

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»
В.о. завідувача кафедри
Ігор ШЕЛЕХОВ

(підпис)

18 грудня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційна технологія розроблення віртуальних тренажерів для
розв'язування задач багатовимірної оптимізації»
здобувачки групи ІН.м - 22 Гончаренко Софії Андріївни

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

Софія ГОНЧАРЕНКО

(підпис)

Керівник,
старший викладач,
кандидат фізико-математичних наук

Оксана ШОВКОПЛЯС

(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття освітнього ступеня магістра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН.М-22 Гончаренко Софії Андріївни

1. Тема роботи: « Інформаційна технологія розроблення віртуальних тренажерів для розв'язування задач багатовимірної оптимізації»

затверджую наказом по СумДУ від «06» грудня 2023 року № 1412-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 16 грудня 2023 року

3. Вихідні дані до кваліфікаційної роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.

2) Огляд методів, що використовуються для розв'язування задач багатовимірної оптимізації. 3)

Розроблення віртуального тренажера для розв'язування задач багатовимірної оптимізації. 4) Аналіз

результатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «__» _____.

Завдання прийняв до виконання _____

_____ (підпис)

Керівник _____

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження.</i>		
2	<i>Огляд методів, що використовуються для розв'язування задач багатовимірної оптимізації.</i>		
3	<i>Розроблення віртуального тренажера для розв'язування задач багатовимірної оптимізації.</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти _____

_____ (підпис)

Керівник _____

_____ (підпис)

АНОТАЦІЯ

Записка: 89 стр., 36 рис., 1 додаток, 20 використаних джерел.

Обґрунтування актуальності теми роботи. Актуальність даної кваліфікаційної роботи обумовлена потребою в створенні та розвитку ефективних інструментів для опанування методів розв'язування задач багатовимірної оптимізації, які мають широке застосування та практичну значущість.

Об'єкт дослідження – процес розроблення інтерактивних практичних завдань навчального призначення.

Мета роботи — вивчення та розроблення інформаційної технології для створення віртуальних тренажерів, спрямованих на розв'язання задач багатовимірної оптимізації.

Методи дослідження — аналіз предметної області, переваг та перспектив, огляд існуючих аналогів, інструментарію для вирішення поставленої задачі, розроблення самого віртуального тренажера та тестування й оцінка його ефективності.

Результати – розроблено інтерактивний тренажер, який пропонує розглянути два методи вирішення задач багатовимірної оптимізації, а саме короткий огляд та повторення теоретичного матеріалу, а також проходження практичного завдання для закріплення набутих знань та навичок.

БАГАТОВИМІРНА ОПТИМІЗАЦІЯ, ВІРТУАЛЬНИЙ ТРЕНАЖЕР,
ІНТЕРАКТИВНІСТЬ, МЕТОД НЕЛДЕРА-МІДА, МЕТОД ХУКА-ДЖИВСА,
JAVASCRIPT

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ РОБОТИ	7
1.1 Аналіз предметної галузі.....	7
1.2 Короткий огляд переваг і перспективи застосування тренажерів	12
1.3 Огляд існуючих онлайн тренажерів	16
1.4 Постановка задачі.....	21
2 ВИБІР МЕТОДІВ РІШЕННЯ	22
2.1 Вимоги до навчального об'єкту.....	22
2.2 Основні визначення та види задач.....	26
2.3 Вибір програмного інструментарію	29
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	34
3.1 Короткий опис програмної реалізації та структура.....	34
3.2 Інструкція роботи з тренажером.....	40
3.3 Приклад тестування та результати роботи програми.....	50
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТОК А	59

ВСТУП

Актуальність. В межах сучасного освітнього середовища, де вимоги до підготовки студентів є надто динамічними та мінливими, інтерактивні тренажери виявляються потужним інструментом для створення гнучкого та адаптивного навчального процесу. Їх ефективність полягає в здатності сприяти активній взаємодії між викладачами і студентами, розвивати критичне мислення учасників освітнього процесу, надавати перевагу практичному вивченню навичок через імітацію реальних ситуацій та завдань. Дослідження та розроблення такого тренажеру стає актуальним завданням, спрямованим на створення ефективного та доступного інструменту для навчання і вдосконалення навичок у напрямку розв'язання задач багатовимірної оптимізації.

Об'єкт дослідження – процес розроблення інтерактивних практичних завдань навчального призначення.

Предмет дослідження – інформаційна технологія розроблення віртуального тренажера для розв'язування задач багатовимірної оптимізації.

Гіпотеза. Результати цього дослідження можуть знайти застосування в навчальних і наукових закладах для оптимізації реальних задач, де ефективне вирішення багатовимірних оптимізаційних завдань є ключовим фактором успіху. В освітньому процесі даний тренажер може покращити ефективність роботи студентів та автоматизувати роботу викладачів (таку як ведення звітності, створення тестових чи модульних контролів та інше). Даний тренажер можна використовувати в якості практичного завдання, так і для самостійної роботи, а також для контролю знань.

Новизна даної кваліфікаційної роботи полягає в розробленні унікального віртуального тренажера, який покриває вивчення теми розв'язання задач багатовимірної оптимізації. Під час розробки було враховано як реалізацію

спрямованого на користувача інтерфейсу, так і зрозумілого та доступного функціоналу, що суттєво відрізняє розроблений тренажер від аналогів.

Структура роботи складається зі вступу, аналізу предметної галузі, огляду переваг та перспектив використання віртуальних тренажерів, оглядку існуючих онлайн тренажерів, постановки задачі дослідження, вибору методів рішення, опис основних визначень та методів вирішення задач, вибору програмного інструментарію, інформаційної та програмної реалізації, висновків, списку використаних джерел та додатків.

Зв'язок роботи з науковою темою. Кваліфікаційна робота виконана на кафедрі комп'ютерних наук та пов'язана з виконанням науково-дослідної роботи № 0120U103407 «Застосування технологій games learning, blended learning, віртуальної та доповненої реальності в навчальному процесі» (2020-2025).

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ РОБОТИ

1.1 Аналіз предметної галузі

У сфері інформаційно-комунікаційних технологій освіти поняття «інтерактивний тренажер» визначає навчальний інструмент, який дає змогу користувачам, зокрема учням та студентам, взаємодіяти з навчальним матеріалом у специфічних, активних форматах [1]. Це може бути електронний, веб-заснований чи програмний продукт, який забезпечує можливість практичного вивчення концепцій, закріплення вже набутих навичок, проходження кроків для виконання алгоритму, процесу тощо.

Застосування віртуальних тренажерів позитивно вирішує безліч завдань, серед яких [2]:

- інформування (підручники, довідники, навчальні плакати, відеозаписи, алгоритми, інформаційно-технологічні карти, навчальні програми та тренажери);
- організації (алгоритми, навчальні програми та тренажери, покажчики);
- управління (навчальні тренажери та програми, тести).

Ключовою особливістю роботи з таким навчальним інструментом є елемент залучення користувача до «комунікації» з тренажером. Це спілкування може включати в себе: відповіді на питання (як обрання студентом одного варіанту відповіді серед багатьох, так і самостійне її вписування; встановлення відповідності; побудова графіків і діаграм; виконання малюнку тощо), вирішення завдань та, навіть, симуляцію певних сценаріїв (наприклад, проведення медичних маніпуляцій, налаштування технічних приладів та інше). Можливості відтворення будь-якого сценарію безмежні, головне вміти застосувати певний інструментарій та розробити план проходження тренажера. В цьому і полягає

перевага створення і застосування тренажера в освітньому процесі – його складність і вигляд може бути різноманітним та індивідуальним.

Здатність візуалізувати інформацію та процеси графічно є важливою частиною інтерактивних тренажерів. Інформацію можна подати у вигляді діаграм, схем, таблиць, дво- або тривимірних об'єктів. Такі візуалізації можуть відтворювати не тільки статистичні набори даних, а й симулювати реальні об'єкти, події та ситуації. Тренажери можуть використовувати аудіо- та відеоелементи для кращої демонстрації складних концепцій і теорій, а також для урізноманітнення і розбавлення звичного матеріалу, що подають у вигляді тексту. Дані методики сприяють не тільки кращому розумінню і запам'ятовуванню, а й активному залученню студентів до виконання завдання. Задача може бути описаною не тільки «сухими» словами, а зображатися грою, інтерактивом – що дозволяє працювати з різними категоріями студентів (учні різного віку, з різними інтелектуальними здібностями, які навчаються онлайн чи офлайн тощо).

В контексті дослідження даних освітніх інструментів варто також розглянути поняття імерсивності. Технології навчання, які засновані на ідеї імерсії (процес занурення у різні сфери за допомогою різних органів відчуттів), отримали назву «імерсивні технології» [3]. Наразі вищезазначені методи навчання широко застосовуються у технічних, медичних, інформаційних і біологічних галузях. В межах цього напрямку розробляється потужне програмне забезпечення, а також випробовуються та адаптуються існуючі технологічні продукти, такі як симулятивні методи навчання [4].

Переваги імерсивних тренажерів полягають у їхній здатності створювати виразне та вражаюче навчальне середовище, яке сприяє ефективності роботи студентів протягом освітнього процесу. Такі інструменти дозволяють учням «поглиблюватися» у вивчений матеріал, використовуючи сенсорні, зорові та

звукові форми осягнення нового. Даний підхід стимулює активну участь, в якій студенти не просто спостерігають за інформацією, але і взаємодіють з нею, вирішуючи завдання та відтворюючи сценарії. Спеціально розроблені тренажери можуть симулювати реальні виклики та завдання, що дозволяють учням стикнутися з реальними ситуаціями та проблемами, які вони можуть зустріти у майбутній професійній діяльності. У віртуальному середовищі студенти можуть випробовувати різні стратегії та дії, вчитися на власних помилках без реальних наслідків, що робить навчання ефективним, цікавим та різноманітним [5].

Окрім зазначених вище можливостей, які містить у собі тренажер як освітня технологія, слід розглянути актуальність використання у академічному процесі.

По-перше, різноманітність видів тренажерів дає змогу застосовувати їх за будь-яких обставин. Інтерактивні тренажери в освіті розрізняються за своєю специфікою та призначенням. Ось декілька видів інтерактивних тренажерів в освіті:

1. Віртуальні навчальні середовища:

- 1.1. Віртуальні лабораторії, в яких емулюють реальні лабораторні умови для вивчення студентами наукових і технічних дисциплін.

- 1.2. Симулятори реальних ситуацій, тобто навчання у віртуальних сценаріях для практики реальних життєвих ситуацій.

2. Інтерактивні навчальні платформи:

- 2.1. Групові інтерактивні завдання, які сприяють співпраці та обміну інформацією між учнями у віртуальному середовищі.

- 2.2. Електронні тести та ігри, що забезпечують ігровий спосіб вивчення та перевірки знань.

3. Віртуальні тренажери для професійного розвитку:

- 3.1. Тренажери для медичної практики, які симулюють хірургічні втручання та інші медичні процедури.
 - 3.2. Симулятори для військового навчання, що використовуються для тренування військового та безпекового персоналу.
4. Імерсивні технології в освіті:
- 4.1. Віртуальна реальність (VR) та доповнена реальність (AR). Забезпечують імерсивні навчальний досвід у віртуальному чи розширеному середовищі.
 - 4.2. Симулятори водіння для освіти дозволяють водійським школам тренувати слухачів у безпечних умовах.
5. Електронні навчальні ігри:
- 5.1. Ігри для вивчення мов. Використовують гру для ефективного вивчення нових мов та лексики.
 - 5.2. Математичні ігри та головоломки, які забезпечують ігровий підхід до вивчення математики та розвитку логічного мислення.
6. Інтерактивні симуляції для навчання наукових концепцій:
- 6.1. Симуляції фізичних експериментів, що дозволяють студентам експериментувати і спостерігати результати віртуальних дослідів.
 - 6.2. Математичні симуляції. Вони допомагають у візуалізації та розумінні математичних концепцій.

Це лише кілька прикладів інтерактивних тренажерів, які використовуються в освіті для різних цілей та вивчення навчальних предметів.

З урахуванням сучасних викликів, таких як пандемія та повномасштабна війна, інтерактивні засоби забезпечують можливість ефективного впровадження

дистанційного навчання, де студенти можуть активно брати участь в освітньому процесі, перебуваючи у будь-якій точці земної кулі.

Використання інтерактивних технологій в освіті дозволяє впроваджувати інновації та залишатися в кривих сучасних технологічних тенденцій, підтримуючи цифровий розвиток. Попри те, що використання технологічних новинок в освіті є доцільним з позиції ефективності навчання, також це важливо для розвитку багатьох сфер і галузей нашої країни. Як зазначають Н. Г. Базавлук та Л. С. Франко у своїх матеріалах до XLII Міжнародної науково-методичної конференції з проблем дистанційної освіти: «Найважливішими умовами формування конкурентоспроможної національної економіки на основі чинників інтелектуально-інноваційного розвитку є: створення найсучаснішого освітнього середовища, орієнтованого на гармонізацію національних і міжнародних стандартів вищої освіти; інтеграція до європейського освітнього простору; впровадження сучасних інноваційних підходів до навчання» [6].

У сучасних реаліях розвинуті країни ставлять за мету реформування системи освіти у таких напрямках, як «навчання вмінню самостійно здобувати потрібну інформацію, виділяти проблеми і знаходити шляхи їх раціонального вирішення, вміти критично аналізувати набуті знання і застосовувати їх для розв'язання нових завдань» [7]. Робити акцент необхідно на розвитку особистісно-орієнтованої освіти та системи дистанційного навчання.

Наразі, під час дистанційного навчання, викладачі стикаються з проблемою втрати зацікавленості учнів в освітньому процесі, після довгих годин сидіння за комп'ютером увага розсіюється, а вмотивованість з часом втрачається. У цьому випадку тренажери є вирішенням проблеми. Вони створюють динамічне та захоплююче навчальне середовище, що сприяє активному залученню студентів у процес заняття.

Інтерактивні інструменти сприяють переходу від пасивного сприйняття інформації до активної участі та взаємодії. Це сприяє кращому розумінню та запам'ятовуванню матеріалу. Тренажери можуть бути використані для створення колективного навчального досвіду, де студенти можуть спільно вирішувати завдання, обговорювати концепції та взаємодіяти, сприяючи розвитку комунікаційних навичок. Інтерактивні інструменти надають можливість студентам самостійно досліджувати та виражати свої ідеї. Це стимулює творчий підхід до навчання і розвиток критичного мислення.

Не варто забувати про актуальність не тільки для учнів, а й для вчителів., адже використання даних технологій сприяє різноманітності методів навчання. Застосування інтерактивних інструментів дозволяє викладачам розширити репертуар методів навчання і пристосовувати їх до різних стилів вивчення. Багато інтерактивних інструментів дозволяють вчителям отримувати зворотний зв'язок та оцінювати роботу студентів в режимі реального часу. Це дає можливість оперативно адаптувати навчальний процес залежно від потреб класу.

Загалом, віртуальні тренажери в освіті є важливим елементом для створення стимулюючого та ефективного навчального середовища в сучасному світі.

1.2 Короткий огляд переваг і перспективи застосування тренажерів

Тренажери, як інноваційний інструмент у сфері навчання, мають значний потенціал та переваги для поліпшення якості освіти і розвитку навичок студентів.

По-перше, вони реалізують інтерактивність та адаптабельність протягом заняття. Віртуальні тренажери можуть відтворювати сценарії різного призначення, що дозволяє студентам взаємодіяти з теоретичними і практичними матеріалами. Нові інформаційні технології пропонують різноманітні потужні інструментарії для опрацювання текстової, числової та графічної інформації, що

становлять основу освітнього середовища. Адаптивні тренажери можуть також підлаштовувати рівень складності в залежності від успішності студента, забезпечуючи оптимальне навчання. Вони адаптуються до рівня знань і навичок користувача, надаючи більші можливості для тих, хто вже має певний рівень експертизи, чи, навпаки, надаючи додаткову допомогу для тих, хто її потребує.

Доступність віртуальних тренажерів стала широкою завдяки розвитку технологій та їхньому впровадженню у різні сфери освіти та професійного навчання. Врахування потреб користувачів з різними рівнями знань і можливостей може бути важливим аспектом. Інтерактивні тренажери часто розробляються з урахуванням доступності для широкого кола аудиторії.

Питання доступності є дуже важливим при розробці. Доступність віртуальних тренажерів визначається рядом факторів, які впливають на те, наскільки легко користувачі можуть отримувати і використовувати ці тренажери [8]. Далі наведено перелік деяких ключових аспектів:

- фінансова доступність. Вартість віртуальних тренажерів та пов'язаних з ними технологій може бути значущим чинником. Безкоштовні або дешеві рішення дозволяють більшій кількості людей використовувати їх. Проте, вища вартість свідчить про якість і можливість частіше оновлювати додаток розробниками.

