться:

- множення матриць;
- додавання матриць;
- множення матриці на число;
- знаходження оберненої матриці методом приєднаної матриці.

Після переходу до прикладу відображається умова і питання, на яке потрібно дати відповідь (рис. 2.4).

🗊 Знаходження визначника другого порядку		
Задано матрицю А: $\mathbf{A} = \begin{pmatrix} \mathbf{a}_{11} \\ \mathbf{a}_{21} \end{pmatrix}$		
Обчислити визначник.		
1. Яка вимірність матриці А?		
Вимірність матриці A mxn =	×	

Рисунок 2.4 – Приклад з обчислення визначника

Якщо ввести неправильне значення відразу з'явиться повідомлення про помилку (рис. 2.5).

Тренажер: визначники і 🔀	
Введене значення неправильне. Матриця А має m=2 рядки.	
ОК	

Рисунок 2.5 – Повідомлення про помилку

Покроково виводяться інші питання по даному прикладу. В кінці вже необхідно ввести значення визначника (рис. 2.6).

🕼 Знаходження визначника другого порядку	
Задано матрицю А: $\mathbf{A} = \begin{pmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & \mathbf{a}_{22} \end{pmatrix} = \begin{pmatrix} 10 & 6 \\ 2 & 5 \end{pmatrix}$	
Обчислити визначник.	
1. Яка вимірність матриці А?	
Вимірність матриці А mxn = 2 × 2	
2. Чи можна знайти визначник матриці А? Г ⊤ак Г Ні	
3. Визначник якого порядку необхідно знаи́ти? Визначник матриці А 🛛 🔁 -го порядку	
4. Яка формула використовується для обчислення визн	ачника 2-го порядку?
$\Box a) \det \mathbf{A} = \mathbf{a}_{12}\mathbf{a}_{21} + \mathbf{a}_{11}\mathbf{a}_{22} \Box B) \det \mathbf{A} = \mathbf{a}_{11}\mathbf{a}_{22} + \mathbf{a}_{12}\mathbf{a}_{21}$	
$rac{1}{5}$ det $\mathbf{A} = \mathbf{a}_{11}\mathbf{a}_{22} - \mathbf{a}_{12}\mathbf{a}_{21}$ $rac{1}{5}$ det $\mathbf{A} = \mathbf{a}_{12}\mathbf{a}_{21} - \mathbf{a}_{11}\mathbf{a}_{22}$	
5. Ввести значення визначника матриці А	Далі
$\det \mathbf{A} = \begin{vmatrix} \mathbf{z} & \mathbf{z} \\ \mathbf{z} & \mathbf{z} \end{vmatrix} = \begin{vmatrix} \mathbf{z} \\ \mathbf{z} \end{vmatrix}$	

Рисунок 2.6 – Завершення виконання прикладу з обчислення визначника

Для операцій з матрицями виконуються ан	налогічні дії (рис. 2	2.7).
---	-----------------------	-------

🗊 Множення матриць				
Задано матриці А і В:				
$\mathbf{A} = \begin{pmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} & \mathbf{a}_{13} \\ \mathbf{a}_{21} & \mathbf{a}_{22} & \mathbf{a}_{23} \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 \\ 3 & 0 & -5 \end{pmatrix}$	$\mathbf{B} = \begin{pmatrix} \mathbf{b}_{11} \\ \mathbf{b}_{21} \\ \mathbf{b}_{31} \end{pmatrix}$	b ₁₂ b ₂₂ b ₃₂	$ \begin{vmatrix} \mathbf{b}_{13} \\ \mathbf{b}_{23} \\ \mathbf{b}_{33} \end{vmatrix} = \begin{pmatrix} -1 \\ 5 \\ 0 \end{pmatrix} $	$\begin{pmatrix} 1 & 4 \\ 2 & 8 \\ 3 & -1 \end{pmatrix}$
<u>Знаи́ти А*В.</u>		52	<i>557</i> (,
1. Яка вимірність матриць А і В?				
Для матриці А mxn = 🛛 🛛 🛛				
Для матриці В mxn = 🧃 x 🕄				
2. Чи можна помножити матрицю А н	а матрицк	5 B?		
🛙 Так 🗆 Ні				
3. Яка вимірність матриці C = AB?				
Для матриці С mxn = 🛛 🛛 🛛				
4. Знаи́ти елементи матриці C= AB:				
$\mathbf{C} = \mathbf{A}\mathbf{B} = \begin{pmatrix} \mathbf{c}_{11} & \mathbf{c}_{12} & \mathbf{c}_{13} \\ \mathbf{c}_{21} & \mathbf{c}_{22} & \mathbf{c}_{23} \end{pmatrix} =$				

Рисунок 2.7 – Виконання прикладу з множення матриць

Таким чином, тренажер з теми «Матриці і визначники» дистанційного курсу «Алгебра та геометрія» пропонує допомогу у засвоєнні матеріалу з основних операцій по визначникам і матрицям.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Огляд матеріалу з теми

Вектором (геометричним вектором) називається направлений відрізок, що має певну довжину та певний напрямок.

Вектор AB – це напрямлений відрізок із початком у точці A і кінцем у точці B. Вектори позначаються як двома великими літерами, так і однією малою зі стрілкою, наприклад, \vec{a} , \overrightarrow{AB} .

Довжина відрізка *AB* називається *довжиною* (модулем, нормою) вектора \vec{a} і позначається символом $|\vec{a}| = |\overrightarrow{AB}|$.

Вектори називаються *колінеарними*, якщо вони лежать на одній прямій або на паралельних прямих. Позначення: $\vec{a} / / \vec{b}$.

Вектори називаються *компланарними*, якщо вони лежать на одній площині або на паралельних площинах.

Під *лінійними операціями* над векторами розуміють операцію множення вектора на число та додавання векторів.

Добутком вектора \vec{a} на дійсне число λ , називається вектор, що позначається $\lambda \vec{a}$, такий, що

- 1. $|\lambda \vec{a}| = |\lambda| |\vec{a}|;$
- 2. вектори \vec{a} та $\lambda \vec{a}$ однаково напрямлені, якщо $\lambda > 0$ та протилежно напрямлені, якщо $\lambda < 0$, тобто, $\vec{a} \uparrow \uparrow \lambda \vec{a}$, якщо $\lambda > 0$; $\vec{a} \uparrow \downarrow \lambda \vec{a}$, якщо $\lambda < 0$ [1][5].

Властивості операції множення вектора на число:

- 1. $(\lambda + \beta)\vec{a} = \lambda\vec{a} + \beta\vec{a}$ (дистрибутивна властивість);
- 2. $\lambda(\beta \vec{a}) = (\lambda \beta) \vec{a}$ (асоціативна властивість) [6].

Сума векторів. Нехай $\vec{a} = \overrightarrow{AB}$, $\vec{b} = \overrightarrow{BC}$, тоді вектор $\vec{c} = \overrightarrow{AC}$ називається сумою векторів \vec{a} та \vec{b} і позначається $\vec{a} = \overrightarrow{a+b}$ [1][5].

Властивості операції додавання векторів

- 1. $\vec{a} + \vec{b} = \vec{b} + \vec{a}$ (комутативна властивість);
- 2. $(\vec{a} + \vec{b}) + \vec{c} = \vec{a} + (\vec{b} + \vec{c})$ (асоціативна властивість);
- 3. $\vec{a} + \vec{0} = \vec{a}$ [6].

Упорядкована сукупність *n* лінійно незалежних векторів *n* - вимірного простору утворює його базис.

Упорядкована трійка не компланарних векторів $\vec{e_1}, \vec{e_2}, \vec{e_3}$ називається базисом у трьохвимірному просторі (у R^3).

Аналогічно, упорядкована пара не колінеарних векторів $\vec{e_1}, \vec{e_2}$ називається *базисом на площині (у R²)*. Базисом на прямій (у *R¹*) називається будь-який ненульовий вектор.