- технічні вимоги. Вимоги до апаратного забезпечення (комп'ютери, смартфони, VR-пристрої) можуть впливати на те, наскільки легко користувачі можуть отримати доступ до тренажерів. Широке використання мобільних пристроїв може полегшити доступність.

- доступ до мережі. Наявність стабільного та швидкого Інтернет-з'єднання може бути критичною для використання онлайн-тренажерів та інтерактивних ресурсів.

- мова та локалізація. Доступність тренажерів може залежати від того, чи надається інтерфейс та контент у різних мовах, а також чи адаптовані вони до різних культурних контекстів.

- інклюзивність. Чи враховуються різні потреби користувачів, такі як інвалідність людини або відмінності у рівнях знань та вмінь.

- навички та освіта. Рівень технологічної грамотності користувачів впливає на їхню здатність використовувати віртуальні тренажери.

Урахування цих аспектів дозволяє розробникам і педагогам створювати більш доступні та інклюзивні віртуальні тренажери для різних груп користувачів.

Також важливим моментом є можливість використання даних розробок у змішаному і дистанційному навчанні. Наприклад, теоретичні тренажери стають ідеальним інструментом для віддаленої роботи, дозволяючи учням вивчати теоретичний матеріал з будь-якого місця та в будь-який час. Тренажери для відточення практичних навичок також можуть використовуватися навіть вдома, якщо користувач має усі необхідні пристрої для цього.

Окрім вищезазначеного варто згадати про системи оцінювання та звітність, які є невід'ємною складовою роботи будь-якого викладача. Тренажери для повторення теоретичного матеріалу і виконання практичних завдань можуть включати в себе системи автоматизованого оцінювання та звітності, що дозволяє викладачам відстежувати прогрес студентів і надавати звіти про їхню активність й результати. Багато інтерактивних інструментів дозволяють викладачам отримувати зворотний зв'язок та оцінювати роботу студентів в режимі реального часу. Це дає можливість оперативно адаптувати навчальний процес залежно від потреб класу. Окрім того вони надають можливість формативної оцінки, тобто збирають дані про продуктивність користувача в режимі реального часу, дозволяючи викладачам чи студентам відстежувати прогрес і виправляти помилки [9].

Для студентів переваги також є значними. По-перше, сприяння самостійному навчанню. Використання тренажерів може стимулювати самостійне навчання, дозволяючи студентам вивчати теорію у власному темпі та використовувати різноманітні ресурси для додаткового розширення знань. По-друге, інтерактивні завдання та інструменти часто сприяють розвитку критичного мислення, аналітичних навичок і здатності знаходити рішення для проблем різної складності. Також такі інструменти дозволяють створювати персоналізовані шляхи навчання, враховуючи індивідуальні потреби та рівень знань кожного студента. Віртуальні тренажери можуть включати в себе використання спеціалізованого апаратного та програмного забезпечення, що дозволяє студентам отримувати практичний досвід використання актуальних інструментів і технологій, з якими вони познайомляться у своїй майбутній професійній діяльності [9].

Якщо говорити про перспективи застосування віртуальних тренажерів, то варто згадати наступні. Для початку це підтримка неперервного навчання, тобто люди можуть мати можливість навчатися у будь-якому віці, у будь-який час та у будь-якому місці. Використання різних новітніх технологій у навчанні стимулюють і заохочують студентів до постійного навчання, отримання нових та оновлення застарілих знань.

Також ця новація сприяє розвитку дистанційного навчання, адже віддалена освіта зараз тільки набирає оберти у своєму розвитку. Так само і з гібридною моделлю, що поєднує традиційне очне та онлайн-навчання. У майбутньому варто очікувати розширення можливостей у даній сфері, і віртуальні тренажери стануть невід'ємною її складовою. Такі моделі дозволять студентам опановувати матеріал власним темпом, забезпечуючи персоналізацію, доступність і гнучкість.

І це лише деякі з перспектив, які відкриваються для української системи освіти. Сучасні розробки, новації і наполегливість наших педагогів створюють

умови для впровадження цих перспектив у реальність. Ці технології мають потенціал змінити не тільки формат навчання, але й цінності та пріоритети освітнього процесу.

1.3 Огляд існуючих онлайн тренажерів

На різних онлайн ресурсах для широкого доступу, так і обмеженого, можна зустріти вдосталь віртуальних тренажерів, калькуляторів як для учнів шкіл, так і для студентів, і навіть викладачів. Зазвичай такі інструменти покривають певну предметну область, тобто не є широкофункціональними, представляють собою набір певних вправ з обмеженою кількістю дій. Звичайно є й більш широкомасштабні тренажери, але процес їхньої розробки вимагає більшої якості, часу та людського ресурсу. Тому приклади таких тренажерів можна знайти в якості частини якогось платного курсу.

Тим не менш, огляд існуючих онлайн тренажерів у сфері освіти може бути корисним для визначення того, які вже існуючі рішення використовуються та які можливості можна вдосконалити чи розширити. Нижче подано кілька прикладів онлайн тренажерів, які ми розглянемо більш детально.

На рисунку 1.1 представлено інтерактивний математичний тренажер для формування навичок обчислення визначеного інтегралу [10]. Спочатку перед користувачем з'являється повідомлення, в якому прописано коротку інструкцію щодо користування тренажером. Після натискання на кнопку «ОК» студент бачить перед собою набір із 12 прикладів для обчислення визначеного інтеграла. Він або вона мають вписати свою відповідь відповідно до попередньо прочитаних правил, а потім натиснути на кнопку «перевірка завдання».

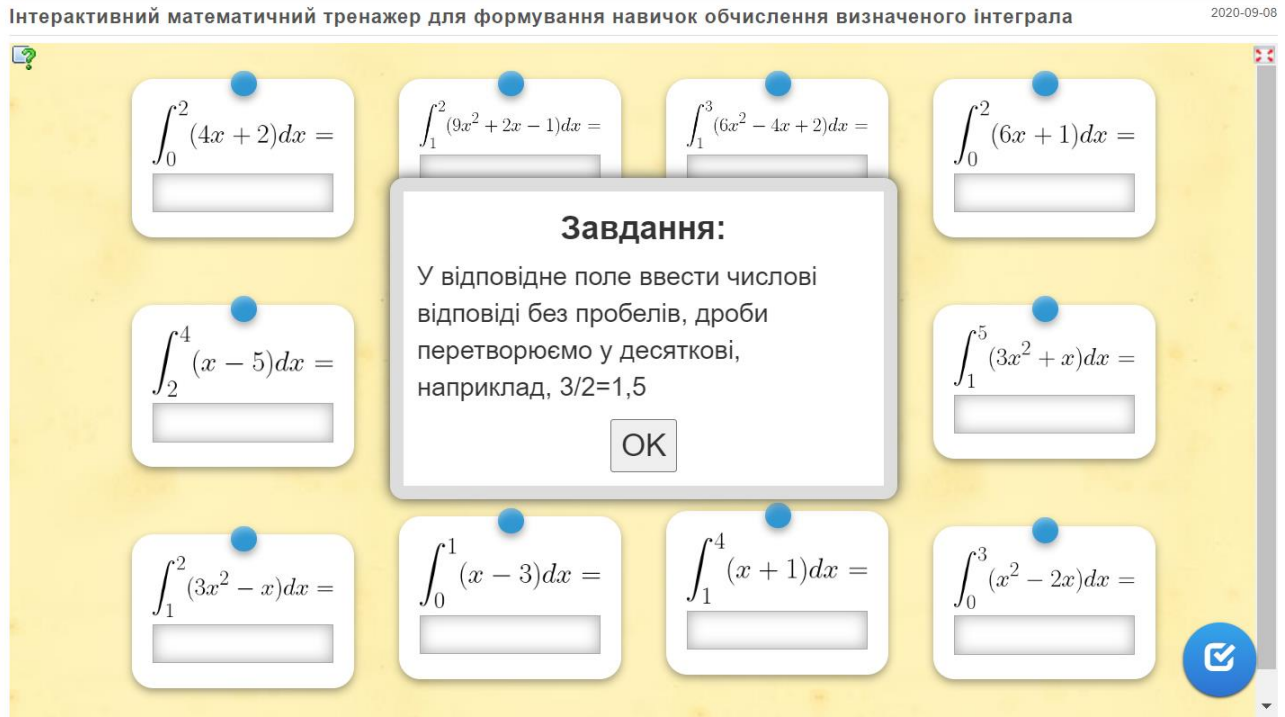


Рисунок 0.1 – Інтерактивний математичний тренажер для формування навичок обчислення визначеного інтегралу

Вважаємо, що даний тренажер є доволі приємним на вигляд, адже він не обтяжує користувача зайвими деталями та елементами. Завдання подані у зручному форматі, присутня інструкція до використання, тобто інтерфейс є зрозумілим і стриманим. Звичайно, набір прикладів є сталим та обмеженим, проте загалом тренажер справляє позитивне враження.

Наступним до розгляду тренажером буде матричний калькулятор [11], який зображений на рисунках 1.2 та 1.3. Для певних обчислювальних функцій можна використовувати як одну (матриця A), так і дві матриці (A і B).

Між матрицями розташовано набір різноманітних функцій для роботи з матрицями. Наприклад, транспонування, знаходження визначника, рангу, оберненої матриці, а також множення матриць і піднесення до степеня. Варто зазначити, що присутні й звичайні функціональні кнопки, такі як «задати розмір матриці» та «очистити поле».

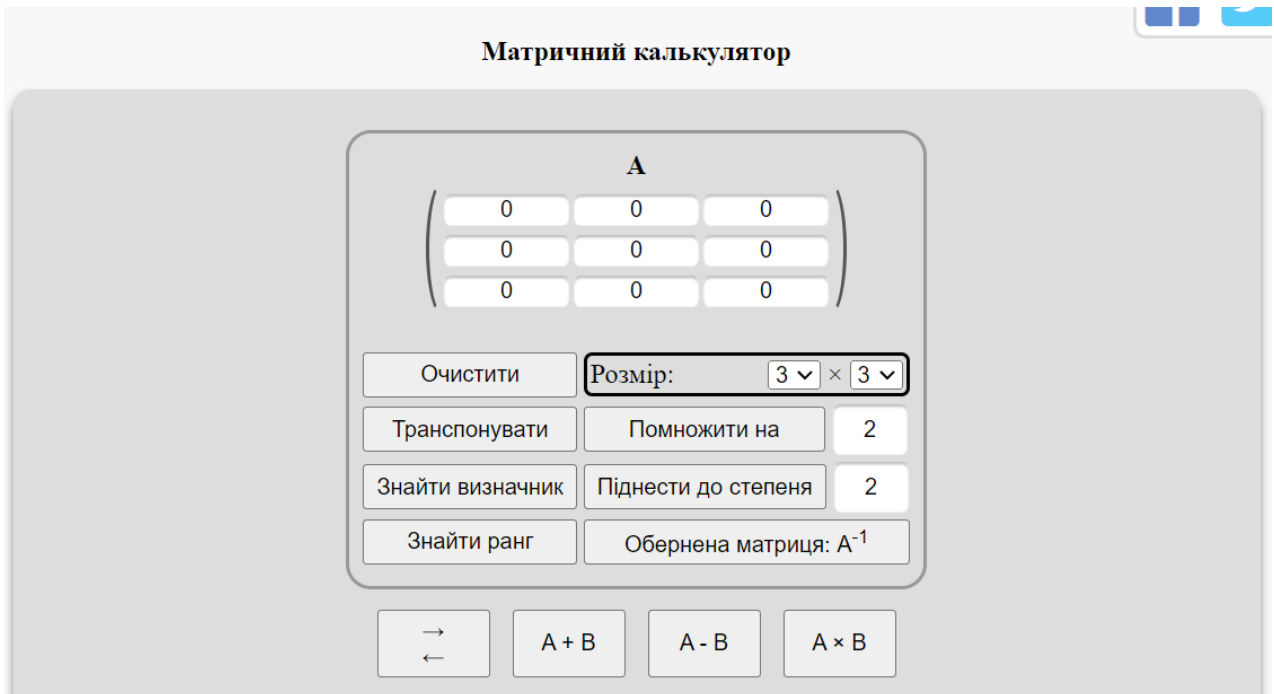


Рисунок 0.2 – Матричний калькулятор, ввід матриці A та функціонал



Рисунок 0.3 – Матричний калькулятор, ввід матриці B та функціонал

Третім і останнім тренажером у нашому розгляді буде тренажер для написання SQL-запитів під назвою SQLZoo [12]. Він підійде тим студентам, які тільки починають вивчати мову. Для ознайомлення пропонується 9 розділів із завданнями з SQL. Серед тем – вкладені запити або обробка значень null. Користувач має перед собою умову завдання та блок, у який він має вписати своє рішення задачі (або відредагувати уже наявний запит) (рис. 1.4). При вводі правильної відповіді праворуч з'являється результат запиту та позначення, що студент коректно виконав завдання. Також доступні розбори рішень. У SQLZoo є довідник термінів.

Після проходження усіх завдань або на будь-якому етапі курсу, можна пройти тестування на перевірку отриманих знань. Даний залік представлений звичайним переліком питань з однією правильною відповіддю (рис. 1.5).

Introducing the world table of countries

1.

The example uses a WHERE clause to show the population of 'France'. Note that strings should be in 'single quotes';

Modify it to show the population of Germany

```
SELECT population FROM world
WHERE name = 'France'
```

Submit SQL

restore default

result

Рисунок 0.4 – Приклад одного із завдань в SQLZoo

3. Select the code which shows the countries that end in A or L

```
SELECT name FROM world
WHERE name LIKE 'a%' AND name LIKE 'l%'
```

```
SELECT name FROM world
WHERE name LIKE 'a%' OR name LIKE 'l%'
```

```
SELECT name FROM world
WHERE name LIKE '%a' AND name LIKE '%l'
```

```
SELECT name FROM world
WHERE name LIKE '%a' OR 'l%'
```

```
SELECT name FROM world
WHERE name LIKE '%a' OR name LIKE '%l'
```

Рисунок 0.5 – Одне із завдань у тестуванні SQLZoo

Підсумовуючи викладене, можна стверджувати, що розглянуті тренажери виконують завдання, які були поставлені на меті при їх розробці. Щодо їхнього дизайну, то ці тренажери є лаконічними, не перевантаженими різними відволікаючими деталями та факторами; разом з цим, вони не виглядають занадто аскетично, мають приємний вигляд для ока користувача. Завдання сформовані доступно і зрозуміло, поля для вводу значень чітко виділені, блоки для роботи на екрані розташовані візуально зручно. Звичайно, наявні деякі нюанси, які б хотілося виправити: наприклад, перший тренажер містить орфографічні помилки в інструкції, дизайн другого тренажеру не максимально привабливий для користувача. Проте, зазначені нюанси все ж таки спрямовані на покращення візуалу, а не функціоналу, тому використовувати їх в роботі доволі зручно. Розгляд вже наявних зразків допомагає виділити корисні моменти для подальшої роботи над власним віртуальним тренажером і сформулювати постановку задачі та завдань кваліфікаційної роботи.

1.4 Постановка задачі

Метою даної кваліфікаційної роботи є розроблення та програмна реалізація віртуального тренажера, який дозволить повторити і закріпити теоретичний матеріал, а також виконати практичне завдання з теми «Розв'язання задач багатовимірної оптимізації».

Серед завдань виділимо наступні:

1. Визначити функції та можливості віртуального тренажера.
2. Сформувати математичну модель двох методів розв'язання задач багатовимірної оптимізації.
3. Розробити зручний та інтуїтивно зрозумілий інтерфейс для взаємодії користувача з тренажером.
4. Перевірити якість та ефективність створеного віртуального тренажера.

2 ВИБІР МЕТОДІВ РІШЕННЯ

2.1 Вимоги до навчального об'єкту

Створення ефективного та функціонального віртуального тренажера вимагає врахування ряду важливих аспектів, пов'язаних з навчальним об'єктом. Вимоги до навчального об'єкту віртуального тренажера можуть варіюватися залежно від конкретної галузі та завдань, які ставляться перед тренажером, але деякі загальні аспекти є загальноприйнятими.

Автор спочатку розробляє сценарій, який є докладною покроковою інструкцією щодо виконання завдання, а також описує зовнішній вигляд і вміст інтерактивного практичного завдання, перед тим, як реалізувати навчальні об'єкти відповідного типу.

Інтерактивне практичне завдання має забезпечувати [13]:

1. Можливість спрощувати чи ускладнювати навчальну задачу.

Перша і найважливіша вимога – це можливість адаптувати рівень складності завдання під конкретного учня. Це не лише дозволяє кожному студентові навчатися у комфортних умовах, але й створює умови для глибокого засвоєння матеріалу.

2. Зміну параметрів, масштабу часу.

Інтерактивні практичні завдання мають бути гнучкими у плануванні часу та визначенні параметрів. Зміна масштабу часу та інших параметрів дозволяє студентам експериментувати і спостерігати за змінами в реальному часі.

3. Призупинення процесу у будь-який момент часу для обговорення ситуації, що виникла, аналізу рішень та дій того, хто навчається.

Унікальна можливість призупинення навчального процесу у будь-який момент часу гарантує, що студенти можуть активно обговорювати ситуації,

аналізувати рішення та дії, що сприяє глибшому розумінню ситуації і розвитку критичного мислення.

4. Неодноразове повторення необхідної ситуації для закріплення навичок.

Повторення є ключовим елементом навчання. Інтерактивні практичні завдання повинні надавати можливість неодноразового вирішення одних й тих самих або схожих завдань для більш ефективного закріплення отриманих навичок.

5. Постійний контроль якості дій того, хто навчається.

Моментальний контроль дозволяє студентам не лише вчитися на власних помилках, але й швидко виправляти їх. Інтерактивні практичні завдання повинні забезпечувати засоби для постійного моніторингу та відгуку.

6. Здійснення реєстрації досягнень того, хто навчається.

Здійснення реєстрації досягнень важливо для стимулювання студентів до саморозвитку. Інтерактивні практичні завдання повинні автоматично фіксувати та відзначати досягнення, надаючи студентам можливість відстежувати власний прогрес.

Для означення низки функцій, які буде виконувати користувач було створено Use Case діаграму. Use Case діаграма – це графічний інструмент для відображення функціональності системи та її взаємодії з зовнішніми елементами (користувачами, іншими системами тощо). Вона допомагає описати, як система взаємодіє з окремими компонентами (акторами) для виконання конкретних завдань або випадків використання.

В якості головних учасників процесу вказані користувач (тобто студент) та веб сервер (рис. 2.1). Для них ми визначили певні функції, серед яких задання локалізації, перехід до наступного/минулого кроку, виконання завдань, допомога викладача, завершення сесії, отримання результатів.



Рисунок 2.1 - Use Case діаграма для віртуального тренажера

Після того як ми визначили акторів та функціонал необхідно подбати про сценарій проходження тренажера. Це важливо для того, щоб вибудувати послідовність дій та скласти план для подальшого розроблення. Для цього побудуємо блок-схему. Блок-схема — це графічне зображення послідовності дій або процесу, розбитого на блоки, що представляють окремі етапи чи етапи виконання. Кожен блок містить інформацію про конкретну операцію чи етап процесу, а стрілки вказують напрямок та послідовність виконання цих дій.

На рисунку 2.2 представлена загальна схема проходження тренажера. Основним розгалуженням є вибір методу рішення задач багатовимірної оптимізації: Хука-Дживса або Нелдера-Міда. Подальші кроки є однаковими для двох опцій: перехід з кроку на крок (всього має бути три, для контролю є цикл), перевірка введених значень, рахунок використаних спроб (також три, валідується циклом). Після проходження користувач може переглянути результат.

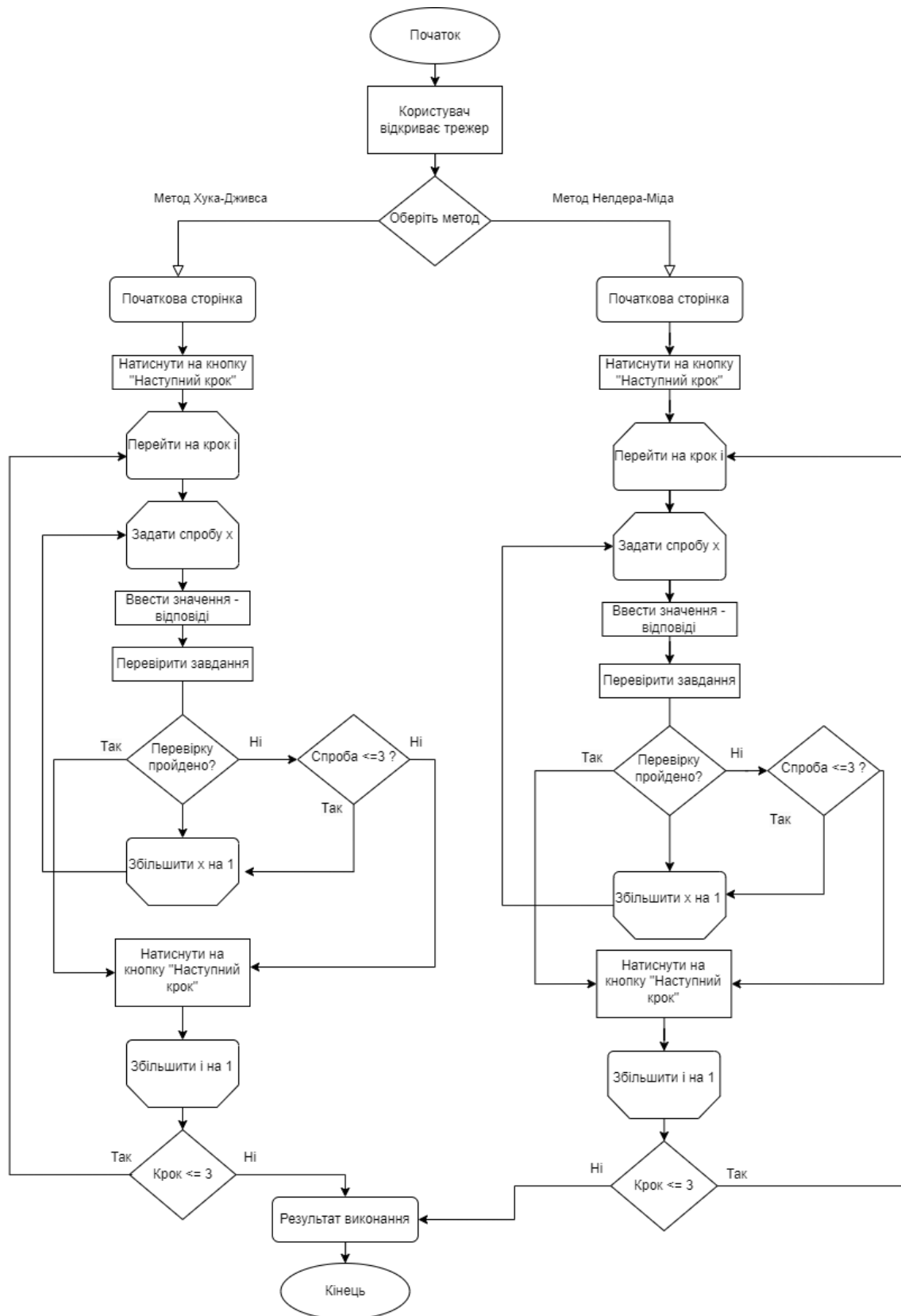


Рисунок 2.2 – Блок-схема виконання тренажера

Для визначення того, як будуть працювати додаткові функції – допомога викладача та завершення сеансу проходження створено наступну блок-схему (рис.2.3).

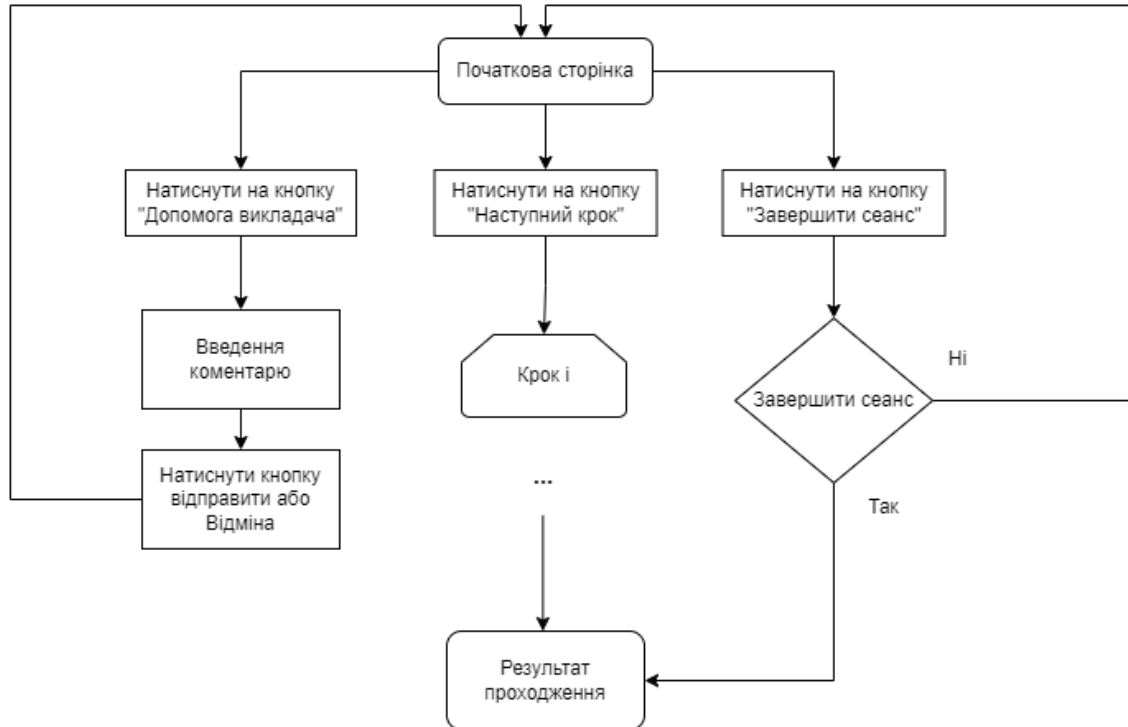


Рисунок 2.3 – Блок-схема додаткових опцій тренажера

2.2 Основні визначення та види задач

Розв’язання задач багатовимірної оптимізації є складним і важливим завданням в різних галузях, включаючи науку, освіту, техніку, економіку та інші. Цей процес спрямований на пошук найкращого рішення для задачі, яка має багато різних параметрів чи критеріїв, що піддаються оптимізації.

Щоб зрозуміти сутність розв’язання задач багатовимірної оптимізації, важливо врахувати кілька ключових етапів.

На першому етапі – формалізації задачі, необхідно чітко сформулювати цільову функцію, яку слід оптимізувати, а також визначити обмеження, які мають бути враховані під час пошуку оптимального рішення.

Вибір методу залежить від характеру задачі та сфери застосування. Деякі з популярних методів включають градієнтні методи, еволюційні алгоритми, методи часткових похідних, генетичні алгоритми та інші.

Після вибору методу оптимізації необхідно реалізувати або використовувати існуючі інструменти для моделювання задачі. Це може включати в себе використання програмних пакетів для оптимізації, розроблення власних алгоритмів та їхню імплементацію.

Після отримання результатів оптимізації, їх потрібно тестувати та аналізувати, що включає в себе перевірку на відповідність вимогам і врахування можливих побічних ефектів.

Процес оптимізації може вимагати ітераційного підходу, особливо якщо параметри системи чи умови змінюються з часом. Після завершення оптимізації важливо забезпечити підтримку та можливість адаптації рішення до нових умов [14].

Розроблений віртуальний інтерактивний тренажер буде розглядати два методи для розв'язання задач багатовимірної оптимізації: метод Хука-Дживса (або метод конфігурацій) та метод Нелдера-Міда (або метод деформуючого многогранника).

Метод Хука-Дживса є одним з популярних методів розв'язання задач оптимізації, особливо в неперервному n -вимірному просторі. Цей метод комбінує експлуатацію локального пошуку з експлорацією пошуку в широких межах. Метод Хука-Дживса також відомий як метод конфігурацій, що складається з ітераційного процесу, де в кожній ітерації виконуються такі кроки [14]:

1. Пошук за допомогою дослідження (експлорація). Починається з поточної точки, яка називається базисною точкою. В цій фазі здійснюється широкий пошук у певному околі базисної точки. Застосовуються різні напрямки та кроки, щоб перевірити, чи можна знайти кращу точку у цьому околі.

2. Пошук за допомогою зразка (експлуатація). Якщо після експлорації вдалося знайти кращу точку, то вона стає новою базисною точкою і процес повторюється. В іншому випадку, якщо краща точка не знайдена, базисна точка оновлюється у напрямку найкращої знайденої точки, і процес починається знову.

3. Завершення. Метод продовжується до досягнення певної умови зупинки, такої як досягнення максимальної кількості ітерацій або заданого значення точності.

Застосування методу Хука-Дживса є ефективним у випадках, коли функція має багато локальних оптимумів і важко знайти аналітичну формулу для пошуку глобального оптимуму.

Метод Нелдера-Мида (Nelder-Mead) є ітераційним числовим методом для знаходження мінімуму (або максимуму) функції без використання похідних. Цей метод також відомий як метод симплексів.

Основна ідея полягає в тому, щоб побудувати і поновлювати симплекс ($n+1$ вершина в n -вимірному просторі), змінюючи його розміщення та розміри. Кожен крок включає оцінку значень функції в вершинах симплексу, а потім переходить до нового симплексу на основі цих значень [14].

Початковим кроком алгоритму є ініціалізація, в якій обираються початкові точки у просторі параметрів, які утворюють симплекс. Далі проходить оцінка значень функції, які обчислюються в вершинах симплексу. Після цього проходить сортування. Вершини симплексу сортуються за значенням функції, тобто від меншого до більшого.

Наступним етапом є обчислення «центру мас» (або «центру ваги») симплексу. Обчислюється центр мас вершин, за винятком останньої (найгіршої) вершини. Потім виконується відображення. На цьому моменті обчислюється нова точка, яка є відображенням найгіршої вершини відносно центру мас.

Далі відбувається оцінка нової точки, в якій обчислюється значення функції. В залежності від отриманих значень функції в новій точці вирішується, чи розтягувати або стискати симплекс.

Потім знову проходить оцінка значень функції. Це раз обчислюються значення функції в вершинах симплексу.

Важливим моментом є поняття збіжності. Необхідно перевіряти умову збіжності або інші критерії зупинки в кінці кожної ітерації. Якщо умова виконується, алгоритм завершується; в іншому випадку повторюються кроки з пункту сортування.

Цей алгоритм ефективно працює в багатьох задачах оптимізації, зокрема там, де функція не має аналітичного визначення похідних. Однак, варто враховувати, що метод може стикатися з проблемами у випадку, коли функція має різкі зміни.

2.3 Вибір програмного інструментарію

JavaScript – це високорівнева, інтерпретована, об'єктно-орієнтована мова програмування, що широко використовується для розробки динамічних веб-сайтів. Додатково, JavaScript зараз також використовується для розробки серверної сторони (за допомогою Node.js), мобільних додатків (за допомогою фреймворків, таких як React Native), інтерфейсів користувача (за допомогою бібліотеки React або фреймворків, таких як Angular або Vue.js), а також для різноманітних інших завдань.

До переваг використання JavaScript в розробці можна віднести наступні аспекти:

- Широке використання: JavaScript широко використовується у веб-розробці для створення інтерактивних веб-сайтів та односторінкових додатків.
- Динамічність: Мова може динамічно змінювати структуру та зміст веб-сторінок без перезавантаження.
- Об'єктно-орієнтований підхід: JavaScript базується на об'єктно-орієнтованому програмуванні, що дозволяє створювати об'єкти з властивостями та методами.
- Асинхронність: Підтримка асинхронних операцій дозволяє ефективно взаємодіяти з сервером та іншими ресурсами.
- Крос-платформенність: JavaScript може використовуватися як на клієнтській, так і на серверній стороні розробки (за допомогою Node.js), що полегшує розширення функціональності від клієнтської до серверної частини.
- Переносність: Код, написаний на JavaScript, може запускатися на різних браузерах, що робить мову переносною.
- Спрощене розроблення інтерфейсів: Використання фреймворків та бібліотек, таких як React, Angular або Vue.js, дозволяє легко розробляти складні інтерфейси користувача.
- Активна спільнота: Має велику та активну спільноту розробників, що сприяє обміну знань та розвитку екосистеми інструментів.
- Можливість розширення: Можливість використання різноманітних бібліотек і фреймворків для полегшення розробки та розширення функціональності.