Якщо вектор *а* поданий у вигляді лінійної комбінації базисних векторів:

$$\vec{a} = x_1 \vec{e_1} + x_2 \vec{e_2} + x_3 \vec{e_3},$$

то кажуть, що він розкладений за базисом $\beta = (\vec{e_1}, \vec{e_2}, \vec{e_3})$. Тут $x_1\vec{e_1}, x_2\vec{e_2}, x_3\vec{e_3} - cкладові$ (компоненти) вектора \vec{a} ; $x_1, x_2, x_3 - числа$, що називаються координатами вектора \vec{a} у базисі β . Коротко:

$$\vec{a}=(x_1,x_2,x_3).$$

Базис β називається *прямокутним* або *ортонормованим*, якщо в $\vec{e_1}, \vec{e_2}, \vec{e_3}$ попарно перпендикулярні та мають одиничну довжину. У цьому випадку прийнято позначення:

$$\vec{e}_1 = \vec{i}, \vec{e}_2 = \vec{j}, \vec{e}_3 = \vec{k}$$

Якщо x, y, z – координати вектора \vec{a} у прямокутному базисі, то вони співпадають з проекціями вектора \vec{a} на базисні орти $\vec{i}, \vec{j}, \vec{k}$ відповідно: $x = np_{\vec{i}}\vec{a}, x = np_{\vec{i}}\vec{a}, y = np_{\vec{j}}\vec{a}, z = np_{\vec{k}}\vec{a}.$ Довжина вектора \vec{a} визначається так:

$$\left|\vec{a}\right| = \sqrt{x^2 + y^2 + z^2}$$

Кажуть, що у тривимірному просторі введена *декартова прямокутна система координат*, якщо задано:

- 1. точка О початок координат;
- 2. прямокутний базис $\beta = (\vec{i}, \vec{j}, \vec{k})$ у множині усіх геометричних векторів. Позначення: $(O, \vec{i}, \vec{j}, \vec{k})$.

Осі O_x , O_y і O_z , що проведені через точку O у напрямку базисних ортів $\vec{i}, \vec{j}, \vec{k}$ називаються координатними осями системи координат.

Вектор \overrightarrow{OM} називають *радіусом-вектором* точки M(x, y, z) і позначають $\vec{r}(M)$ або просто \vec{r} . Оскільки координати вектора \vec{r} співпадають з координатами точки M, то розклад \vec{r} за ортами має вигляд [1]:

$$\vec{r} = x\vec{i} + y\vec{j} + z\vec{k} \; .$$

Завдання 1. Задано дві точки A(-3,6,-5) та B(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} [7].

Розв'язок:
$$AB = (6 - (-3); -6 - 6; 3 - (-5)) = (9; -12; 8).$$

 $\left|\vec{a}\right| = \sqrt{9^2 + (-12)^2 + 8^2} = \sqrt{81 + 144 + 64} = \sqrt{289} = 17.$
 $\cos \alpha = \frac{9}{17}, \ \cos \beta = -\frac{12}{17}, \ \cos \gamma = \frac{8}{17}.$
 $\vec{a}_0 = \left(\frac{9}{17}, -\frac{12}{17}, \frac{8}{17}\right).$

Завдання 2. Чи колінеарні вектори $\vec{c_1} = -\vec{a} + 4\vec{b}$ і $\vec{c_2} = 3\vec{a} - 2\vec{b}$, побудовані на векторах $\vec{a} = \{2, -2, 3\}$ і $\vec{b} = \{3, 1, -1\}$ [7].

Розв'язок: $\vec{c_1} = \{(-2) + 4 \cdot 3; 2 + 4 \cdot 1; (-3) + 4 \cdot (-1)\} = \{-2 + 12; 2 + 4; -3 - 4\} = =\{10; 6; -7\}.$

$$\overrightarrow{c_2} = \{3 \cdot 2 - 2 \cdot 3; \ 3 \cdot (-2) - 2 \cdot 1; \ 3 \cdot 3 - 2 \cdot (-1)\} = \{6 - 6; -6 - 2; 9 + 2\} = \{0; -8; \}$$

11}.

Вектори
$$\vec{c_1}$$
 і $\vec{c_2}$ колінеарні, якщо $\vec{c_2} = n \cdot \vec{c_1}$.
 $n = \frac{0}{10} = 0, \ 0 \cdot \vec{c_1} = \{0 \cdot 10; \ 0 \cdot 6; \ 0 \cdot (-7)\} = \{0; 0; 0\}.$

Так як $0 \cdot \vec{c_1} \neq \vec{c_2}$, то вектори не колінеарні.

3.2. Алгоритмізація задачі за темою роботи

Алгоритм роботи тренажера з теми "Елементи векторної алгебри" для дистанційного курсу передбачає:

Етап 1: Виведення теми тренажера. Вибір перегляду теоретичного матеріалу, перехід до практичної частини.

Етап 2: Виведення практичної частини. Вибір проходження тестування чи виконання завдань.

Етап 3: Якщо вибрано тестування – проходження тестів з теми. Якщо вибрано виконання завдань – розв'язання запропонованих завдань. Перехід до результатів.

Етап 4: Виведення оцінки за пройдений матеріал. Пропозиція продовжити виконання тренажера чи вихід та закриття тренажера. При продовженні перехід до етапу 2.

Оскільки тестування і розв'язання завдань виконується окремо один від одного, то і алгоритми для них будуть окремі. Так, для тестування буде наступний алгоритм роботи:

Крок 1. Відображення питання «Що називається вектором?» і варіантів відповіді:

 направлений відрізок, що має певну довжину та певний напрямок;

- фізична величина, яка має тільки один параметр модуль, а напрям вектора не має значення;
- графічне зображення фігури на площині або в просторі;
- матриця з одним стовпцем або рядком, яка використовується для розв'язання систем лінійних рівнянь.

Якщо вибрано перший варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 2. Відображення питання «Вектор – це напрямлений відрізок із початком у точці *A* і кінцем у точці *B*. Як позначаються вектори?» і варіантів відповіді:

- \overrightarrow{AB} ;
- $\left| \overrightarrow{AB} \right|;$
- *ā*;
- $|\vec{a}|$.

Якщо вибрано перший і третій варіанти, то позначається, що ці варіанти були правильними. Зараховується 1 бал і відбувається перехід на наступний крок. При виборі лише одного з них відповідно лише він позначається і зараховується 0,5 бала. В іншому разі вибране відзначається неправильним.

Крок 3. Відображення питання «Які вектори називаються колінеарними?» і варіантів відповіді:

- Колінеарними векторами називаються вектори, які знаходяться в різних площинах і не перетинаються;
- Колінеарними векторами називаються вектори, які мають різні напрямки, але однакові модулі;
- Вектори називаються колінеарними, якщо вони лежать на одній прямій або на паралельних прямих;

• Вектори називаються колінеарними, якщо вони лежать на одній площині або на паралельних площинах.

Якщо вибрано третій варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 4. Відображення питання «Які вектори називаються компланарними?» і варіантів відповіді:

- Компланарними векторами називаються вектори, які знаходяться в різних площинах і не перетинаються;
- Компланарними векторами називаються вектори, які мають різні напрямки, але однакові модулі;
- Вектори називаються компланарними, якщо вони лежать на одній прямій або на паралельних прямих;
- Вектори називаються компланарними, якщо вони лежать на одній площині або на паралельних площинах.

Якщо вибрано четвертий варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 5. Відображення питання «Які операції над векторами називаються лінійними?» і варіантів відповіді:

- множення вектора на число;
- додавання векторів;
- перехід вектора до нового базису;
- скалярний добуток векторів.

Якщо вибрано перший і другий варіанти, то позначається, що ці варіанти були правильними. Зараховується 1 бал і відбувається перехід на наступний крок. При виборі лише одного з них відповідно лише він позначається і зараховується 0,5 бала. В іншому разі вибране відзначається неправильним. Крок 6. Відображення питання «Як називається упорядкована трійка не компланарних векторів $\vec{e_1}, \vec{e_2}, \vec{e_3}$?» і варіантів відповіді:

- базисом на прямій (у *R*¹);
- базисом на площині (у *R*²);
- базисом у трьохвимірному просторі (у R^3).

Якщо вибрано третій варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 7. Відображення питання «Як називається упорядкована пара не колінеарних векторів $\vec{e_1}, \vec{e_2}$?» і варіантів відповіді:

- базисом на прямій (у *R*¹);
- базисом на площині (у *R*²);
- базисом у трьохвимірному просторі (у R^3).

Якщо вибрано другий варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 8. Відображення питання «Як називається будь-який ненульовий вектор?» і варіантів відповіді:

- базисом на прямій (у *R*¹);
- базисом на площині (у *R*²);
- базисом у трьохвимірному просторі (у R^3).