- Інтеграція з іншими технологіями: Легко інтегрується з іншими технологіями, такими як HTML, CSS, AJAX, що робить його невід'ємною частиною сучасного стеку веб-технологій [15].

JavaScript є ключовим елементом у сучасному веб-середовищі та допомагає забезпечувати інтерактивність та динаміку веб-застосунків.

Для відображення елементів на сторінці було використано мови розмітки та стилів HTML, CSS

HTML (HyperText Markup Language) – це мова розмітки для створення та представлення веб-сторінок. Вона визначає структуру документа за допомогою різних елементів та їх атрибутів, таких як заголовки, абзаци, зображення, посилання та інші. HTML використовується для створення основної структури веб-сайтів та надає основний контент, який браузері відображають користувачеві [16].

CSS (Cascading Style Sheets) – це мова стилізації, яка використовується для визначення зовнішнього вигляду елементів, створених за допомогою HTML. CSS включає в себе правила, які визначають вигляд тексту, кольори, розміри, відступи, розташування та інші аспекти дизайну. Використовуючи CSS, розробники можуть забезпечити стиль та макет веб-сторінки, щоб зробити її привабливою та зручною для користувачів. HTML та CSS часто використовуються разом для створення повноцінних веб-сайтів [16].

Також при розробці ми зверталися до фреймворку Bootstrap, який використовується для розробки веб-сайтів та веб-додатків. Він базується на HTML, CSS та JavaScript, і його основною метою є полегшення та прискорення процесу розробки, забезпечуючи готові компоненти та стилі, які можна легко використовувати.

Основні особливості Bootstrap:

- **Готові компоненти:** Bootstrap надає широкий набір готових компонент, таких як кнопки, форми, навігація, таблиці, каруселі, модальні вікна та багато інших. Це дозволяє розробникам швидше створювати сторінки та додатки без необхідності писати код з нуля.
- **Відзначена адаптивність:** Bootstrap має вбудовану адаптивність, що дозволяє створювати веб-сайти, які ефективно працюють на різних пристроях та екранах, включаючи мобільні пристрої.
- **Сітка:** Bootstrap використовує 12-колонкову сітку, яка полегшує розміщення та організацію вмісту на сторінці.
- **Теми та звільнені стилі:** Відображає можливість легко налаштовувати вигляд веб-сайту за допомогою тем та змінних.
- **Спільнота та підтримка:** Bootstrap користується великою та активною спільнотою розробників. Також надається документація та офіційні засоби підтримки [17].

Bootstrap став популярним інструментом у веб-розробці завдяки своїй простоті використання та можливості швидкого створення привабливих та адаптивних веб-інтерфейсів.

Для запуску та тестування додатку було використано XAMPP – безкоштовний, відкритий та легкий у використанні пакет програмного забезпечення, який містить всі необхідні компоненти для створення та розгортання веб-сайтів на локальному комп'ютері. Ім'я «XAMPP» походить від аббревіатури компонентів, які входять до складу цього пакету:

X – (Cross-platform). Підтримується на різних операційних системах, таких як Windows, Linux, Mac OS X та інші.

A – Apache. Веб-сервер Apache, який відповідає за обробку запитів веб-сторінок та їх виведення в браузері.

М – MySQL. Система управління базами даних MySQL, яка забезпечує зберігання та обробку даних на веб-сервері.

Р – PHP. Мова програмування PHP, яка використовується для розробки динамічних веб-сайтів.

Р – Perl. Включає перл, іншу мову програмування, яка також може використовуватися для розробки веб-додатків.

ХАМРР розроблений, щоб допомогти розробникам та веб-майстрам легко встановлювати та налаштовувати повне робоче середовище для розробки та тестування веб-сайтів на локальному комп'ютері перед їхнім розгортанням на живому сервері. Він є популярним інструментом для початківців та досвідчених розробників, оскільки дозволяє швидко отримати робоче середовище з необхідними компонентами [18].

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Короткий опис програмної реалізації та структура

Для розробки віртуального тренажера було використано JavaScript (для написання скриптів) та мови розмітки та стилів HTML, CSS. За основу взято шаблон JavaScript-тренажера, який використовується на платформах Міх та Lectur.ED Сумського державного університету.

У таблиці 3.1 подано назви основних папок і файлів проєкту та їх короткий опис.

Таблиця 3.1 – Короткий опис папок і файлів проєкту

Назва папки/файлу	Короткий опис
css	Папка, які містить файл стилів для поточного тренажера
fonts	Папка, ка містить набір шрифтів
img	Папка, яка містить зображення
trainer	Папка, яка містить html-сторінки для кожного кроку, скрипти та файли налаштувань
scripts	Папка, яка містить скрипти кроків (js файли)
step1_(method).js ... stepN_(method).js	Файли зі скриптами
settings	Налаштування
trainer.config.json	Файл, що містить інформацію про локалізацію, мову запуску,

	максимальну кількість відсотків за тренажер
trainer.steps.json	Файл, що містить інформацію про розподіл відсотків за виконання за кожен крок
introduction.html	Сторінка вступу, яка містить інформацію про назву тренажера, кількість кроків, можливість вибору мови.
report.html	Файл, у якому містяться налаштування для зв'язку із сервером
trainer.html	Файл для підключення усіх скриптів, файлів стилів, розмітка кнопок та інше
results.html	Файл, який відображає кількість набраних відсотків та час виконання.
step1_(method).html stepN_(method).html	... Файл із розміткою кроків

Для початку були розроблені .html файли для кроків, в яких міститься розмітка сторінки та всі елементи, які знаходяться на кожному кроці (рис. 3.1). В html файлах є посилання на масиви тексту, який міститься в файлах ua.json та en.json, для локалізації (рис. 3.2).

```

1 <div class="jumbotron page2">
2   <h1>{{ELEMENTSTEST_TITLE2}}</h1>
3
4   <h4>{{EXAMPLE_PART1}}</h4>
5   <p>{{EXAMPLE_PART2}}</p>
6
7   <h4>
8     {{LATEX1_TEXT}}
9   </h4>
10  <h4>
11    {{{STEP2_LATEX}}}
12  </h4>
13
14  <p>{{EXAMPLE_PART3}}</p>
15  <br/>
16  <h4>{{EXAMPLE_PART4}}</h4>
17
18  <p>
19    {{ELEMENTSTEST_PUT_INPUT1}}:
20    {{{STEP2_INPUT1}}}
21  </p>
22  <p>{{EXAMPLE_PART5}}</p>
23
24  <h5>
25    {{ELEMENTSTEST_PUT_INPUT2}}:
26    {{{STEP2_INPUT2}}}
27  </h5>
28  <br/>
29  <p>{{EXAMPLE_PART6}}</p>

```

Рисунок 3.6 – Розмітка для одного з кроків

```

1 {
2   "local": "Українська",
3   "Lang": {
4     "TRAINER_NAME": "Метод Нелдера-Міда",
5     "MAX_SCORE": "Кількість балів",
6     "NEXT_STEP": "Наступний крок",
7     "PREV_STEP": "Попередній крок",
8     "ENTER_TEXT": "Введіть текст",
9     "CHOOSE_LANG": "Оберіть мову",
10    "CHOOSE_SELECT": "Оберіть елемент зі списку",
11    "TEACHER_HELP": "Допомога викладача",
12    "ENTER_HELP_TEXT": "Опишіть з чим Вам потрібна допомога...",
13    "HELP_REQUEST_SEND": "Відправити",
14    "HELP_REQUEST_CLOSE": "Відмінити",
15    "CALCULATOR": "Калькулятор",
16    "ALERT_WELL_DONE": "Відмінно!",
17    "ALERT_FAIL": "Погано!",
18    "ALERT_STEP_SUCCESS": "Крок пройдений! Перейдіть до наступного.",
19    "ALERT_STEP_FAIL": "Крок провалений. Перейдіть до наступного.",
20    "STEP_COUNT": "Кількість кроків",
21    "END_TRAINER": "Завершити сеанс",
22    "END_ARE_YOU_SURE": "Ви впевнені, що хочете завершити виконання тренажера?",
23    "END_TRAINER_CLOSE": "Продовжити поточний сеанс",
24    "END_TRAINER_SEND": "Завершити поточний сеанс",
25    "CHECK": "Перевірити",
26    "ATTEMPTS_LEFT": "Ще спроб - ",
27

```

Рисунок 3.2 – Файли локалізації

Основні функції для роботи містяться в файлах .js. Основна з них це `addValidator` (вона міститься в `this.postDispatch`), яка валідує отриману відповідь від студента, перевіряючи її на правильність (рис. 3.3).

```

JS step3.js x JS step2.js
trainer > scripts > JS step3.js > step3 > postDispatch
1 var VStep3; // A variable for future validator
2 var step3 = function () {
3   var plot = null;
4   this.preDispatch = function (callback) {
5     var w = new WolframAlpha(); // Making a call to wolfram api to build a plot
6     w.setQuery('3x^2+2x+5').plot(function (data) {
7       plot = ' settings > {} trainer.config.json > ...
1  {
2    "LANGUAGES": {
3      "ua": "Українська",
4      "en": "English"
5    },
6    "DEFAULT_LANG": "ua",
7    "TRAINER_SCORE": 100,
8    "TRAINER_AUTHOR": "_",
9    "COURSE_AUTHOR": "_",
10   "PRODUCTION": false
11 }

```

Рисунок 3.4 – Файл trainer.config.json

У файлі `trainer.steps.json` (рис. 3.5) міститься інформація про кроки. Тобто про послідовність проходження, кількість балів, які може отримати користувач на цьому кроці, а також чи є для цього кроку скрипт.

```
trainer > settings > {} trainer.steps.json > ...
1  [
2    {
3      "filename"      : "introduction",
4      "score"         : 0,
5      "hasScript"    : true
6    },
7    {
8      "filename"      : "step1",
9      "score"         : 30,
10     "hasScript"     : true
11   },
12   {
13     "filename"      : "step2",
14     "score"         : 40,
15     "hasScript"    : true
16   },
17   {
18     "filename"      : "step3",
19     "score"         : 30,
20     "hasScript"    : true
21   },
22   {
23     "filename"      : "results",
24     "score"         : 0,
25     "hasScript"    : true
26   }
27 ]
```

Рисунок 3.5 – Файл `trainer.steps.json`

3.2 Інструкція роботи з тренажером

На самому першому етапі перед користувачем з'являється початкова сторінка. На вибір студенту дається можливість обрати з яким методом буде проводитися подальша робота: метод Хука-Дживса або метод Нелдера-Міда (рис.3.6).



Рисунок 3.6 – Початковий етап тренажеру – вибір методу

Якщо ми підвантажимо даний тренажер на LecturED, то спочатку користувачу буде необхідно ввести своє ім'я та натиснути на кнопку «Далі». Таким чином, викладач зможе вести звітність і простежувати результати студентів та контролювати подолання завдання.

Загалом структура проходження кожного методу є однаковою: спочатку першим кроком є повторення теоретичного матеріалу, а потім виконання практичного завдання. Для початку розглянемо роботу з методом Хука-Дживса.

На першій сторінці (див. рис. 3.7) вступу користувачу доступна можливість зміни локалізації. Для цього слід натиснути на прапорець – український чи англійський. Також доступні чотири кнопки, які будуть незмінними впродовж усіх кроків. «Попередній крок» та «Наступний крок» виконують відповідно функцію направлення на певний етап тренажеру. При натисканні на «Допомога викладача» перед користувачем з'являється віконце, в якому студент має можливість описати проблему, з якою він стикнувся і відправити викладачу (рис. 3.8).

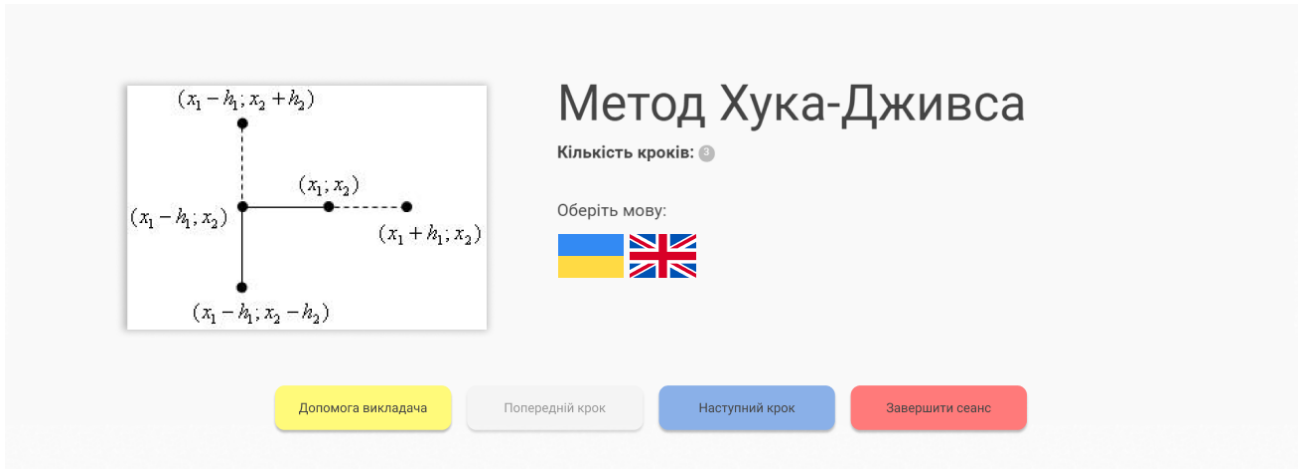


Рисунок 3.7 – Початкова сторінка методу Хука-Дживса

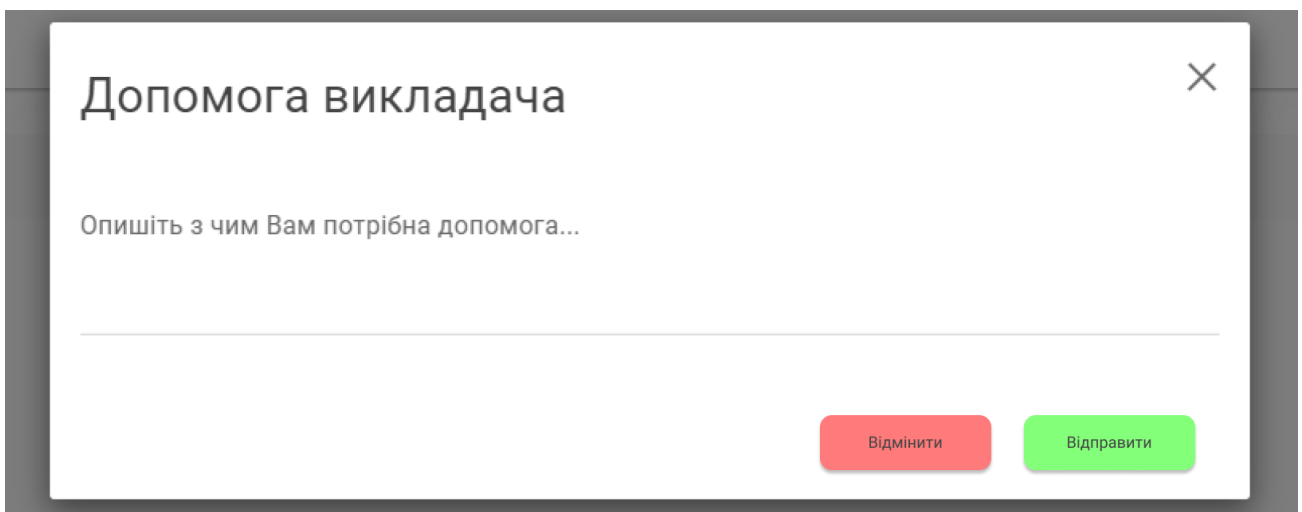


Рисунок 3.8 – Віконце з опцією допомоги викладача

При натисканні кнопки «Завершити сеанс» перед користувачем з'являється віконце, в якому треба підтвердити завершення проходження тренажера. На вибір дається дві опції: продовжити сеанс або завершити його (рис. 3.9). Перший варіант необхідний для того, якщо студент випадково натиснув на кнопку, або вирішив продовжити проходження. Відповідно після натискання на кнопку «Завершити поточний сеанс» студент втрачає можливість продовжити, або ж він вже пройшов усі кроки і бажає завершити проходження.

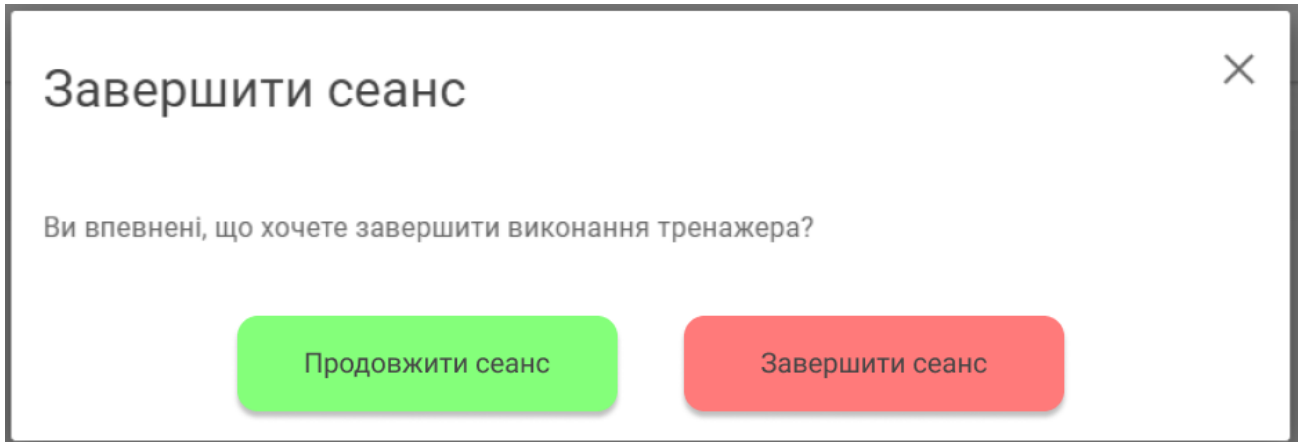


Рисунок 3.9 – Віконце з опцією завершення сеансу

Для проходження на наступний крок натискаємо відповідну кнопку. На даному етапі ми бачимо коротке повторення теоретичного матеріалу – основних понять та визначень (рис. 3.10). Є короткі теоретичні відомості та декілька тестів. Перший – перетягування правильних відповідей. Другий – обрати правильний елемент зі списку. Третій – обрати правильний варіант з перелічених. Четвертий – ввести правильне ім'я координати (рис. 3.11).

Повторення теоретичного матеріалу

Метод Хука – Дживса або пошук за зразком призначений для пошуку безумовного локального екстремуму функції і відноситься до прямих методів, тобто спирається безпосередньо на значення функції. Алгоритм складається з двох фаз: досліджуючий пошук і пошук за зразком.

Досліджувальний пошук орієнтований на виявлення характеру локального поведіння цільової функції та визначення напрямку подальшого дослідження.

Пошук за зразком полягає в реалізації єдиного кроку з одержаної базової точки уздовж прямої, що з'єднує цю точку з попередньою базовою точкою.

Для виконання алгоритму, на першому кроці необхідно обрати:

Початкова точка Вектор Крок Базис

Досліджувальний пошук починається у: Оберіть елемент зі списку ▾

Рисунок 3.10 – Крок 1 (1)

В результаті досліджувального пошуку одержують:

- Початковий крок
- Базову точку
- Початкову точку
- Базовий вектор

Пошук за зразком. Пошук за зразком полягає в реалізації єдиного кроку з одержаної базової точки уздовж прямої, що з'єднує цю точку з попередньою базовою точкою. Нова точка зразка визначається відповідно до формули

$$P_i = b_i + 2(b_{i+1} - b_i)$$

Як тільки рух за зразком не приводить до зменшення цільової функції, точка P_i фіксується як тимчасова базова точка і знову проводиться досліджувальний пошук.

Якщо в результаті виходить точка з меншим значенням цільової функції, ніж у точці (ведіть назву точки у форматі $x(0)$, де 0 - це індекс точки):

то вона розглядається як нова базова точка

З іншого боку, якщо досліджувальний пошук виявиться невдалим, необхідно повернутися в точку $b_i + 1$ і провести досліджувальний пошук для виявлення нового напрямку мінімізації. У результаті виникає ситуація, де такий пошук не приводить до успіху. У цьому разі потрібно зменшити величину кроку шляхом введення деякого множника і відновити досліджувальний пошук. Пошук завершується, якщо величина кроку стає досить малою.

Рисунок 3.11 – Крок 1 (2)

Після вводу відповідей необхідно натиснути на кнопку «Перевірити», яка провалідує усі поля та визначить правильні та неправильні відповіді. Перейти на наступний крок можна виконавши одну з двох умов. Перша – ввести усі відповіді правильно, друга – використати усі спроби, яких усього на крок дається три (цей параметр при бажанні викладача можна змінити).

Переходимо до другого кроку (рис. 3.12). Нам пропонується виконати практичне завдання, а саме першу ітерацію. На другому кроці ми виконуємо перший етап ітерації 1 – досліджувальний пошук. На початку нам наведена цільова функція та початкова точка. Далі необхідно ввести координати початкової точки та значення цільової функції в ній.

Потім починаємо виконувати досліджувальний пошук, змінюючи по черзі кожен координату додаванням довжини кроку до них (рис. 3.13). Ці значення, а також значення цільової функції в цих координатах, необхідно обрахувати.

Досліджувальний пошук

Виконаємо тренувальне завдання.

Знайти мінімум функції

$$f(x) = (x_1 - 2)^2 + (x_2 - 5)^2 + (x_3 + 2)^4$$

з точністю до $\epsilon = 0,1$. Використати як початкову точку $(4, -2, 3)$. Взяти початковий крок $h = 1$.

Виконаємо першу ітерацію

1. В якості початкової точки оберемо точку з координатами (введіть координати для x у z через пробіл, наприклад 1 -1 1):

Введіть текст

Обчислимо значення функції в початковій точці $b1$:

$f(b1) = :$ Введіть текст

Рисунок 3.12 – Крок 2 (1)

2. Виконаємо досліджувальний пошук, змінюючи по черзі кожну змінну додаванням довжини кроку. Обчислимо відповідні значення функції (спочатку збільшене на крок значення, потім зменшене).

$f($ Введіть текст $, -2, 3) =$ Введіть текст

$f($ Введіть текст $, -2, 3) =$ Введіть текст

Продовжимо пошук від нової точки $b1^*$, в якій значення функції менше, ніж в початковій точці. Значення координат x та u впишіть через пробіл (спочатку збільшене на крок значення, потім зменшене).

$f($ Введіть текст $, 3) =$ Введіть текст

$f($ Введіть текст $, 3) =$ Введіть текст

Продовжимо пошук від нової точки $b1^*$, в якій значення функції менше, ніж в початковій точці. Значення координат x , u та z впишіть через пробіл (спочатку збільшене на крок значення, потім зменшене).

$f($ Введіть текст $, 3) =$ Введіть текст

$f($ Введіть текст $, 3) =$ Введіть текст

Одержимо точку $b2($ Введіть текст $)$, в якій було досягнуто найменшого значення функції на даному кроці.

Рисунок 3.13 – Крок 2 (2)

Після того як ми виконали завдання або використали всі спроби, натискаємо на кнопку «Наступний крок» та переходимо до третього кроку. На цьому кроці ми виконуємо другий етап першої ітерації – пошук за зразком (**Error! Reference source not found.**). На цьому моменті нам необхідно визначити координати точки через пробіл та обрахувати значення цільової функції в цій точці.



Рисунок 3.14 – Крок 3

Після того як ми виконали завдання або використали всі спроби, натискаємо на кнопку «Наступний крок» або на кнопку «Завершити сеанс» та переходимо до кроку з результатами (Рисунок).

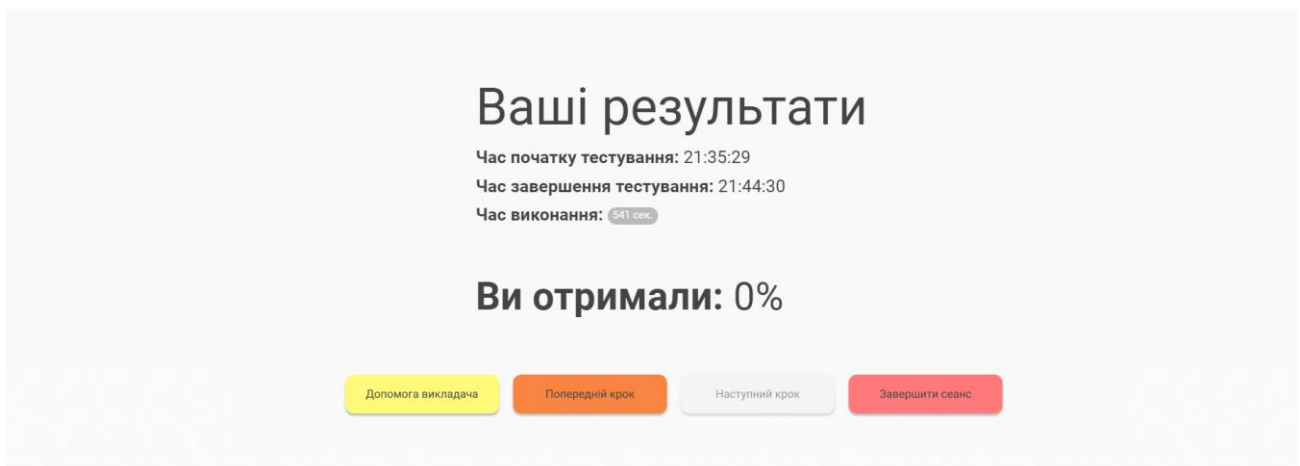


Рисунок 3.15 – Сторінка з результатами

Наступний метод до розгляду – метод Нелдера-Міда (рис. 3.16). Перед користувачем постає віконце для початку роботи, кнопки є такими ж самими як і

у минулому методі. Структура роботи з цим методом аналогічна до минулого. Першим кроком є повторення теоретичного матеріалу (рис. 3.17 та рис. 3.18).

Рисунок 3.16 – Початкова сторінка методу Нелдера-Міда

Повторення теоретичного матеріалу

Метод Нелдера-Міда, також відомий як метод симплексу чи метод деформації, є числовим методом для оптимізації функцій без обмежень.

Основна ідея методу Нелдера-Міда полягає в тому, що на кожному кроці алгоритм використовує симплекс (геометричну фігуру у вигляді трикутника в двовимірному просторі, тетраедра в тривимірному та т.д.), щоб оцінити значення функції в різних точках простору параметрів. Після цього виконуються операції зменшення, розширення або згортання симплексу залежно від результатів оцінок.

Розглянемо кроки алгоритму

1. Знайдемо значення функції $f_1 = f(x_1)$, $f_2 = f(x_2)$, ..., $f_{n+1} = f(x_{n+1})$ у вершинах симплекса.

Початковий симплекс (координати його вершин) можна вибрати довільно. Приращення δ_1 та δ_2 залежать лише від n та вибраного масштабного множника h і визначаються за формулами.

2. Серед значень функцій f_i необхідно знайти:

Найбільше значення f_h та відповідну до неї точку x_h

Найменше значення функції f_l та відповідну до неї точку x_l

Наступне за найбільшим значенням функції f_g та відповідну до неї точку x_g

Наступне за найменшим значенням f_m та відповідну до неї точку x_m

Рисунок 3.17 – Крок 1 (1)

3. Знайдемо центр тяжіння всіх точок і обчислимо $f(x_0) = f_0$.

4. Відбивши точку x_h відносно точки x_0 , одержимо нову точку x_g і знайдемо $f_g = f(x_g)$.

Якщо $\alpha > 0$ – це коефіцієнт відбиття, то положення x_g визначається таким способом::

$x_g = \alpha(x_0 - x_h)$

$x_g = (1 - \alpha)x_0 - \alpha x_h$

$x_g = \alpha(x_h)$

$x_g = (1 + \alpha)x_0 - \alpha x_h$

5. Далі проходить порівняння значень функцій в різних точках.

Для початку порівняємо значення : Оберіть елемент зі списку ▾

Знаходимо x_e , f_e і порівнюємо $f_e < f$ при умові якщо: Оберіть елемент зі списку ▾

x_g є кращою точкою порівняно з іншими двома точками симплекса і ми замінюємо точку x_h на точку x_g і, якщо збіжність не досягнута, повертаємося на крок 2 при умові: Оберіть елемент зі списку ▾

6. Порівнюємо значення функцій f_g і f_h .

6. Порівняємо значення функцій f_g і f_h . При будь-яких варіантах, заміні значень та інше, проводиться процедура:

7. Порівнюємо значення функцій f_g і f_h .

8. Перевірка збіжності.

Рисунок 3.18 – Крок 1 (2)

На цій сторінці користувач має можливість оновити свої знання щодо заданої теми, а також потренуватися на тестах, серед яких перетягування відповіді, вибір одного варіанту та введення своєї відповіді. Після проходження кроку, або ж провалу студент переходить на другий етап (рис. 3.19 та рис. 3.20)

Виконання першої ітерації

Умова завдання

Знайти мінімум функції $f(x)$ із точністю до $\epsilon = 0,01$. Використати як початкову точку $(0, 0)$. Взяти початковий крок $h = 2$.

Використати значення $\alpha = 1$, $\beta = 0,5$, $\gamma = 2$.

$$f(x) = (1 + x_1)^2 + (2 - x_2)^2$$

Побудуємо початковий симплекс (введіть значення до 4 цифр після коми, у форматі 1.1111)

$$\delta_1 = \left(\frac{\sqrt{n+1} + n - 1}{n\sqrt{2}} \right) h = \text{Введіть текст}$$

$$\delta_1 = \left(\frac{\sqrt{n+1} - 1}{n\sqrt{2}} \right) h = \text{Введіть текст}$$

Рисунок 3.19 – Крок 2 (1)

Маємо $f_r > f_l$, проте $f_r < f_g$, тобто x_g є кращою точкою порівняно з двома іншими точками, і ми замінюємо точку x_h на точку x_g

Перевіряємо збіжність за формулою

$$\bar{f} = \sum_{i=1}^n f_i / (n + 1) = \text{Введіть текст} \quad \sigma^2 = \sum_{i=1}^{n+1} (f_i - \bar{f})^2 / (n + 1) \quad \text{Введіть текст}$$

Оскільки збіжність не досягнуто, починаємо другу ітерацію

Рисунок 3.20 – Крок 2 (2)

Другий крок представляє собою проходження одного практичного завдання. Так як рішення задач багатовимірної оптимізації може бути як багатоетапним, так і маленьким, для другого кроку ми обрали рішення першої ітерації. Для студента подані формули і коротка умова. Основним завданням є введення значень, які користувач розраховує самостійно. Після перевірки введення або провалу кроку студент переходить на третій та останній етап (рис. 3.21).

Виконання останньої ітерації

Давайте виконаємо останню ітерацію для того, щоб зрозуміти яким чином пошук мінімуму функції завершується.

Маємо такі значення для точок, розрахуйте значення функції в них:

$x_1(1.1676; 2.1620); f_1 =$

$x_2(0.9685; 1.7584); f_2 =$

$x_3(0.5176; 1.9318); f_3 =$

Виконавши операцію відбиття, одержимо точку

$x_4(1.6185; 1.9886); f_4 =$

Оскільки $f_r > f_l$ та $f_r > f_g$, переходимо на кроку стискання

$x_5(1.7077; 1.3541); f_5 =$

Перевірка збіжності $\zeta = 0,0097 < \epsilon$, збіжність досягнуто.

Отже, мінімум функції (з точністю до заданого значення ϵ) має місце в точці (для x ввести два значення координат):

$x_{min} =$

$f_{min} =$

Рисунок 3.21 – Крок 3

Так як вирішення даного прикладу є багатоетапним, то було вирішено на останньому кроці розв'язати останню ітерацію, для того щоб зрозуміти, коли алгоритм проходження закінчується – тобто коли збіжність досягнута. На

рисунок продемонстровано, що дані з минулих ітерацій присутні, користувачу лиш необхідно обрахувати значення функції в заданих точках.

Після переходу на наступний крок студент бачить свій результат (рис. 3.22) – аналогічно до минулого прикладу.