Якщо вибрано перший варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 9. Відображення питання «Якщо вектор *а* поданий у вигляді лінійної комбінації базисних векторів:

$$\vec{a} = x_1 \vec{e_1} + x_2 \vec{e_2} + x_3 \vec{e_3},$$

то кажуть, що він розкладений за базисом. За яким базисом він розкладений?» і варіантів відповіді:

- $\vec{a} = (x_1, x_2, x_3);$
- $\beta = \left(\vec{e_1}, \vec{e_2}, \vec{e_3}\right)$
- $\beta = (x_1, x_2, x_3);$
- $\beta = (x_1 \overrightarrow{e_1}, x_2 \overrightarrow{e_2}, x_3 \overrightarrow{e_3}).$

Якщо вибрано другий варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 10. Відображення питання «Який базис називається ортонормованим?» і варіантів відповіді:

- Базис β називається прямокутним або ортонормованим, якщо в $\vec{e_1}, \vec{e_2}, \vec{e_3}$ попарно перпендикулярні та мають одиничну довжину;
- Базис β називається прямокутним або ортонормованим, якщо в $\vec{e_1}, \vec{e_2}, \vec{e_3}$ попарно паралельні та мають одиничну довжину;
- Базис β називається прямокутним або ортонормованим, якщо в $\vec{e_1}, \vec{e_2}, \vec{e_3}$ попарно перпендикулярні та мають різну довжину;
- Базис β називається прямокутним або ортонормованим, якщо в $\vec{e_1}, \vec{e_2}, \vec{e_3}$ попарно паралельні та мають різну довжину.

Якщо вибрано перший варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 11. Відображення питання «Що називається декартовою прямокутною системою координат у просторі?» і варіантів відповіді:

• У тривимірному просторі введена декартова прямокутна система координат, якщо задана точка *О* (початок координат);

- У тривимірному просторі введена декартова прямокутна система координат, якщо задано прямокутний базис β = (i, j, k) у множині усіх геометричних векторів;
- У тривимірному просторі введена декартова прямокутна система координат, якщо задано точку O (початок координат) і прямокутний базис β = (i, j, k) у множині усіх геометричних векторів.

Якщо вибрано третій варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 12. Відображення питання «Що називається радіусом-вектором точки M(x, y, z) відносно декартової прямокутної системи координат?» і варіантів відповіді:

- точка M(x, y, z);
- вектор \overrightarrow{MO} ;
- вектор \overrightarrow{OM} .

Якщо вибрано третій варіант, то позначається, що цей варіант був правильним. Зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі вибране відзначається неправильним.

Крок 13. Підрахунок кількості набраних балів. Виведення отриманої оцінки.

Алгоритм роботи для розв'язання завдань:

Крок 1. Відображення умови «Задано дві точки A(-3,6,-5) та B(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} » і завдання: Знайти координати вектора \overrightarrow{AB} .

 $\overrightarrow{AB} = (\underline{\qquad}; \underline{\qquad}).$

Якщо відповідь (9; -12; 8), то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про

помилку «Відповідь неправильна! $\overrightarrow{AB} = (6 - (-3); -6 - 6; 3 - (-5)) = (9; -12; 8)».$

Крок 2. Відображення умови «Задано дві точки A(-3,6,-5) та B(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} » і завдання: Знайти довжину вектора \overrightarrow{AB} .

$$\left| \vec{a} \right| =$$
_____.

Якщо відповідь 17, то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про помилку «Відповідь неправильна!

$$\left| \vec{a} \right| = \sqrt{9^2 + (-12)^2 + 8^2} = \sqrt{81 + 144 + 64} = \sqrt{289} = 17$$
».

Крок 3. Відображення умови «Задано дві точки A(-3,6,-5) та B(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} » і завдання: Знайти напрямні косинуси вектора \overrightarrow{AB} .

 $\cos \alpha =$ ____.

Якщо відповідь 9/17, то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про помилку «Відповідь неправильна! $\cos \alpha = \frac{9}{17}$ ».

Крок 4. Відображення умови «Задано дві точки A(-3,6,-5) та B(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} » і завдання: Знайти напрямні косинуси вектора \overrightarrow{AB} .

 $\cos\beta =$ _____.

Якщо відповідь -12/17, то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про помилку «Відповідь неправильна! $\cos \beta = -\frac{12}{17}$ ».

Крок 5. Відображення умови «Задано дві точки A(-3,6,-5) та B(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} » і завдання: Знайти напрямні косинуси вектора \overrightarrow{AB} .

 $\cos \gamma =$ _____.

Якщо відповідь 8/17, то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про помилку «Відповідь неправильна! $\cos \gamma = \frac{8}{17}$ ».

Крок 6. Відображення умови «Задано дві точки A(-3,6,-5) та B(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} » і завдання: Знайти напрямні орт вектора \overrightarrow{AB} .

 $a_0 = (\underline{;},\underline{;}).$

Якщо відповідь (9/17;-12/17;8/17), то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про помилку «Відповідь неправильна! $\vec{a}_0 = \left(\frac{9}{17}, -\frac{12}{17}, \frac{8}{17}\right)$ ».

Крок 7. Відображення умови «Чи колінеарні вектори $\vec{c_1} = -\vec{a} + 4\vec{b}$ і $\vec{c_2} = 3\vec{a} - 2\vec{b}$, побудовані на векторах $\vec{a} = \{2, -2, 3\}$ і $\vec{b} = \{3, 1, -1\}$ » і завдання: Знайти вектор $\vec{c_1}$.

 $\overrightarrow{c_1} = \{ \underline{;} \underline{;} \}.$

Якщо відповідь {10;6;-7}, то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про помилку «Відповідь неправильна! $\vec{c_1} = \{(-2) + 4 \cdot 3; 2 + 4 \cdot 1; (-3) + 4 \cdot (-1)\} =$

 $=\{-2+12; 2+4; -3-4\} = =\{10; 6; -7\}$ ».

Крок 8. Відображення умови «Чи колінеарні вектори $\vec{c_1} = -\vec{a} + 4\vec{b}$ і $\vec{c_2} = 3\vec{a} - 2\vec{b}$, побудовані на векторах $\vec{a} = \{2, -2, 3\}$ і $\vec{b} = \{3, 1, -1\}$ » і завдання: Знайти вектор $\vec{c_2}$.

 $\overrightarrow{c_2} = \{ \underline{\quad}; \underline{\quad}; \underline{\quad} \}.$

Якщо відповідь {0;-8;11}, то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про помилку «Відповідь неправильна! $\vec{c_2} = \{3 \cdot 2 - 2 \cdot 3; 3 \cdot (-2) - 2 \cdot 1; 3 \cdot 3 - 2 \cdot (-1)\} =$

={6 - 6; -6 - 2; 9 + 2} = {0; -8; 11}».

Крок 9. Відображення умови «Чи колінеарні вектори $\vec{c_1} = -\vec{a} + 4\vec{b}$ і $\vec{c_2} = 3\vec{a} - 2\vec{b}$, побудовані на векторах $\vec{a} = \{2, -2, 3\}$ і $\vec{b} = \{3, 1, -1\}$ » і завдання: Перевірити умову $\vec{c_2} = n \cdot \vec{c_1}$.

 $n \cdot \overrightarrow{c_1} = \{ \underline{\qquad}; \underline{\qquad} \}.$

Якщо відповідь 0;{0;0;0}, то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про помилку «Відповідь неправильна! $n = \frac{0}{10} = 0$, $0 \cdot \vec{c_1} = \{0 \cdot 10; 0 \cdot 6; 0 \cdot (-7)\} = \{0; 0; 0\}$ ».

Крок 10. Відображення умови «Чи колінеарні вектори $\vec{c_1} = -\vec{a} + 4\vec{b}$ і $\vec{c_2} = 3\vec{a} - 2\vec{b}$, побудовані на векторах $\vec{a} = \{2, -2, 3\}$ і $\vec{b} = \{3, 1, -1\}$?» і варіантів відповіді:

- Так як $n \cdot \vec{c_1} = \vec{c_2}$, то вектори колінеарні;
- Так як $n \cdot \overrightarrow{c_1} \neq \overrightarrow{c_2}$, то вектори не колінеарні.

Якщо вибрано другий варіант, то зараховується 1 бал і відбувається перехід на наступний крок. В іншому разі виводиться повідомлення про помилку «Відповідь неправильна! Так як $n \cdot \vec{c_1} \neq \vec{c_2}$, то вектори не колінеарні».

Крок 11. Підрахунок кількості набраних балів. Виведення отриманої оцінки.

3.3. Опис засобів програмного забезпечення

React (старі назви: React.js, ReactJS) — відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту вебсторінки, з якими стикаються в розробці односторінкових за стосунків (рис. 3.1). Розробляється Meta (paніше Facebook) і спільнотою індивідуальних розробників.

	Support Ukraine 🗮 Help Provide Humanitarian Aid to Ukraine.								^
React	Q Search	Ctrl K	Learn	Reference	Community	Blog	C	0	
	BLOG >								Î
	Introducing react.dev								
	Today we are thrilled to launch react.dev, the new home for React and its documentatio post, we would like to give you a tour of the new site.	n. In this							
	tl;dr								
	The new React site (react.dev) teaches modern React with function components and Hooks.								
	We've included diagrams, illustrations, challenges, and over 600 new interactive examples.								
	The previous React documentation site has now moved to legacy.reactjs.org.								
	New site, new domain, new homepage								
	First, a little bit of housekeeping.								
	To celebrate the launch of the new docs and, more importantly, to clearly separate the old and the new	N							
	content, we've moved to the shorter react.dev domain. The old reactjs.org domain will now redirect he	ere.							
	The old React docs are now archived at legacy.reactls.org. All existing links to the old content will auto redirect there to avoid "breaking the web", but the legacy site will not get many more updates.	omatically							
	Believe it or not, React will soon be ten years old. In JavaScript years, it's like a whole century! We've r	refreshed							
	the React homepage to reflect why we think React is a great way to create user interfaces today, and	updated							
	the getting started guides to more prominently mention modern React-based frameworks.								~

Рисунок 3.1 – Сайт react.dev

React дозволяє розробникам створювати великі вебзастосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React обробляє тільки користувацький інтерфейс у застосунках. Це відповідає видові у шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови вебзастосунків. Як бібліотеку

інтерфейсу користувача React найчастіше використовують разом з іншими бібліотеками, такими як Redux.

В даний час React використовують Khan Academy, Netflix, Yahoo, Airbnb, Sony, Atlassian та інші [8].

React - це одна з найбільш популярних бібліотек JavaScript для розробки користувацьких інтерфейсів. Ось деякі переваги використання React:

1. Компонентна архітектура: React побудований на компонентній архітектурі, що дозволяє розбити складну структуру користувацького інтерфейсу на менші і простіші компоненти. Це забезпечує більшу гнучкість і повторне використання коду.

2. Висока продуктивність: React використовує віртуальну модель DOM (Virtual DOM), яка забезпечує більшу продуктивність порівняно з роботою з реальним DOM. Віртуальний DOM дозволяє React здійснювати оптимальні зміни в DOM лише тоді, коли вони потрібні, що зменшує кількість перерисовок і забезпечує швидку роботу додатка.

3. Спрощена робота зі станом: React має вбудовану систему керування станом додатка, що дозволяє спростити роботу зі станом компонентів. Це забезпечує більшу читабельність коду і полегшує налагодження додатка.

4. *Розширюваність*: React має багато сторонніх бібліотек та інструментів, що дозволяють розширити його функціональність та додати нові функції до додатка.

5. Можливість використовувати на стороні клієнта та сервера: React можна використовувати як на стороні клієнта, так і на стороні сервера. Це забезпечує більшу гнучкість у розробці додатків та спрощує їхнє впровадження.

6. *Велика спільнота*: React має велику спільноту розробників, яка створює багато корисних інструментів, бібліотек і плагінів.

Незважаючи на те, що React є однією з найпопулярніших бібліотек для розробки користувацьких інтерфейсів, він також має кілька недоліків:

1. *Необхідність вивчення додаткових інструментів*: Для повноцінної розробки веб-додатків з React необхідно вивчити додаткові інструменти, такі як Redux, Jest, Enzyme тощо. Це може зайняти додатковий час та зусилля.

2. *Складність*: React має складніший синтаксис порівняно зі звичайним HTML та CSS. Він вимагає від розробника більш високого рівня володіння JavaScript із знанням сучасних практик.

3. *Інтеграція зі старим кодом*: Інтеграція React зі старим кодом може бути проблемою, оскільки React вимагає більш сучасних технологій та структури додатка.

4. *Ресурсоємність*: Використання React може вимагати більш великих ресурсів, таких як пам'ять та процесор, що може призвести до повільної роботи додатка на старих комп'ютерах або мобільних пристроях.

5. Залежність від сторонніх бібліотек: React зазвичай використовується разом зі сторонніми бібліотеками та інструментами, що може створювати проблеми з інтеграцією та підтримкою.

6. *Неуніверсальність*: React не підходить для всіх видів вебдодатків, особливо для тих, які вимагають багато анімацій та складних взаємодій з користувачем. Для таких випадків можуть бути кращим вибором інші фреймворки або бібліотеки.

Одним з важливих компонентів є **react-latex** – це пакет, який надає компонент React для відображення математичних формул у веб-додатках. Він використовує бібліотеку KaTeX, що забезпечує швидкий та легкий рендеринг математичних формул на стороні клієнта.

Завантаження react-lateх можна здійснити з npm або yarn. Після цього потрібно імпортувати компонент Latex та передати йому рядок з математичною формулою у вигляді властивості children.

Отже, наприклад, для відображення формули $E = mc^2$ можна використати наступний код:

import { Latex } from 'react-latex';

```
function App() {
  return (
      <Latex>
      E = mc^2
      </Latex>
  );
}
```

react-latex також підтримує додаткові налаштування для настройки форматування та розміру шрифту, такі як displayMode, throwOnError, errorColor, macros, minRuleThickness тощо.

Крім того, react-latex має підтримку серверного рендерингу, що дозволяє рендерити математичні формули на сервері та передавати їх як HTML-розмітку клієнту, що забезпечує покращення продуктивності та доступності для користувачів.

Загалом, react-latex є корисним пакетом для розробників, які працюють з математичними формулами у веб-додатках, та може значно спростити процес їх відображення.

Visual Studio Code, який також зазвичай називають VS Code — це редактор вихідного коду, створений Microsoft із Electron Framework для Windows, Linux і macOS (рис. 3.2). Функції включають підтримку налагодження, підсвічування синтаксису, інтелектуальне завершення коду, фрагменти, рефакторинг коду та вбудований Git. Користувачі можуть змінювати тему, комбінації клавіш, параметри та встановлювати розширення, які додають функціональність.



Рисунок 3.2 – Сайт Visual Studio Code

В опитуванні розробників Stack Overflow 2022 серед 71 010 респондентів Visual Studio Code назвали найпопулярнішим інструментом середовища розробника, при цьому 74,48% повідомили, що вони ним користуються [9].

Visual Studio Code (VS Code) - це безкоштовний, відкритий для розробників текстовий редактор. Він має численні особливості, які дозволяють розробникам підвищити ефективність своєї роботи та зробити процес розробки більш приємним. Ось декілька з них:

1. Розширення та плагіни. VS Code має велику кількість розширень та плагінів, що дозволяють налаштувати редактор під конкретні потреби розробника. Розширення можна знайти в магазині VS Code, або ж розробити своє власне.

2. Швидкість та продуктивність. VS Code відмінно працює навіть з великими проектами, завдяки своєму швидкому та продуктивному ядру. Також, VS Code має вбудовану підтримку Git, що дозволяє легко працювати з репозиторіями прямо з редактора.

3. *Debugging*. VS Code має вбудовану підтримку відлагоджування коду для більшості мов програмування, що дозволяє розробникам швидко знаходити та виправляти помилки.

4. *IntelliSense*. VS Code має потужну підсвічування та автодоповнення коду за допомогою функції IntelliSense, яка підтримує багато мов програмування.

5. Вбудований термінал. VS Code має вбудований термінал, що дозволяє розробникам взаємодіяти зі своїм середовищем безпосередньо з редактора.

6. *Мультиплатформеність*. VS Code працює на Windows, macOS та Linux, що робить його доступним для розробників на будь-якій платформі.

7. Відлагодження в браузері. VS Code має вбудовану можливість відлагодження коду в браузері

4. ПРАКТИЧНА ЧАСТИНА

4.1. Блок-схема алгоритму роботи тренажера

Спочатку розроблено загальну блок-схему алгоритму роботи тренажера. На рисунку 4.1 зображено цю блок-схему.



Рисунок 4.1 – Блок-схема алгоритму роботи тренажера

Алгоритми тестування і розв'язання завдань описується окремими блок-схемами (рис. 4.2, 4.3).



Рисунок 4.2 – Блок-схема алгоритму тестування



Рисунок 4.3 – Блок-схема алгоритму розв'язання завдань

4.2. Опис програмної реалізації

Редактор коду Visual Studio підтримує React.js IntelliSense та навігацію кодом із коробки (рис. 4.4).

Щоб використовувати генератор, а також запустити сервер додатків React, потрібно встановити середовище виконання JavaScript Node.js і npm (менеджер пакетів Node.js). npm входить до складу Node.js, який можна завантажити та встановити із завантажень Node.js.



Рисунок 4.4 – Запуск стандартного проєкту

Щоб перевірити, чи правильно встановлено Node.js і npm на машині, можна ввести node --version і npm --version у терміналі або командному рядку.