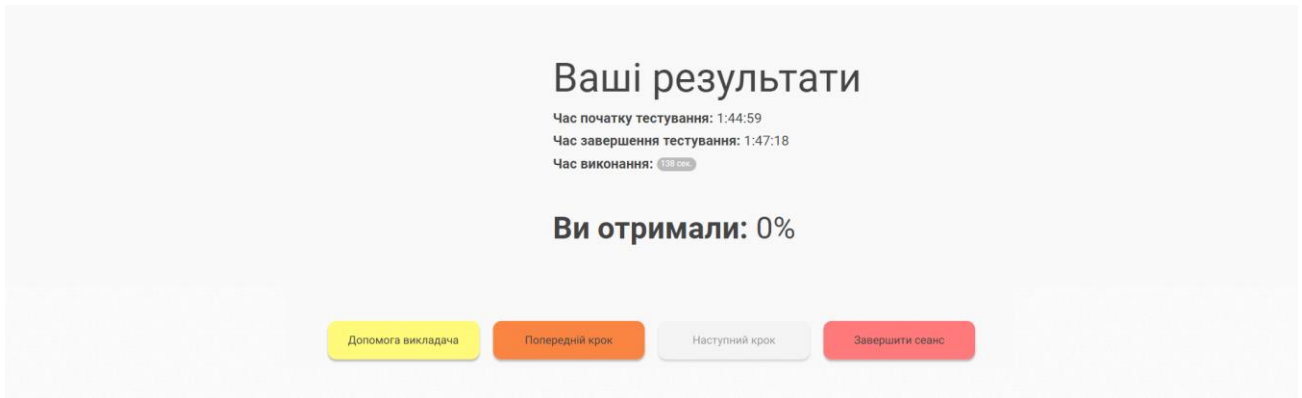


Рисунок 3.22 – Крок 3

3.3 Приклад тестування та результати роботи програми

Останнім етапом розробки є тестування виконаного тренажера та демонстрація результатів роботи програми. На початку оберемо українську локалізацію (рис. 3.23).

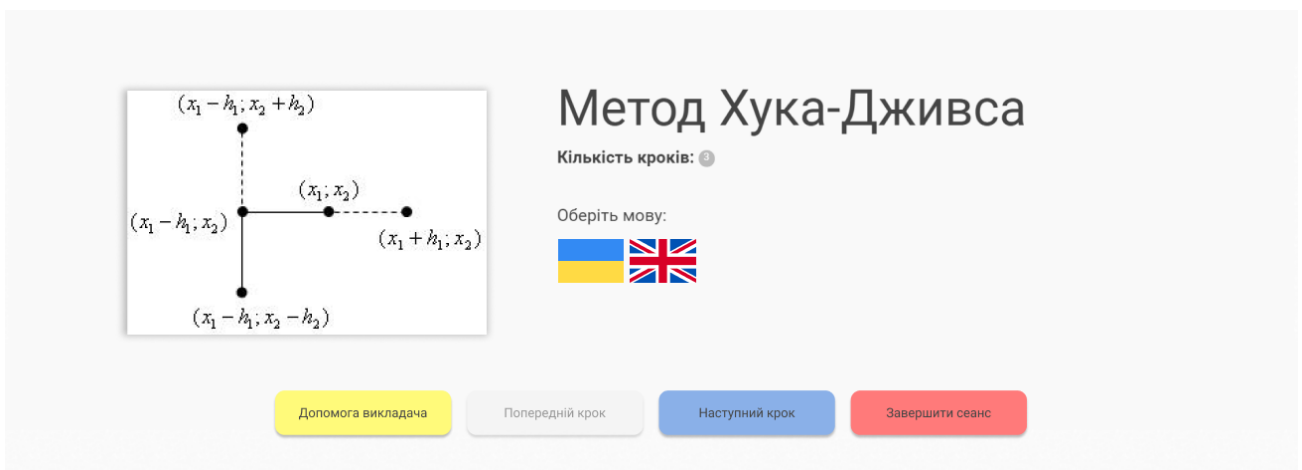


Рисунок 3.23 – Крок 1 (а) – перевірка валідації відповідей

Натискаємо на кнопку «Наступний крок». На першому кроці спробуємо обрати правильні і неправильні відповіді. Натискаємо на кнопку «Перевірити» і бачимо, що відповіді підсвічуються зеленим і червоним відповідно до їх правильності (**Error! Reference source not found.** 3.24, рис. 3.25). Через те, що маємо невірні відповіді у нас немає можливості перейти на наступний крок. Про це свідчить напис на кнопці – у нас залишилось 2 спроби.

Для виконання алгоритму, на першому кроці необхідно обрати:

Початкова точка Вектор Крок Базис

Досліджувальний пошук починається у: початковий (вихідний) точці

В результаті досліджувального пошуку одержують:

- Початкову точку
- Базовий вектор
- Базову точку
- Початковий крок

Рисунок 3.24 – Крок 1 (а) – перевірка валідації відповідей

Якщо в результаті виходить точка з меншим значенням цільової функції, ніж у точці (ведіть назву точки у форматі $x(0)$, де 0 - це індекс точки): $b(i+1)$ то вона розглядається як нова базова точка

З іншого боку, якщо досліджувальний пошук виявиться невдалим, необхідно повернутися в точку $b_i + 1$ і провести досліджувальний пошук для виявлення нового напрямку мінімізації. У результаті виникає ситуація, де такий пошук не приводить до успіху. У цьому разі потрібно зменшити величину кроку шляхом введення деякого множника і відновити досліджувальний пошук. Пошук завершується, якщо величина кроку стає досить малою.

Перевірити (Ще спроб - 2)

Рисунок 3.25 – Крок 1 (б) – перевірка валідації відповідей

Після того, як ми спробували кілька спроб, щоб дати правильні відповіді, але нас спіткала невдача, ми отримали повідомлення про те, що крок провалений (**Error! Reference source not found.**). Проте, за правильні відповіді ми все одне отримали бали.

Пошук за зразком. Пошук за зразком полягає в реалізації єдиного кроку з одержаної базової точки уздовж прямої, що з'єднує цю точку з попередньою базовою точкою. Нова точка зразка визначається відповідно до формули

$$P_i = b_i + 2(b_{i+1} - b_i)$$

Як тільки рух за зразком не приводи

Крок провалений! Перейдіть до наступного

знову проводиться досліджувальний пошук.

Якщо в результаті виходить точка з меншим значенням цільової функції, ніж у точці (ведіть назву точки у форматі x{0}, де 0 - це індекс точки):

b{i+1} то вона розглядається як нова базова точка

З іншого боку, якщо досліджувальний пошук виявиться невдалим, необхідно повернутися в точку $b_i + 1$ і провести досліджувальний пошук для виявлення нового напрямку мінімізації. У результаті виникає ситуація, де такий пошук не приводить до успіху. У цьому разі потрібно зменшити величину кроку шляхом введення деякого множника і відновити досліджувальний пошук. Пошук завершується, якщо величина кроку стає досить малою.

Перевірити (Ще спроб - 0)

Рисунок 3.26 – Повідомлення про невдале проходження першого кроку

Переходимо до другого кроку та спробуємо дати усі правильні відповіді. Спостерігаємо, що у вікні з'явилося повідомлення, що крок пройдено (рис. 3.27) і кнопка «Наступний крок» стала доступною для натискання.

f (5 , -2, 3) = 683

f (3 , -2, 3) = 675

Продовжимо пошук від нової точки b_1^* , в якій значення функції менше, ніж в початковій точці. Значення координат x та y впишіть через пробіл (спочатку збільшене на крок значення, потім зменшене).

f (3-1 , 3) = 662

f (3-3 , 3) = 69 Крок пройдений! Перейдіть до наступного

Продовжимо пошук від нової точки b_1^* , в якій значення функції менше, ніж в початковій точці. Значення координат x, y та z впишіть через пробіл (спочатку збільшене на крок значення, потім зменшене).

f (3-1-4 , 3) = 1333

f (3-1-2 , 3) = 293

Одержимо точку b_2 (3-1-2), в якій було досягнуто найменшого значення функції на даному кроці.

Перевірити (Ще спроб - 2)

Рисунок 3.27 – Повідомлення про невдале проходження першого кроку

Переходимо до останнього кроку і також вводимо правильні відповіді. Тепер настала черга перевірити свій результат. Як бачимо (Рисунок), через те, що ми вводили як правильні, так і неправильні відповіді, то тренажер ми склали на 84%. Окрім того, можемо передивитися за який час ми пройшли тренажер.

Ваші результати

Час початку тестування: 21:8:26
Час завершення тестування: 21:10:44
Час виконання: 138 сек.

Ви отримали: 84%

Допомога викладача | Попередній крок | Наступний крок | Завершити сеанс

Рисунок 3.28 – Сторінка з результатами

ВИСНОВКИ

Метою даної кваліфікаційної роботи було визначено розроблення та дослідження інноваційного віртуального тренажера, призначеного для розв'язання складних задач багатовимірної оптимізації. В ході роботи було поставлено низку завдань, які були успішно виконані.

На початку дослідження нами був проведений аналіз предметної галузі, а саме було визначено, що інструменти, засновані на використанні цифрових технологій та інтерактивних платформ, відкривають нові можливості для залучення студентів до активної участі у процесі навчання та самоосвіти. Зауважено, що одним із найважливіших і перспективних напрямків у цьому контексті є використання віртуальних тренажерів. Зроблено огляд переваг і перспектив використання таких інструментів в освітньому процесі. Акцентовано, що їх ефективність полягає в здатності сприяти активній взаємодії між викладачами і студентами, розвивати критичне мислення учасників освітнього процесу, надавати перевагу практичному вивченню навичок через імітацію реальних ситуацій.

Проведено аналіз вже існуючих онлайн аналогів з метою визначення функцій та можливостей віртуального тренажера. Сформульовано вимоги до розробленого продукту, його цільовий функціонал та сфера застосування.

На другому етапі розроблено математична модель двох методів розв'язання задач багатовимірної оптимізації. Отримані моделі дозволили глибше розібратися в принципах роботи різних методів та визначити їхні переваги та обмеження.

В межах третього завдання розроблено інтуїтивно зрозумілий і зручний інтерфейс для взаємодії користувача з віртуальним тренажером. Нами був

створений інтерфейс, який спростив уведення вхідних даних, представлення результатів та взаємодію з різними функціональними можливостями тренажера.

В рамках третього завдання було оцінено якість та ефективність розробленого віртуального тренажера. Здійснено тестування та експериментальне підтвердження високого рівня функціональності та ефективності тренажера; отримано результати щодо підтримки для використання розробленого інструменту в різних сферах.

У результаті виконання поставлених завдань відзначено успішне досягнення поставлених цілей та створення високоефективного віртуального тренажера, який відповідає актуальним вимогам і може знайти практичне застосування у сфері розв'язання задач багатовимірної оптимізації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Rebecca M. Logan, Cynthia E. Johnson, Jeremy W. Worsham. Development of an e-learning module to facilitate student learning and outcomes. *Teaching and Learning in Nursing*. 2021. Volume 16, Issue 2. P. 139–142.
2. Ignatius Adrian Mastan, Dana Indra Sensuse, Ryan Randy Suryono, Kautsarina. Evaluation of distance learning system (e-learning): a systematic literature review. *Jurnal TEKNOINFO*. 2022. Vol. 16, No. 1. P. 132–137.
3. Баценко С. В. Імерсивні технології: теоретичний аспект. *Імерсивні технології в освіті* : зб. матеріалів учасн. І наук.-практ. конф. з міжнар. участю: Київ, 2021. С. 36–39.
4. Литвонова С. Г. Імерсивне середовище навчання: порівняльний аналіз. *Науковий вісник Ужгородського університету*. Серія: «Педагогіка. Соціальна робота». 2023. № 2(53). С. 76–82.
5. Соколюк О. М. Імерсивність в сучасних освітніх середовищах. *Імерсивні технології в освіті* : зб. матеріалів учасн. І наук.-практ. конф. з міжн. участю: Київ, 2021. С. 143–148.
6. Купченко О. В., Черненко О. О. Створення тренажера з теми «Коди із виправленням помилок» дистанційного навчального курсу «Теорія інформації і кодування». *Збірник наукових статей магістрів* : Полтава, 2021. С. 106–110.
7. Martin Ebner, Sandra Schön, Clarissa Braun, Markus Ebner, Ypatios Grigoriadis, Maria Haas, Philipp Leitner and Behnam Taraghi. COVID-19 Epidemic as E-Learning Boost? Chronological Development and Effects at an Austrian University against the Background of the Concept of “E-Learning Readiness”. *Future Internet*. 2020. №12. 94.
7. Прилипко С. М., Іванов С. М. Сучасні інноваційні підходи в навчальному процесі. *Право та інноваційне суспільство* : електрон. наук. вид.

2013 р., №1. С. 89–106. URL: http://nbuv.gov.ua/UJRN/pric_2013_1_14 (дата звернення 20.11.2023).

8. Chen L., Chen P., Lin Z. Artificial Intelligence in Education: A Review. *IEEE Access*. 2020. Volume 8. P. 75264–75278. URL: https://openresearch.amsterdam/image/2021/8/11/artificial_intelligence_in_education_a_review.pdf (дата звернення: 20.11.2023).

9. Вплив платформ для онлайн-курсів на навчання в школах: веб-сайт. URL: <https://osvita.ua/news/89291/> (дата звернення: 28.11.2023).

10. Інтерактивний математичний тренажер для формування навичок обчислення визначеного інтеграла: веб-сайт. URL: <https://learningapps.org/13872489> (дата звернення: 28.11.2023).

11. Розв'язування задач з матрицями: веб-сайт. URL: https://ua.onlinemschool.com/math/assistance/matrix/#google_vignette (дата звернення: 28.11.2023).

12. SQLZoo: SQL Tutorial: веб-сайт. URL: https://sqlzoo.net/wiki/SQL_Tutorial (дата звернення: 28.11.2023).

13. Вимоги до навчально-методичних матеріалів дистанційної форми навчання та критерії їх оцінювання [Електронний ресурс] // Реєстр нормативної бази Сумського державного університету. – Режим доступу : <https://normative.sumdu.edu.ua/?task=getfile&tmpl=component&id=ee2d07c5-f464-e411-afcd-001a4be6d04a&kind=1>.

14. Нефьодов Ю. М., Балицька Т. Ю. Методи оптимізації в прикладах та задачах. Київ: Кондор, 2011. С. 199–205.

15. Advantages of JavaScript by Daragh Ó Tuama. URL: <https://codeinstitute.net/global/blog/advantages-of-javascript/> (дата звернення: 22.11.2023).

16. What are HTML and CSS used for? The basics of coding for the web. URL: <https://www.futurelearn.com/info/blog/what-are-html-css-basics-of-coding> (дата звернення: 22.11.2023).

17. What Is Bootstrap & Advantages Of Bootstrap In Web Development. URL: <https://www.cybersuccess.biz/advantages-of-bootstrap/>(дата звернення: 22.11.2023).

18. What is XAMPP? URL: <https://www.educba.com/what-is-xampp/>(дата звернення: 22.11.2023).

19. Alves Claudia, Bach John. Javascript For Beginners: The Ultimate Guide to Understand JavaScript Code and Its Fundamentals. 2nd edition. Independently published, 2020. 1189 p.

20. Du Preez O. J. JavaScript for Gurus: Use JavaScript programming features, techniques and modules to solve everyday problems. New Delhi: BPB Publication, 2020. 126 p.