Тепер можна створити нову програму React, ввівши:

npx create-react-app my-app

де my-аpp — це назва папки для програми. Створення програми React і встановлення її залежностей може зайняти кілька хвилин.

Якщо раніше вже інсталювалось create-react-app глобально за допомогою npm install -g create-react-app, слід видалити пакет за допомогою npm uninstall -g create-react-app, щоб гарантувати, що npx завжди використовує останню версію.

Для швидкого запуску програми React необхідно перейти до нової папки та ввести npm start, щоб запустити веб-сервер і відкрити програму в браузері:

cd my-app npm start Відобразиться логотип React і посилання на «Learn React» на http://localhost:3000 у браузері. Залишиємо веб-сервер запущеним, поки працюватимемо з кодом VS.

Щоб відкрити програму React у VS Code, потрібно відкрити інший термінал або вікно командного рядка, перейти до папки my-app і ввести код:

cd my-app

code.

У Провіднику файлів можна побачити файл програми README.md Markdown. Тут міститься багато чудової інформації про програму та React загалом. Хороший спосіб перегляду README — використання VS Code Markdown Preview. Можна відкрити попередній перегляд у поточній групі редакторів (Markdown: Open Preview Ctrl+Shift+V) або в новій групі редакторів збоку (Markdown: Open Preview to the Side Ctrl+K V). Після цього буде гарне форматування, навігація за гіперпосиланнями до заголовків і підсвічування синтаксису в блоках коду (рис. 4.5).



Рисунок 4.5 – Файл програми README.md

Тепер розгортаємо папку src і вибераємо файл index.js. Можна помітити, що VS Code має підсвічування синтаксису для різних елементів вихідного коду, і якщо помістити курсор на круглу дужку, відповідна дужка також буде виділена (рис. 2.6).



Рисунок 4.6 – Файл програми index.js

VS Code використовує службу мови TypeScript для аналізу коду JavaScript і має функцію під назвою Automatic Type Acquisition (ATA). ATA збирає файли декларації типу npm (*.d.ts) для модулів npm, на які посилається файл package.json. Якщо вибрати метод, також отримається довідка щодо параметрів.

Також встановимо інструмент командного рядка ESLint:

npm install -g eslint

Потім інсталюємо розширення ESLint, перейшовши до перегляду розширень і ввівши «eslint» (рис. 4.7).



Рисунок 4.7 – Розширення ESLint

Після встановлення розширення ESLint і перезавантаження коду VS необхідно створити файл конфігурації ESLint .eslintrc.js. Створюємо його за допомогою розширення ESLint: команда Create ESLint configuration з панелі команд (Ctrl+Shift+P) (рис. 4.8).



Рисунок 4.8 – Команда Create ESLint configuration

Це створить файл .eslintrc.js у корені проекту.

Тепер можна змінювати App.js відповідно до алгоритму роботи тренажера.

Для створення функціонального React-компонента потрібно імпортувати потрібні бібліотеки та ресурси, і задати початковий стан для додатку.

import React, { useState } from 'react'; import questions from './questions'; import tasks from './tasks'; import './App.css'; 1. import React, { useState } from 'react';: Цей рядок імпортує дві речі з бібліотеки React. Перше - це головний об'єкт React, який використовується для створення React-компонентів. Друге - useState імпортується з React і використовується для створення змінних стану в компонентах.

2. questions i tasks імпортуються з файлів './questions' i './tasks'. Це дані для заповнення додатку, такі як питання для тестування та завдання для виконання.

3. './App.css': Цей рядок імпортує CSS-файл для стилізації компоненту.

Створюється функціональний компонент Арр, який буде представляти основний вміст додатку.

```
function App() {
  return (
      <div className="App">
      ...
      </div>
  );
}
```

export default App;

Компонент Арр відображає вміст в обгортці <div className="App">. На початку компоненту виводиться зображення з класом Lang.

```
<img className='Lang'
src={language === 'en' ? '/icon/english_icon.png' : '/icon/ukrainian_icon.png'}
alt="Moвa"
/>
```

Після зображення відображається заголовок великим шрифтом, який представляє назву тренажера для дистанційного курсу «Алгебра та геометрія» з теми «Елементи векторної алгебри».

<h1>Тренажер дистанційного курсу «Алгебра та геометрія» з теми «Елементи векторної алгебри»</h1> Нижче знаходяться два абзаци (тег). Перший абзац виводить інформацію про розробника курсу.

Розробник: Косогов Т.Л.

Другий абзац виводить текст, який міститься в змінній pText. Змінна pText відображає текстовий контент, який, може змінюватися в процесі використання додатку.

{pText}

Наступною частиною компонента є контейнер, що містить дві кнопки. Ці кнопки динамічно змінюють свій текст та функцію обробки подій на основі значень змінних b1Text, b2Text та isButtonActive.

```
<div>
           <button
            onClick={b1Text
                                        'Переглянути
                                                         теоретичний
                                                                         матеріал'
                                                                                      ?
                                ____
showNewContentHandler : showNewTestHandler}
            disabled={!isButtonActive}
            className={isButtonActive ? 'active-button' : 'inactive-button'}
           >
            {b1Text}
           </button>
           <button
                                                                                     ?
            onClick={b2Text
                                ===
                                        'Перейти
                                                           практичної
                                                                         частини'
                                                    ДО
changeBContentHandler : showNewTaskHandler}
            disabled={!isButtonActive}
            className={isButtonActive ? 'active-button' : 'inactive-button'}
           >
            {b2Text}
           </button>
          </div>
```

Перша кнопка визначається таким чином:

 onClick: Ця властивість вказує на функцію, яка викликається при кліку на кнопку. Функція обробки подій може бути або showNewContentHandler, або showNewTestHandler, залежно від умови.

• disabled: Вказує, чи кнопка активна (якщо isButtonActive true) чи вимкнена (якщо isButtonActive false).

• className: Визначає клас кнопки (якщо isButtonActive true, то 'active-button', інакше 'inactive-button').

• b1Text: Текст кнопки, який визначається відповідно до значення b1Text.

Друга кнопка працює аналогічно першій, але використовує b2Text і може викликати changeBContentHandler або showNewTaskHandler.

На останніх двох рядках компоненту відбувається відображення вмісту в залежності від значень showNewTest та showNewTask. Якщо showNewTest дорівнює true, то викликається функція renderQuiz, і відображається тест. Якщо showNewTask дорівнює true, то викликається функція renderTask, і відображаються завдання.

{showNewTest && renderQuiz()}
{showNewTask && renderTask()}

Потім визначаються початкові стани за допомогою useState:

• showNewTest i showNewTask: Показники, які вказують, чи потрібно відображати тест чи завдання.

• isButtonActive: Вказує, чи активні кнопки.

• pText, b1Text, b2Text: Текстові рядки для відображення на головному екрані.

const [showNewTest, setShowNewTest] = useState(false); const [showNewTask, setShowNewTask] = useState(false); const [isButtonActive, setIsButtonActive] = useState(true); const [pText, setPText] = useState('); const [b1Text, setB1Text] = useState('Переглянути теоретичний матеріал'); const [b2Text, setB2Text] = useState('Перейти до практичної частини');

Тепер визначаються функції-обробники подій.

Функція changeBContentHandler викликається при натисканні на кнопку та спрямована на зміну вмісту кнопок та тексту. Вона встановлює новий текст в стан setPText('Здійсніть вибір'), що змінює відображений текст на сторінці, встановлює нові значення в стан setB1Text і setB2Text. Нові значення цих текстів - 'Пройти тестування' і 'Виконати завдання'.

const changeBContentHandler = () => { setPText('Здійсніть вибір'); setB1Text('Пройти тестування'); setB2Text('Виконати завдання'); };

Функція showNewContentHandler викликається при натисканні на кнопку і спрямована на відображення теоретичного матеріалу. Вона використовує window.open, щоб відкрити нове вікно браузера з вмістом, який має шлях process.env.PUBLIC_URL + "/Theory.pdf". Це PDF-файл із теоретичним матеріалом, який відкривається у новому вікні браузера.

```
const showNewContentHandler = () => {
  window.open(process.env.PUBLIC_URL + "/Theory.pdf", "_blank");
};
```

Функція showNewTestHandler викликається при натисканні на кнопку або дію, спрямовану на відображення тестування. Для цього вона встановлює стан setShowNewTest(true). Також значення стану setIsButtonActive(!isButtonActive) інвертується, що забороняє використання стартових кнопок на сторінці.

```
const showNewTestHandler = () => {
  setShowNewTest(true);
  setIsButtonActive(!isButtonActive);
};
```

Функція showNewTaskHandler діє анаоргічно попередній, але відображає завдання.

```
const showNewTaskHandler = () => {
  setShowNewTask(true);
  setIsButtonActive(!isButtonActive);
};
```

Наступним кроком визначаються стани і функції для тестування:

- currentQuestionIndex: Поточний індекс питання в тесті.
- score: Кількість балів, набраних користувачем.
- selectedAnswers: Масив індексів вибраних відповідей.
- showScore: Показник, чи відображати результат тестування.

const [currentQuestionIndex, setCurrentQuestionIndex] = useState(0); const [score, setScore] = useState(0); const [selectedAnswers, setSelectedAnswers] = useState([]); const [showScore, setShowScore] = useState(false);

Функція handleAnswerButtonClick приймає answerIndex, який представляє індекс вибраної відповіді у масиві варіантів відповідей. Потім спробує визначити, чи користувач вже обрав цей варіант відповіді, шляхом перевірки, чи індекс answerIndex вже існує у масиві selectedAnswers.

Якщо answerIndex вже існує в selectedAnswers, то він видаляється з масиву (за допомогою .splice()). Якщо – додається до масиву newSelectedAnswers.

Нарешті, оновлений newSelectedAnswers встановлюється в стан setSelectedAnswers, щоб оновити вибрані відповіді користувача.

```
const handleAnswerButtonClick = (answerIndex) => {
  const newSelectedAnswers = [...selectedAnswers];
  if (newSelectedAnswers.includes(answerIndex)) {
    newSelectedAnswers.splice(newSelectedAnswers.indexOf(answerIndex), 1);
  } else {
    newSelectedAnswers.push(answerIndex);
  }
  setSelectedAnswers(newSelectedAnswers);
};
```

Для розрахунку результатів тестування після вибору відповідей на питання викликається calculateScore.

Спочатку вона перевіряє, чи кількість обраних відповідей (selectedAnswers) не перевищує максимальну кількість відповідей, яку дозволено відповідно до градації питання. Якщо все в порядку, вона продовжує обраховувати бали.

Функція визначає змінну res для зберігання кількості отриманих балів. Вона проходить по всіх варіантах відповідей для поточного питання (зберігаючи індекс в index і об'єкт в answerOption), перевіряючи, чи користувач обрав цей варіант (selectedAnswers.includes(index)) і чи ця відповідь є правильною (answerOption.isCorrect). В залежності від градації питання (представленої як questions[currentQuestionIndex].grade), бали додаються до res. Якщо градація - 1, то додається 1 бал, в іншому випадку додається 0.5 балів. В кінці сума балів (res) додається до загального результату (зберігається в стані score за допомогою setScore).

Після обрахунку результатів, функція перевіряє, чи є ще наступне питання. Якщо так, вона очищає масив selectedAnswers, оновлює індекс поточного питання (currentQuestionIndex) і продовжує тестування. Якщо наступного питання немає, функція встановлює стан setShowScore(true), щоб показати результати тестування.

```
const calculateScore = () => \{
         if (selectedAnswers.length <= questions[currentQuestionIndex].grade) {
          let res = 0;
           questions[currentQuestionIndex].answerOptions.forEach((answerOption,
                                                                                      index)
=> {
            if (selectedAnswers.includes(index) && answerOption.isCorrect) {
             questions[currentQuestionIndex].grade === 1 ? res++ : res=res+0.5;
            }
           });
          setScore(score + res);
         const nextQuestionIndex = currentQuestionIndex + 1;
         if (nextQuestionIndex < questions.length) {
           setSelectedAnswers([]);
          setCurrentQuestionIndex(nextQuestionIndex);
         } else {
           setShowScore(true);
         }
        };
```

За відображення тестового питання або результатів тестування в залежності від значення стану showScore відповідає функція renderQuiz.

Функція renderQuiz перевіряє, чи змінна showScore дорівнює true. Якщо це так, це означає, що тест завершено і користувачу потрібно відобразити результати тестування.

Кнопка "Вихід" <button> дозволяє користувачеві завершити тестування і оновити сторінку (функція window.location.reload()).

<div className="score-section"> <h2>Baш результат: {score} / {questions.length}</h2> <button onClick={()=>{window.location.reload();}}>Bиxiд</button> </div>

Якщо showScore дорівнює false, це означає, що тестування ще не завершено, і функція повертає блок з класом "question-section", який містить наступне:

• Заголовок <h3> з номером поточного питання: "Питання {currentQuestionIndex + 1}".

• Текст питання, який береться з масиву questions[currentQuestionIndex].questionText.

• Список варіантів відповідей, які виводяться як кнопки, де користувач може обирати відповіді. Кожна кнопка представляє варіант відповіді і має обробник подій onClick, який викликає функцію handleAnswerButtonClick(index) при натисканні на кнопку. Клас кнопок "selected" встановлюється, якщо варіант відповіді був вибраний користувачем (якщо selectedAnswers.includes(index) відповідає true).

• Кнопка "Далі" <button> дозволяє користувачеві перейти до наступного питання і викликає функцію calculateScore() при натисканні.

<div className="question-section">
 <h3>Питання {currentQuestionIndex + 1}:</h3>
 {questions[currentQuestionIndex].questionText}
 <div className="answer-options">
 {questions[currentQuestionIndex].answerOptions.map((answerOption, index)
 <button</pre>

=>(

} }

key={index}
onClick={() => handleAnswerButtonClick(index)}

```
className={selectedAnswers.includes(index) ? 'selected' : "}
>
{answerOption.answerText}
</button>
))}
</div>
<button onClick={() => calculateScore()} className='next'>Далi</button>
</div>
```

Питання та варіанти відповідей для тестування містить масив questions (див. Додаток А). Кожне питання представлене об'єктом у масиві questions. У кожному об'єкті є наступні поля:

• questionText: Текст питання, який може містити HTMLрозмітку та використовувати спеціальні теги для відображення математичних формул.

• answerOptions: Масив об'єктів, який містить варіанти відповідей на питання. Кожен варіант відповіді представлений як об'єкт, що містить answerText (текст відповіді) та isCorrect (логічне значення, що позначає, чи є ця відповідь правильною).

Основна особливість поля answerOptions в кожному об'єкті питання полягає в тому, що текст варіантів відповідей відображається з використанням математичних формул, які перетворюються з допомогою бібліотеки MathJax.

import { MathComponent } from "mathjax-react";

```
const Formula = ({text}) => {
  return (
     <MathComponent tex={String.raw`${text}`} display={false} />
  )
}
```

Для цього використовується компонент <MathComponent> з бібліотеки MathJax. <MathComponent> приймає два важливих параметри:

• tex: Текст математичної формули у форматі LaTeX, який буде відображено. • display: Логічне значення, яке вказує, чи потрібно відображати формулу у вигляді окремого блоку. Якщо display дорівнює false, то формула відображається в рядку.

Далі реалізується функціонал завдань.

Спочатку встановлюються стани за допомогою useState:

• wrongText ініціалізується пустим рядком і використовується для відображення повідомлення про неправильну відповідь користувача.

• answered ініціалізується значенням false і використовується для визначення, чи вже користувач відповідав на питання.

const [wrongText, setWText] = useState("); const [answered, setAnswered] = useState(false);

Функція taskScore викликається при натисканні на кнопку для перевірки відповіді користувача на практичне завдання. Вона ініціалізує зміну checkAnswer значенням false. Ця змінна буде вказувати на те, чи відповідь користувача вірна.