ДОДАТОК А

introduction_Hook_Jeeves.html

```

<div class="row jumbotron">
  <div class="col-md-5 animated zoomIn">
    
  </div>
  <div class="col-md-7 animated fadeInDown">
    <h1>{{TRAINER_NAME}}</h1>
    <p>
      <h5><strong>{{STEP_COUNT}}:</strong><span
class="badge">{{STEPS_COUNT}}</span></h5>
    <br/>
    <h5>{{CHOOSE_LANG}}:</h5>
    <div class="flag-icon flag-icon-ua" title="ua" id="ua"></div>
    <div class="flag-icon flag-icon-en" title="en" id="en"></div>
  </p>
</div>
</div>

```

step1_Hook_Jeeves.html

```

<div class="jumbotron page1">
  <h1>{{ELEMENTSTEST_TITLE1}}</h1>
  <p>{{THEORY_LINE1}}</p>
  <p>{{THEORY_LINE2}}</p>
  <p>{{THEORY_LINE3}}</p>

  <h5>
    {{ELEMENTSTEST_PUT_DROPPABLE}}:

```

```

        {{{STEP1_DROPPABLE}}}
        {{{STEP1_DRAGGABLE}}}
    </h5>
<br/>
<h5>
    {{ELEMENTSTEST_CHOOSE_WRITE_OPTION}}:
    {{{STEP1_SELECT}}}
</h5>
<br/>
<h5>
    {{ELEMENTSTEST_RADIOS}}:
    <br/>
</h5>
<h5>
    {{{STEP1_1_RADIOS}}}
</h5>
<br/>
<p>{{THEORY_LINE4}}</p>
<h4>
    {{LATEX1_TEXT}}
</h4>
<h4>
    {{{STEP1_LATEX}}}
</h4>
<p>{{THEORY_LINE5}}</p>
<h5>
    {{ELEMENTSTEST_ENTER_TEST}}:
    {{{STEP1_TEXTINPUT}}}
    {{ELEMENTSTEST_ENTER_TEST1}}
</h5>
<p>{{THEORY_LINE6}}</p>

```

```

    <div style="text-align: center;"><button class="btn btn-success
check">{{CHECK}}</button></div>
</div>

```

step2_Hook_Jeeves.html

```

<div class="jumbotron page2">
  <h1>{{ELEMENTSTEST_TITLE2}}</h1>
  <h4>{{EXAMPLE_PART1}}</h4>
  <p>{{EXAMPLE_PART2}}</p>
  <h4>
    {{LATEX1_TEXT}}
  </h4>
  <h4>
    {{{STEP2_LATEX}}}
  </h4>

  <p>{{EXAMPLE_PART3}}</p>
  <br/>
  <h4>{{EXAMPLE_PART4}}</h4>
  <p>
    {{ELEMENTSTEST_PUT_INPUT1}}:
    {{{STEP2_INPUT1}}}
  </p>
  <p>{{EXAMPLE_PART5}}</p>
  <h5>
    {{ELEMENTSTEST_PUT_INPUT2}}:
    {{{STEP2_INPUT2}}}
  </h5>
  <br/>
  <p>{{EXAMPLE_PART6}}</p>
  <h5>

```

```
        {{ELEMENTSTEST_PUT_INPUT3_1}}
        {{{STEP2_INPUT3}}}
        {{ELEMENTSTEST_PUT_INPUT3_2}}
        {{{STEP2_INPUT4}}}
</h5>
<h5>
        {{ELEMENTSTEST_PUT_INPUT3_1}}
        {{{STEP2_INPUT5}}}
        {{ELEMENTSTEST_PUT_INPUT3_2}}
        {{{STEP2_INPUT6}}}
</h5>
<p>{{EXAMPLE_PART7}}</p>
<h5>
        {{ELEMENTSTEST_PUT_INPUT3_1}}
        {{{STEP2_INPUT7}}}
        {{ELEMENTSTEST_PUT_INPUT3_4}}
        {{{STEP2_INPUT8}}}
</h5>
<h5>
        {{ELEMENTSTEST_PUT_INPUT3_1}}
        {{{STEP2_INPUT9}}}
        {{ELEMENTSTEST_PUT_INPUT3_4}}
        {{{STEP2_INPUT10}}}
</h5>
<p>{{EXAMPLE_PART8}}</p>
<h5>
        {{ELEMENTSTEST_PUT_INPUT3_1}}
        {{{STEP2_INPUT11}}}
        {{ELEMENTSTEST_PUT_INPUT3_4}}
        {{{STEP2_INPUT12}}}
</h5>
```

```

<h5>
    {{ELEMENTSTEST_PUT_INPUT3_1}}
    {{{STEP2_INPUT13}}}
    {{ELEMENTSTEST_PUT_INPUT3_4}}
    {{{STEP2_INPUT14}}}
</h5>
<h5>
    {{ELEMENTSTEST_PUT_INPUT4_1}}
    {{{STEP2_INPUT15}}}
    {{ELEMENTSTEST_PUT_INPUT4_2}}
</h5>
<div style="text-align: center;"><button class="btn btn-success
check">{{CHECK}}</button></div>
</div>

```

step3_Hook_Jeeves.html

```

<div class="jumbotron page1">
    <h1>{{ELEMENTSTEST_TITLE3}}</h1>
    <p>{{STEP3_TEXT1}}</p>

    <h4>
        {{LATEX1_TEXT}}
    </h4>
    <h4>
        {{{STEP3_LATEX}}}
    </h4>
    <p>{{STEP3_TEXT2}}</p>
    <h5>
        {{ELEMENTSTEST_PUT_INPUT5}}
        {{{STEP3_INPUT1}}}
        {{ELEMENTSTEST_PUT_INPUT3_3}}

```

```

</h5>
<p>{{STEP3_TEXT3}}</p>
<h5>
    {{ELEMENTSTEST_PUT_INPUT6_1}}
    {{{STEP3_INPUT2}}}
    {{ELEMENTSTEST_PUT_INPUT6_2}}
</h5>
<p>{{STEP3_TEXT4}}</p>
    <div style="text-align: center;"><button class="btn btn-success
check">{{CHECK}}</button></div>
</div>

```

step1_Nelder-Mead.html

```

<div class="jumbotron page1">

    <h1>{{ELEMENTSTEST_TITLE1}}</h1>
    <p>{{THEORY_LINE1}}</p>
    <p>{{THEORY_LINE2}}</p>
    <p>{{THEORY_LINE3}}</p>
    <p>{{THEORY_LINE4}}</p>
    <p>{{THEORY_LINE5}}</p>

    <h5>
        {{ELEMENTSTEST_PUT_DROPPABLE}}:
        {{{STEP1_DROPPABLE}}}
        {{{STEP1_DRAGGABLE}}}
    </h5>

    <p>{{THEORY_LINE6}}</p>
    <p>{{THEORY_LINE7}}</p>

```



```
<h5>
    {{ELEMENTSTEST_RADIOS}}:
    <br/>
</h5>
<h5>
    {{{STEP1_1_RADIOS}}}
</h5>

<p>{{THEORY_LINE8}}</p>

<h5>
    {{ELEMENTSTEST_CHOOSE_WRITE_OPTION1}}:
    {{{STEP1_SELECT1}}}
</h5>

<h5>
    {{ELEMENTSTEST_CHOOSE_WRITE_OPTION2}}:
    {{{STEP1_SELECT2}}}
</h5>

<h5>
    {{ELEMENTSTEST_CHOOSE_WRITE_OPTION3}}:
    {{{STEP1_SELECT3}}}
</h5>
<p>{{THEORY_LINE9}}</p>

<h5>
    {{ELEMENTSTEST_ENTER_TEST}}:
    {{{STEP1_TEXTINPUT}}}
    {{ELEMENTSTEST_ENTER_TEST1}}
```

```

</h5>
<p>{{THEORY_LINE10}}</p>
<p>{{THEORY_LINE11}}</p>

```

```

<div style="text-align: center;"><button class="btn btn-success
check">{{CHECK}}</button></div>
</div>

```

step2_Nelder-Mead.html

```

<div class="jumbotron page2">

  <h1>{{ELEMENTSTEST_TITLE2}}</h1>
  <h4>{{EX_STEP2_0}}</h4>
  <p>{{EX_STEP2_1}}</p>
  <p>{{EX_STEP2_1_2}}</p>

  <h4>
    {{{STEP2_LATEX_1}}}
  </h4>

  <br>
  <p>{{EX_STEP2_2}}</p>

  <h4>
    {{{STEP2_LATEX_2}}} {{{STEP2_INPUT1}}}
  </h4>
  <h4>
    {{{STEP2_LATEX_3}}} {{{STEP2_INPUT2}}}
  </h4>

  <p>{{EX_STEP2_3}}</p>
  <h5>{{{STEP2_LATEX_4}}} {{{STEP2_INPUT3}}}</h5>

```

<h5>{{{STEP2_LATEX_5}}} {{{STEP2_INPUT4}}}</h5>

<p>{{EX_STEP2_4}}</p>

<h5>{{{STEP2_LATEX_6}}} {{{STEP2_INPUT5}}} {{{STEP2_LATEX_7}}} {{{STEP2_INPUT6}}} {{{STEP2_LATEX_8}}} {{{STEP2_INPUT7}}}</h5>

<p>{{EX_STEP2_5}}</p>

</br>

<p>{{EX_STEP2_6}}</p>

<h5>{{{STEP2_LATEX_9}}} {{{STEP2_INPUT8}}}</h5>

<h5>{{{STEP2_LATEX_10}}} {{{STEP2_INPUT9}}}</h5>

<p>{{EX_STEP2_7}}</p>

<h5>{{{STEP2_LATEX_11}}} {{{STEP2_INPUT10}}}</h5>

<h5>{{{STEP2_LATEX_12}}} {{{STEP2_INPUT11}}}</h5>

<p>{{EX_STEP2_8}}</p>

</br>

<p>{{EX_STEP2_9}}</p>

<h4>

{{{STEP2_LATEX_13}}} {{{STEP2_INPUT12}}}

{{{STEP2_LATEX_14}}} {{{STEP2_INPUT13}}}

</h4>

<p>{{EX_STEP2_10}}</p>

<div style="text-align: center;"><button class="btn btn-success check">{{CHECK}}</button></div>

```
</div>
```

step3_Nelder-Mead.html

```
<div class="jumbotron page1">
```

```
  <h1>{{ELEMENTSTEST_TITLE3}}</h1>
```

```
  <p>{{STEP3_TEXT1}}</p>
```

```
  <p>{{STEP3_TEXT2}}</p>
```

```
  <h5>{{{STEP3_LATEX_1}}} {{{STEP3_INPUT1}}}</h5>
```

```
  <h5>{{{STEP3_LATEX_2}}} {{{STEP3_INPUT2}}}</h5>
```

```
  <h5>{{{STEP3_LATEX_3}}} {{{STEP3_INPUT3}}}</h5>
```

```
  <p>{{STEP3_TEXT3}}</p>
```

```
  <h5>{{{STEP3_LATEX_4}}} {{{STEP3_INPUT4}}}</h5>
```

```
  <p>{{STEP3_TEXT4}}</p>
```

```
  <h5>{{{STEP3_LATEX_5}}} {{{STEP3_INPUT5}}}</h5>
```

```
  <p>{{STEP3_TEXT5}}</p>
```

```
  </br>
```

```
  <p>{{STEP3_TEXT6}}</p>
```

```
  <h5>{{{STEP3_LATEX_6}}} {{{STEP3_INPUT6}}}</h5>
```

```
  <h5>{{{STEP3_LATEX_7}}} {{{STEP3_INPUT7}}}</h5>
```

```
    <div style="text-align: center;"><button class="btn btn-success
check">{{CHECK}}</button></div>
```

```
</div>
```

results.html

```
<div class="row jumbotron">
```

```
  <div class="col-md-5">
```

```

        <div id="gc"></div>

</div>

<div class="col-md-7">

    <h1>{{RESULTS_TITLE}}</h1>

    <p>

        <h5><strong>{{RESULTS_START_TIME}}:</strong> {{START_TIME}}</h5>

        <h5><strong>{{RESULTS_END_TIME}}:</strong> {{END_TIME}}</h5>

        <h5><strong>{{RESULTS_TIME_DIFF}}:</strong>
        <span
class="badge">{{TIME_DIFF}} {{RESULTS_TIME_SEC}}</span></h5>

        <br/>

        <h2><strong>{{RESULTS_YOUR_SCORE}}:
</strong>{{RESULTS_POINTS_IN_PERCENT}}%</h2>

    </p>

</div>

</div>

```

index.html

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
    </head>
    <body style="margin:0; padding:0; overflow:hidden;">
        <iframe src="trainer.html" frameborder="0" style="position: absolute;
height: 100%;" height="100%" width="100%"></iframe>
    </body>
</html>

```

report.html

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=koi8-r" />
<title>test2</title>
</head>
<body bgcolor="#ffffff">
<br><br><p align="center">
<object
classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553510000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#
version=8,0,0,0" width="1000" height="1000" id="test2" align="middle">
<param name="allowScriptAccess" value="sameDomain" />
<param name="movie" value="test2.swf" /><param name="quality" value="high"
/><param name="bgcolor" value="#ffffff" /><embed src="test2.swf" quality="high"
bgcolor="#ffffff" width="1000" height="1000" name="test2" align="middle"
allowScriptAccess="sameDomain"
type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object></p>
</body>
</html>

```

introduction_Hook_Jeeves.js

```

var introduction = function () {
  this.preDispatch = function () {

  };
  this.postDispatch = function () {
    $('div.flag-icon').each(function() {
      $(this).attr('onclick', 'window.location.href = \'?lang=\' +
$(this).attr("id")');
    });
  };
  this.mustache = function () {

```

```

        return {
            STEPS_COUNT: Rotator.getStepsCount()-2
        }
    }
};

step1_Hook_Jeeves.js

var VStep1;// A variable for future validator
var step1 = function () {

    var plot = null;
    this.preDispatch = function (callback) {
        var w = new WolframAlpha(); // Making a call to wolfram api to build a
plot
        w.setQuery('3x^2+2x+5').plot(function (data) {
            plot = '<img src="' + data + '>';
            callback();
        });
        //Making a call to LaTeX to generate an image of formula
        step1_latex = null;
        step1_latex = new LaTeX();
        step1_latex.setFormula("P_{\\i}=b_{\\i}+2(b_{\\i+1}-b_{\\i})");
    };

    this.postDispatch = function () {
        VStep1 = new Validator();
        //first param - DOM object, second - correct value or array of values,
third - if there are multiple correct answers
        //fourth - if you need to have 2 answers at the same time to be entered
        VStep1.addValidator($('select[name="step1-select"]'), 2)
            .addValidator($('input[name="step1-textinput"]'), ['b{i+1}'], true)
//It means user is required to enter both 'test' and 'text' separated by comas.
They will be validated separately

```

```

        .addValidator($('div.droppable[name="step1-droppable"]'), ['one',
'four'], false, true) //<----- Please, don't use numbers as values if you want
multiple answers at same time

        .addValidator($('input[name="step1-1-radios"]'), 'fourr')

        .setStrictMode(true) // Restrict number of attempts to 3 (default)

        .setIgnoreCase(false) // Ignore letter case (eg. TEXT, text)

        .enableStepFinishAlert(true); // Enable showing alert after step is
done

        //disableAnswersBacklight(true); //-- Disable green/red color of
correct/incorrect answers

        $('page1 button.check').click(function () {
            VStep1.setAttemptsOnCheckButton($(this)); //dynamically changing
amount of attempts left on check button

            VStep1.validate(); // validate the validators
        });
    };

    this.mustache = function () {
        return {
            STEP1_SELECT: new Select('step1-select')
                .addOption('{{ELEMENTSTEST_OPTION_ONE}}', 0)
                .addOption('{{ELEMENTSTEST_OPTION_TWO}}', 1)
                .addOption('{{ELEMENTSTEST_OPTION_THREE}}', 2)
                //randomize() -- You can randomize select choice elements
                .render(),
            STEP1_DROPPABLE: new DroppableArea('step1-droppable')
                .addClass('input')
                .render(),
            STEP1_DRAGGABLE: new DraggableGroup('step1-draggable')
                .addClass('value')
                .addOption('{{ELEMENTSTEST_PUT_DROPPABLE_ANSWER1}}', 'one')
                .addOption('{{ELEMENTSTEST_PUT_DROPPABLE_ANSWER2}}', 'two')

```



```

        .addOption('{{ELEMENTSTEST_PUT_DROPPABLE_ANSWER3}}', 'three')
        .addOption('{{ELEMENTSTEST_PUT_DROPPABLE_ANSWER4}}', 'four')
        .randomize() // You can randomize draggable elements
        .render(),
STEP1_TEXTINPUT: new TextInput('step1-textinput')
    .render(),
STEP1_1_RADIOS: new Radios('step1-1-radios')
    .addRadio('{{RADIO_TEXT_1_1}}', 'onee')
    .addRadio('{{RADIO_TEXT_2_1}}', 'twoo')
    .addRadio('{{RADIO_TEXT_3_1}}', 'threee')
    .addRadio('{{RADIO_TEXT_4_1}}', 'fourr')
    .randomize() // You can randomize radio elements
    .render(),
STEP1_LATEX: step1_latex != null ? step1_latex.render() : ""
    }
}
};

```

step2_Hook_Jeeves.js

```

var VStep2;// A variable for future validator
var step2 = function () {
    this.preDispatch = function () {
        step2_latex = null;
        step2_latex = new LaTeX();
        step2_latex.setFormula("f(x)=(x_{\\1}-2)^2 + (x_{\\2}-5)^2 + (x_{\\3}+2)^4");
    };

    this.postDispatch = function () {
        VStep2 = new Validator();
    };
};