Потім перевіряє, чи практичне завдання має лише одну правильну відповідь, шляхом перевірки довжини масиву tasks[currentQuestionIndex].rightAnswer. Якщо довжина дорівнює 1, це означає, що правильна відповідь - це одне значення.

```
const taskScore = () => {
  let checkAnswer = false;
  if (tasks[currentQuestionIndex].rightAnswer.length === 1) {
    ...
  }
  else {
    ...
  }
  ...
};
```

Якщо це практичне завдання має лише одну правильну відповідь, функція отримує значення, що введено користувачем з відповідного поля вводу (наприклад, input1Value) за допомогою document.getElementById('input1'). Вона порівнює введене значення з правильною відповіддю, яка знаходиться у tasks[currentQuestionIndex].rightAnswer[0]. Якщо відповідь правильна, то за допомогою перевірки !answered додає 1 бал до загального результату (score) за допомогою setScore(score + 1), встановлює зміну checkAnswer на true, позначаючи, що відповідь вірна. Після цього очищає поле введення (input1Value.value = ") і встановлює зміну wrongText на пустий рядок, якщо вона була попередньо встановлена.

```
const input1Value = document.getElementById('input1');
if (input1Value.value===tasks[currentQuestionIndex].rightAnswer[0]) {
    if (!answered) {
        setScore(score + 1);
    }
    checkAnswer = true;
    setWText(");
    input1Value.value = ";
```

Якщо практичне завдання має більше однієї правильної відповіді, функція перевіряє введення користувача для кожного з трьох можливих полів вводу (input1, input2, i input3) та порівнює їх з відповідними правильними відповідями з масиву tasks[currentQuestionIndex].rightAnswer.

```
const input1Value = document.getElementById('input1');
const input2Value = document.getElementById('input2');
const input3Value = document.getElementById('input3');
if (input1Value.value===tasks[currentQuestionIndex].rightAnswer[0]&&
input2Value.value===tasks[currentQuestionIndex].rightAnswer[1]&&
input3Value.value===tasks[currentQuestionIndex].rightAnswer[2]) {
if (!answered) {
setScore(score + 1);
}
checkAnswer = true;
setWText('');
input1Value.value = '';
input2Value.value = '';
input3Value.value = '';
input3Value.value = '';
}
```

Нарешті, функція перевіряє значення checkAnswer. Якщо воно true, то вона переходить до наступного питання (якщо таке існує) і оновлює answered.

```
setAnswered(true);
```

```
if (checkAnswer) {
  const nextQuestionIndex = currentQuestionIndex + 1;
  if (nextQuestionIndex < tasks.length) {
    setAnswered(false);
    setCurrentQuestionIndex(nextQuestionIndex);
    } else {
    setShowScore(true);
    }
} else {
    setWText(tasks[currentQuestionIndex].wrongAnswer);
}</pre>
```

Функція renderTask перевіряє, чи змінна showScore дорівнює true. Якщо це так, це означає, що користувач закінчив виконання всіх завдань, і функція повертає блок з класом "score-section":

• Заголовок <h2> з результатами користувача: "Ваш результат: {score} / {tasks.length}", де {score} - це кількість балів, отриманих користувачем, і {tasks.length} - загальна кількість практичних завдань.

• Кнопка "Buxiд" <button>, яка дозволяє користувачеві завершити тестування і оновити сторінку.

```
const renderTask = () => {
  if(showScore) {
    return (
        <div className="score-section">
        <h2>Baш peзультат: {score} / {tasks.length}</h2>
        <br/>
        <button onClick={()=>{window.location.reload();}}>Bиxiд</button>
        </div>
    )
    } else {
    ....
    }
}
```

Якщо showScore дорівнює false, це означає, що користувач ще не завершив всі завдання, і функція відображає практичне завдання.

В цьому випадку відображається наступне:

• Заголовок <h3> з номером поточного завдання: "Завдання {currentQuestionIndex + 1}".

• Текст з умовою завдання, отриманий з tasks[currentQuestion-Index].taskDef.

• Текст завдання, отриманий з tasks[currentQuestion-Index].taskText.

• Поля для введення, які призначені для відповідей на завдання, і вони отримуються з tasks[currentQuestionIndex].input.

• Блок з класом 'wrong-answer' для відображення будь-яких повідомлень про неправильні відповіді, які відображаються в змінній wrongText.

• Кнопка "Далі" <button>, яка дозволяє користувачеві перейти до наступного завдання і викликає функцію taskScore() при натисканні.

```
return (
<div className="question-section">
<h3>Завдання {currentQuestionIndex + 1}:</h3>
{tasks[currentQuestionIndex].taskDef}
{tasks[currentQuestionIndex].taskText}
{tasks[currentQuestionIndex].input}
<div className='wrong-aswer'>
{wrongText}
</div>
<button onClick={() => taskScore()} className='next'>Далi</button>
</div>
```

Завдання отримуються з масиву tasks. Кожне завдання має наступну структуру (див. Додаток А):

- taskDef;
- taskText;
- input: Це блок для введення відповіді користувача. Використовується
 <div> для обгортання тексту та <Formula> для відображення математичних формул.
- rightAnswer: Це масив правильних відповідей для завдання.
- wrongAnswer.

4.3. Тестування розробленого тренажера

На сторінці тренажера відображається тема, розробник і можна вибрати дві дії: переглянути теоретичний матеріал або перейти до практичної частини (рис. 4.9).

Тренажер дистанцій	ного курсу «Алгебра та геом	етрія» з теми «Елементи
	Розробник: К	осогов Т.Л.
	Переглянути теоретичний матеріал	Перейти до практичної частини

Рисунок 4.9 – Сторінка тренажера

Якщо натиснути «Переглянути теоретичний матеріал», то у новій вкладці відкриється pdf- файл з теоретичним матеріалом (рис. 4.10).



Рисунок 4.10 – Вкладка з теоретичним матеріалом

При натисненні «Перейти до практичної частини» пропонується здійснити вибір: пройти тестування або виконати завдання (рис. 4.11).

Тренажер дистанційного курсу «Алгебра та гес	—————————————————————————————————————
Розробник	:: Косогов Т.Л.
Здійсн	іть вибір
Пройти тестувания	Виконати задання

Рисунок 4.11 – Перехід до практичної частини

Кнопка «Пройти тестування» відображає перше питання і варіанти відповіді (рис. 4.12). Серед варіантів можна обирати один або декілька, а потім переходити до наступного питання.



Рисунок 4.12 – Перше питання тестів

Таким чином змінюється текст питання і варіанти відповіді, очищується попередній вибір (рис. 4.13).



Рисунок 4.13 – Друге питання тестів

Після проходження тестування відображається результат (рис. 4.14). Для подальшого проходження тренажера доступна кнопка «Вихід». При цьому сторінка оновлюється і можна знову вибрати одну із запропонованих дій.

-
Тренажер дистанційного курсу «Алгебра та геометрія» з теми «Елементи векторної алгебри»
Розробник: Косогов Т.Л.
Здійсніть вибір
Пройти тестуразния Виномати запданни
Ваш результат: 11 / 12
Boog

Рисунок 4.14 – Результат тестування

Тепер перейдемо по кнопці «Виконати завдання». В даному випадку відображається умова першого завдання і саме завдання. Потрібно ввести відповідь для перевірки (рис. 4.15).

-
Тренажер дистанційного курсу «Алгебра та геометрія» з теми «Елементи векторної алгебри»
Разробник: Косогов Т.Л.
Здійсніть вибір
Пройли тестувания Виконати завдания
Завдання 1:
Задано дві точки А(-3,6,-5) та В(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB}
Знайти координати вектора \overrightarrow{AB}
\overrightarrow{AB} = ()

Рисунок 4.15 – Перше завдання

Якщо відповідь неправильна, то виведеться повідомлення про це (рис. 4.16).

Тренажер дистанційного курсу «Алгебра та геометрія» з теми «Елементи векторної алгебри»	
Разробник: Косогов Т.Л.	
Здійсніть вибір	
Пройти тестувания Виконати заядания	
Завдания 1:	
Задано дві точки А(-3,6,-5) та В(6,-6,3). Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB}	
Знайти координати вектора \overrightarrow{AB}	
$\overrightarrow{AB} = ($	
Відповідь неправильнаї \overrightarrow{AB} = (6 - (-3); -6 - 6; 3 - (-5)) = (9; -12; 8)	
A ani	

Рисунок 4.16 – Виведення повідомлення про неправильну відповідь

Якщо відповідь правильна, то відобразиться наступне завдання (рис. 4.17).

	_
Тренажер дистанційного курсу «Алгебра та геометрія» з теми «Елементи векторної алгебри»	
Разробник: Косогов Т.Л.	
Здійсніть вибір	
Пройти тестування Виконати заядання	
Завдання 2:	
Задано дві точки A(-3,6,-5) та B(6,-6,3).Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} $\overrightarrow{AB} = (9; -12; 8)$	
Знайти довжину вектора \overrightarrow{AB}	
Aani Cari Cari Cari Cari Cari Cari Cari Car	

Рисунок 4.17 – Друге завдання

В кінці виконання всіх завдань також виводиться результат і кнопка «Вихід» (рис. 4.18).

Рисунок 4.18 – Результат виконання завдань

Таким чином, розглянуто весь процес проходження тренажера.

4.4. Необхідна користувачу інструкція

Весь функціонал тренажера розглянуто в попередньому пункті, тому загостримо увагу на деяких аспектах, що будуть корисні для користувача.

При переході до практичної частини кнопки «Пройти тестування» і «Виконати завдання» стають неактивними (рис. 4.19).

Тренажер дистанційного курсу ⊲	«Алгебра та геом	летрія» з теми «	Елементи векторної алгебри»
	Розробник:	Косогов Т.Л.	
	Здійсніт	гь вибір	
	Пройтутестування	Виконати завдання	
Питання 1:			
	Що називаєть	ся вектором?	

Рисунок 4.19 – Неактивні кнопки

В тестуванні можна обрати декілька варіантів відповіді, але правильними будуть одна або дві в залежності від питання. Вибрана відповідь має інший стиль (рис. 4.20). Щоб скасувати свій вибір потрібно ще раз натиснути на вибраний варіант.



Рисунок 4.20 – Вибір варіанту відповіді

Кожне питання оцінюється в 1 бал. В тих питаннях, де дві правильні відповіді, кожен варіант оцінюється в 0,5 бала.

Результат тестування показує на скільки питань було обрано правильні відповіді.

При виконанні завдань потрібно вводити відповідь у відповідне поле чи поля (рис. 4.21). Вводиться або ціле число, або дріб. Дробові значення вводяться через «/» (рис. 4.22).

-
Тренажер дистанційного курсу «Алгебра та геометрія» з теми «Елементи векторної алгебри»
Розробник: Косогов Т.Л.
Здійсніть вибір
Пройти тестурания Виконати закдания
Завданкя 1:
Задано дві точки $A(\cdot3,6,\cdot5)$ та $B(6,\cdot6,3).$ Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB}
Знайти координати вектора \overrightarrow{AB}
AB = (9 .12 (8

Рисунок 4.21 – Введення цілих чисел

-
Тренажер дистанційного курсу «Алгебра та геометрія» з теми «Елементи векторної алгебри»
Розробник: Косотов Т.Л.
Здійсніть вибір
Пройти тестувания Виконати закдания
Завдання 3:
Задано дві точки A(-3,6,-5) та B(6,-6,3).Знайти координати, довжину, напрямні косинуси та орт вектора \overrightarrow{AB} $\overrightarrow{AB}=(9,-12;8), \overrightarrow{a} =17$
Знайти напрямні косинуси вектора \overrightarrow{AB}
cos α =[<u>917</u> Далі

Рисунок 4.22 – Введення дробу

Перехід до наступного завдання не відбудеться доти, поки не буде введено правильні значення в поля.

В одному завданні від користувача потребується вибір із списку, що випадає (рис. 4.23).

-
Тренажер дистанційного курсу «Алгебра та геометрія» з теми «Елементи векторної алгебри»
Розробник: Косогов Т.Л.
Здійсніть вибір
Пройти тестувания Виконати завдания
Завдання 10:
Чи колінеарні вектори $\overrightarrow{c_1} = -\overrightarrow{a} + \overrightarrow{d} \cdot \overrightarrow{c_2} = 3\overrightarrow{a} - 2\overrightarrow{b}$, побудовані на векторах $\overrightarrow{a} = \{2, -2, 3\}$ \overrightarrow{i} $\overrightarrow{b} = \{3, 1, -1\}$? $\overrightarrow{c_1} = \{10; 6; -7\}, \overrightarrow{c_2} = \{0; -8; 11\}, n \cdot \overrightarrow{c_1} = \{0; 0; 0\}$
Так як n · ci ≠ ci, то вектори колінеарні вектори не колінеарні

Рисунок 4.23 – Вибір із списку, що випадає

Результат виконання завдань показує на скільки завдань було дано правильні відповіді. Кожне завдання оцінюється в 1 бал. Цей бал зараховується тільки, коли з першого разу було введено правильну відповідь.

Тренажер можна пере запустити в будь-який момент, оновивши вкладку.

ВИСНОВКИ

У даній роботі були розглянуті питання розробки тренажерів для дистанційного навчання з теми "Елементи векторної алгебри" в рамках дисципліни "Алгебра і геометрія". Були проаналізовані різноманітні методи та інструменти розробки тренажерів, а також розглянуті приклади статей з даної тематики.

Зокрема, було розглянуто Mathway та GeoGebra як популярні програмні продукти для розв'язування математичних задач та візуалізації геометричних об'єктів. Також були проаналізовані статті з розробки тренажерів для дистанційного курсу "Алгебра і геометрія". Серед матеріалів дистанційного курсу «Алгебра та геометрія» було розглянуто тренажер з теми «Матриці і визначники», ознайомлено з основними його можливостями.

Програмне забезпечення для тренажера містить такі елементи:

- навчальний матеріал;
- тести та завдання;
- систему оцінювання.

Алгоритм роботи тренажера з теми "Елементи векторної алгебри" складається з 4 етапів. Тестування і розв'язання завдань сформовано окремими блоками.

Для розробки тренажера використовувалися React та Visual Studio Code, оскільки вони є потужними інструментами для розробки вебдодатків. React надає можливість створювати ефективні, масштабовані та зручні веб-інтерфейси, забезпечуючи зручне та просте керування компонентами. Крім того, React є гнучким інструментом, що дозволяє розробникам працювати з різними фреймворками та бібліотеками.

Visual Studio Code надає потужний та зручний інтерфейс для написання та відлагодження коду. Дана IDE має широкі можливості, такі як автодоповнення, кольорову підсвітку синтаксису, інтегровану

підтримку Git, та інші функції, що дозволяють прискорити розробку та полегшити роботу розробників.

Під час навчання та використання різних інструментів та технологій, важливо не лише знати правильні відповіді на питання, але й розуміти неправильні відповіді та помилки. Це дозволяє зрозуміти сутність концепцій та уникнути неправильного застосування їх у практичних задачах. Тому і в тестах, і в завданням вказуються допущені помилки.

Отже, розробка тренажерів є актуальною задачею у сучасній освіті та може забезпечити ефективне дистанційне навчання студентів. Розглянуті методи та інструменти розробки дозволяють створювати зручні та ефективні тренажери, що є важливим джерелом інформації для розробників та викладачів, що працюють у цій галузі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- Дистанційний курс «Алгебра та геометрія. Частина 1» [Електронний ресурс] // Сайт дистанційного навчання, Вищий навчальний заклад Укоопспілки «Полтавський університет економіки і торгівлі». – Режим доступу: https://el.puet.edu.ua/
- Максименко Д.В. Елементи векторної алгебри та аналітичної геометрії: практичний посібник / уклад.: Д.В. Максименко, В.Ф. Мельничук, Л.В. Соколова, Ю.О. Чорноіван. – К.: КНУБА, 2013. – 48 с.
- Гринь С.В. Програмна реалізація розв'язування задач за темою "Пряма на площині" дисципліни "Алгебра і геометрія" / С.В. Гринь, О.О. Валуйська // Структурні зміни в економіці та освіті під впливом інформаційно-комунікаційних технологій: Матеріали міжнародної інтернет-конференції, 3-14.06.2010 року. - Полтава: PBB ПУЕТ, 2010. -С.213-215. – Режим доступу:

http://dspace.puet.edu.ua//handle/123456789/1384

- Панасенко О.М. Програмна реалізація розв'язування задач за темою "Вектори, площини, прямі у просторі" з дисципліни "Алгебра і геометрія" / О.М. Панасенко, О.О. Валуйська // Структурні зміни в економіці та освіті під впливом інформаційно-комунікаційних технологій: Матеріали міжнародної інтернет-конференції, 3-14.06.2010 року. - Полтава: PBB ПУЕТ, 2010. - С. 230-233. – Режим доступу: http://dspace.puet.edu.ua//handle/123456789/1418
- Ломонос Л. М. Вибрані питання математики. Елементи векторної алгебри та аналітичної геометрії: навч. посіб. / Л. М. Ломонос, В. І. Мамчук, Н. П. Муранова. — К. : Вид-во Нац. авіац. ун-ту «НАУ-друк», 2010. — 2-ге вид., стер. — 128 с.
- Ю. В. Ситникова. Лінійна та векторна алгебра у схемах і таблицях : навчальний довідник для самостійного вивчення курсу вищої математики (для студентів 1, 2 курсів денної та заочної форм навчання) / Ю. В. Ситникова, С. М. Ламтюгова, Г. А. Кузнецова ; Харків. нац. ун-

т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2018. – 109 с.

- Кадубовський О.А Аналітична геометрія. Частина І: Елементи векторної алгебри. Метод координат на площині та в просторі: Навчальний посібник / О.А. Кадубовський, О.Л. Кадубовська, Л.Г. Плесканьова – Видання 2-е, виправлене та доповнене. – Слов'янськ, 2010. – 84 с.
- 8. React [Електронний ресурс] / Матеріал з Вікіпедії вільної енциклопедії.. Режим доступу: https://uk.wikipedia.org/wiki/React
- Visual Studio Code [Електронний ресурс] / Матеріал з Вікіпедії вільної енциклопедії.. – Режим доступу: https://uk.wikipedia.org/wiki/Visual_Studio_Code