```

```

//first param - DOM object, second - correct value or array of values,
third - if there are multiple correct answers
//fourth - if you need to have 2 answers at the same time to be entered
VStep2
    .addValidator($('input[name="step2-input1"]'), ['4 -2 3'], true)
    .addValidator($('input[name="step2-input2"]'), ['678'], true)
    .addValidator($('input[name="step2-input3"]'), ['5'], true)
    .addValidator($('input[name="step2-input4"]'), ['683'], true)
    .addValidator($('input[name="step2-input5"]'), ['3'], true)
    .addValidator($('input[name="step2-input6"]'), ['675'], true)
    .addValidator($('input[name="step2-input7"]'), ['3 -1'], true)
    .addValidator($('input[name="step2-input8"]'), ['662'], true)
    .addValidator($('input[name="step2-input9"]'), ['3 -3'], true)
    .addValidator($('input[name="step2-input10"]'), ['690'], true)
    .addValidator($('input[name="step2-input11"]'), ['3 -1 4'], true)
    .addValidator($('input[name="step2-input12"]'), ['1333'], true)
    .addValidator($('input[name="step2-input13"]'), ['3 -1 2'], true)
    .addValidator($('input[name="step2-input14"]'), ['293'], true)
    .addValidator($('input[name="step2-input15"]'), ['3 -1 2'], true)
    .setStrictMode(true) // Restrict number of attempts to 3 (default)
    .setIgnoreCase(false) // Ignore letter case (eg. TEXT, text)
    .enableStepFinishAlert(true); // Enable showing alert after step is
done
        //disableAnswersBacklight(true); //-- Disable green/red color of
correct/incorrect answers

    $('page2 button.check').click(function () {
        VStep2.setAttemptsOnCheckButton($(this)); //dynamically changing
amount of attempts left on check button
        VStep2.validate(); // validate the validators
    });
};
};

```

```
this.mustache = function () {  
  return {  
    STEP2_INPUT1: new TextInput('step2-input1')  
      .render(),  
    STEP2_INPUT2: new TextInput('step2-input2')  
      .render(),  
    STEP2_INPUT3: new TextInput('step2-input3')  
      .render(),  
    STEP2_INPUT4: new TextInput('step2-input4')  
      .render(),  
    STEP2_INPUT5: new TextInput('step2-input5')  
      .render(),  
    STEP2_INPUT6: new TextInput('step2-input6')  
      .render(),  
    STEP2_INPUT7: new TextInput('step2-input7')  
      .render(),  
    STEP2_INPUT8: new TextInput('step2-input8')  
      .render(),  
    STEP2_INPUT9: new TextInput('step2-input9')  
      .render(),  
    STEP2_INPUT10: new TextInput('step2-input10')  
      .render(),  
    STEP2_INPUT11: new TextInput('step2-input11')  
      .render(),  
    STEP2_INPUT12: new TextInput('step2-input12')  
      .render(),  
    STEP2_INPUT13: new TextInput('step2-input13')  
      .render(),  
    STEP2_INPUT14: new TextInput('step2-input14')  
      .render(),  
    STEP2_INPUT15: new TextInput('step2-input15')
```

```

        .render(),
        STEP2_LATEX: step2_latex != null ? step2_latex.render() : ""
    }
}
};

```

step3_Hook_Jeeves.js

```

var VStep3;// A variable for future validator
var step3 = function () {
    var plot = null;
    this.preDispatch = function (callback) {
        var w = new WolframAlpha(); // Making a call to wolfram api to build a
plot
        w.setQuery('3x^2+2x+5').plot(function (data) {
            plot = '.page1 button.check').click(function () {
            VStep3.setAttemptsOnCheckButton($(this)); //dynamically changing
amount of attempts left on check button
            VStep3.validate(); // validate the validators
        });
    };

```

```

this.mustache = function () {
    return {
        STEP3_INPUT1: new TextInput('step3-input1')
            .render(),
        STEP3_INPUT2: new TextInput('step3-input2')
            .render(),
        STEP3_LATEX: step3_latex != null ? step3_latex.render() : ""
    }
}

```

```
};
```

results.js

```

var results = function () {

    this.preDispatch = function () {
        Scorer.end();
    };

    this.postDispatch = function () {

```

```

    if (PRODUCTION)
        Service.pushResults();
};
this.mustache = function () {
    return {
        START_TIME: Scorer.getFormattedStartTime(),
        END_TIME: Scorer.getFormattedEndTime(),
        TIME_DIFF: Scorer.getTimeDifference(),
        SCORE: Scorer.getScore(),
        RESULTS_POINTS_IN_PERCENT: Scorer.getScoreInPercent()
    }
};

```

step1_Nelder-Mead.js

```

var VStep1;
var step1 = function () {

    var plot = null;
    this.preDispatch = function (callback) {
        var w = new WolframAlpha(); // Making a call to wolfram api to build a
plot
        w.setQuery('3x^2+2x+5').plot(function (data) {
            plot = '<img src="' + data + '>';
            callback();
        });
        //Making a call to LaTeX to generate an image of formula
        step1_latex = null;
        step1_latex = new Latex();
        step1_latex.setFormula("P_{\\i}=b_{\\i}+2(b_{\\i+1}-b_{\\i})");
    };
};

```

```

this.postDispatch = function () {
    VStep1 = new Validator();

    VStep1.addValidator($('select[name="step1-select"]'), 0)
        .addValidator($('select[name="step2-select"]'), 0)
        .addValidator($('select[name="step3-select"]'), 2)
        .addValidator($('input[name="step1-textinput"]'), ['стискання'],
true)
        .addValidator($('div.droppable[name="step1-droppable"]'), ['one',
'two', 'three'], false, true)
        .setStrictMode(true) // Restrict number of attempts to 3 (default)
        .setIgnoreCase(false) // Ignore letter case (eg. TEXT, text)
        .enableStepFinishAlert(true); // Enable showing alert after step is
done

    $('page1 button.check').click(function () {
        VStep1.setAttemptsOnCheckButton($(this)); //dynamically changing
amount of attempts left on check button
        VStep1.validate(); // validate the validators
    });
};

this.mustache = function () {
    return {
        STEP1_SELECT1: new Select('step1-select')
            .addOption('{{ELEMENTSTEST_OPTION1_ONE}}', 0)
            .addOption('{{ELEMENTSTEST_OPTION1_TWO}}', 1)
            .addOption('{{ELEMENTSTEST_OPTION1_THREE}}', 2)
            //randomize() -- You can randomize select choice elements
            .render(),
        STEP1_SELECT2: new Select('step2-select')
            .addOption('{{ELEMENTSTEST_OPTION2_ONE}}', 0)

```

```

    .addOption('{{ELEMENTSTEST_OPTION2_TWO}}', 1)
    .addOption('{{ELEMENTSTEST_OPTION2_THREE}}', 2)
    //randomize() -- You can randomize select choice elements
    .render(),
STEP1_SELECT3: new Select('step3-select')
    .addOption('{{ELEMENTSTEST_OPTION3_ONE}}', 0)
    .addOption('{{ELEMENTSTEST_OPTION3_TWO}}', 1)
    .addOption('{{ELEMENTSTEST_OPTION3_THREE}}', 2)
    //randomize() -- You can randomize select choice elements
    .render(),
STEP1_DROPPABLE: new DroppableArea('step1-droppable')
    .addClass('input')
    .render(),
STEP1_DRAGGABLE: new DraggableGroup('step1-draggable')
    .addClass('value')
    .addOption('{{ELEMENTSTEST_PUT_DROPPABLE_ANSWER1}}', 'one')
    .addOption('{{ELEMENTSTEST_PUT_DROPPABLE_ANSWER2}}', 'two')
    .addOption('{{ELEMENTSTEST_PUT_DROPPABLE_ANSWER3}}', 'three')
    .addOption('{{ELEMENTSTEST_PUT_DROPPABLE_ANSWER4}}', 'four')
    .randomize() // You can randomize draggable elements
    .render(),
STEP1_TEXTINPUT: new TextInput('step1-textinput')
    .render(),
STEP1_1_RADIOS: new Radios('step1-1-radios')
    .addRadio('{{RADIO_TEXT_1_1}}', 'onee')
    .addRadio('{{RADIO_TEXT_2_1}}', 'twoo')
    .addRadio('{{RADIO_TEXT_3_1}}', 'threee')
    .addRadio('{{RADIO_TEXT_4_1}}', 'fourr')
    .randomize() // You can randomize radio elements
    .render(),
STEP1_LATEX: step1_latex != null ? step1_latex.render() : ""

```



```

    }
  }
};
step2_Nelder-Mead.js

var VStep2;// A variable for future validator
var step2 = function () {
  this.preDispatch = function () {
    step2_1_latex = null;
    step2_1_latex = new Latex();
    step2_1_latex.setFormula("f(x)=(1+x_{\\1})^2 + (2-x_{\\2})^2");

    step2_2_latex = null;
    step2_2_latex = new Latex();
    step2_2_latex.setFormula("\\delta_{\\1} = (\\frac{\\sqrt{n+1}}{n-1}
  }{n\\sqrt{2}} )h = ");

    step2_3_latex = null;
    step2_3_latex = new Latex();
    step2_3_latex.setFormula("\\delta_{\\1} = (\\frac{\\sqrt{n+1}}{n-1}
  }{n\\sqrt{2}} )h = ");

    step2_4_latex = null;
    step2_4_latex = new Latex();
    step2_4_latex.setFormula("x_{\\1} = ");

    step2_5_latex = null;
    step2_5_latex = new Latex();
    step2_5_latex.setFormula("x_{\\2} = ");

    step2_6_latex = null;
    step2_6_latex = new Latex();
    step2_6_latex.setFormula("f(x_{\\0}) = ");
  }
};

```

```

step2_7_latex = null;
step2_7_latex = new LaTeX();
step2_7_latex.setFormula("f(x_{\\1}) = ");

```

```

step2_8_latex = null;
step2_8_latex = new LaTeX();
step2_8_latex.setFormula("f(x_{\\2}) = ");

```

```

step2_9_latex = null;
step2_9_latex = new LaTeX();
step2_9_latex.setFormula("x_{\\0} = 1/2(x_{g} + x_{1}) = ");

```

```

step2_10_latex = null;
step2_10_latex = new LaTeX();
step2_10_latex.setFormula("f(x_{\\0}) = ");

```

```

step2_11_latex = null;
step2_11_latex = new LaTeX();
step2_11_latex.setFormula("x_{r} = (1 + \\alpha)x_{\\0} - \\alpha x_{h}
= (1 + 1)x_{\\0} - 1 x_{h} = ");

```

```

step2_12_latex = null;
step2_12_latex = new LaTeX();
step2_12_latex.setFormula("f(x_{r}) = f_{r} = ");

```

```

step2_13_latex = null;
step2_13_latex = new LaTeX();
step2_13_latex.setFormula("\\overline{f} = \\sum_{i=1}^{n} /(n+1) =");

```

```

step2_14_latex = null;
step2_14_latex = new LaTeX();

```

```

        step2_14_latex.setFormula("\\sigma ^{2} = \\sum_{i=1}^{n+1} (f_{i} - \\overline{f})^2 / (n+1)");
    };

    this.postDispatch = function () {
        VStep2 = new Validator();
        VStep2
            .addValidator($('input[name="step2-input1"]'), ['1.9318'], true)
            .addValidator($('input[name="step2-input2"]'), ['0.5176'], true)
            .addValidator($('input[name="step2-input3"]'), ['0,5176 1,9318'],
true)
            .addValidator($('input[name="step2-input4"]'), ['1,9318 0,5176'],
true)
            .addValidator($('input[name="step2-input5"]'), ['5'], true)
            .addValidator($('input[name="step2-input6"]'), ['0.2374'], true)
            .addValidator($('input[name="step2-input7"]'), ['3.0658'], true)
            .addValidator($('input[name="step2-input8"]'), ['1.2247 1.2247'],
true)
            .addValidator($('input[name="step2-input9"]'), ['0.6516'], true)
            .addValidator($('input[name="step2-input10"]'), ['2.4494 2.4494'],
true)
            .addValidator($('input[name="step2-input11"]'), ['2.3027'], true)
            .addValidator($('input[name="step2-input12"]'), ['1.8686'], true)
            .addValidator($('input[name="step2-input13"]'), ['1.4275'], true)
            .setStrictMode(true) // Restrict number of attempts to 3 (default)
            .setIgnoreCase(false) // Ignore letter case (eg. TEXT, text)
            .enableStepFinishAlert(true); // Enable showing alert after step is
done
            // .disableAnswersBacklight(true); //-- Disable green/red color of
correct/incorrect answers

        $('page2 button.check').click(function () {
            VStep2.setAttemptsOnCheckButton($(this)); //dynamically changing
amount of attempts left on check button

```

```
        VStep2.validate(); // validate the validators
    });
};

this.mustache = function () {
    return {
        STEP2_INPUT1: new TextInput('step2-input1')
            .render(),
        STEP2_INPUT2: new TextInput('step2-input2')
            .render(),
        STEP2_INPUT3: new TextInput('step2-input3')
            .render(),
        STEP2_INPUT4: new TextInput('step2-input4')
            .render(),
        STEP2_INPUT5: new TextInput('step2-input5')
            .render(),
        STEP2_INPUT6: new TextInput('step2-input6')
            .render(),
        STEP2_INPUT7: new TextInput('step2-input7')
            .render(),
        STEP2_INPUT8: new TextInput('step2-input8')
            .render(),
        STEP2_INPUT9: new TextInput('step2-input9')
            .render(),
        STEP2_INPUT10: new TextInput('step2-input10')
            .render(),
        STEP2_INPUT11: new TextInput('step2-input11')
            .render(),
        STEP2_INPUT12: new TextInput('step2-input12')
            .render(),
        STEP2_INPUT13: new TextInput('step2-input13')
```

```

        .render(),
    STEP2_LATEX_1: step2_1_latex != null ? step2_1_latex.render() : "",
    STEP2_LATEX_2: step2_2_latex != null ? step2_2_latex.render() : "",
    STEP2_LATEX_3: step2_3_latex != null ? step2_3_latex.render() : "",
    STEP2_LATEX_4: step2_4_latex != null ? step2_4_latex.render() : "",
    STEP2_LATEX_5: step2_5_latex != null ? step2_5_latex.render() : "",
    STEP2_LATEX_6: step2_6_latex != null ? step2_6_latex.render() : "",
    STEP2_LATEX_7: step2_7_latex != null ? step2_7_latex.render() : "",
    STEP2_LATEX_8: step2_8_latex != null ? step2_8_latex.render() : "",
    STEP2_LATEX_9: step2_9_latex != null ? step2_9_latex.render() : "",
    STEP2_LATEX_10: step2_10_latex != null ? step2_10_latex.render() :
    "",
    STEP2_LATEX_11: step2_11_latex != null ? step2_11_latex.render() :
    "",
    STEP2_LATEX_12: step2_12_latex != null ? step2_12_latex.render() :
    "",
    STEP2_LATEX_13: step2_13_latex != null ? step2_13_latex.render() :
    "",
    STEP2_LATEX_14: step2_14_latex != null ? step2_14_latex.render() :
    "",
    }
}
};

```

step3_Nelder-Mead.js

```

var VStep3;
var step3 = function () {
    var plot = null;
    this.preDispatch = function (callback) {
        var w = new WolframAlpha();
        w.setQuery('3x^2+2x+5').plot(function (data) {
            plot = '<img src="' + data + '>';
            callback();
        });
    });
};

```

```
step3_latex1 = null;
step3_latex1 = new LateX();
step3_latex1.setFormula("x_{l}(1.1676; 2.1620); f_{l} =");

step3_latex2 = null;
step3_latex2 = new LateX();
step3_latex2.setFormula("x_{g}(0.9685; 1.7584); f_{g} =");

step3_latex3 = null;
step3_latex3 = new LateX();
step3_latex3.setFormula("x_{h} (0.5176; 1.9318); f_{h} =");

step3_latex4 = null;
step3_latex4 = new LateX();
step3_latex4.setFormula("x_{r} (1.6185; 1.9886); f_{r} =");

step3_latex5 = null;
step3_latex5 = new LateX();
step3_latex5.setFormula("x_{c} (1.7077; 1.3541); f_{c} =");

step3_latex6 = null;
step3_latex6 = new LateX();
step3_latex6.setFormula("x_{min} = ");

step3_latex7 = null;
step3_latex7 = new LateX();
step3_latex7.setFormula("f_{min} =");
};

this.postDispatch = function () {
```

```

VStep3 = new Validator();
VStep3
    .addValidator($('input[name="step3-input1"]'), ['0.0543'], true)
    .addValidator($('input[name="step3-input2"]'), ['0.0594'], true)
    .addValidator($('input[name="step3-input3"]'), ['0.2374'], true)
    .addValidator($('input[name="step3-input4"]'), ['3.4742'], true)
    .addValidator($('input[name="step3-input5"]'), ['0.0458'], true)
    .addValidator($('input[name="step3-input6"]'), ['0.7928 1.946'],
true)

    .addValidator($('input[name="step3-input7"]'), ['0.0458'], true)
    .setStrictMode(true) // Restrict number of attempts to 3 (default)
    .setIgnoreCase(false) // Ignore letter case (eg. TEXT, text)
    .enableStepFinishAlert(true); // Enable showing alert after step is
done

    $(''.page1 button.check').click(function () {
        VStep3.setAttemptsOnCheckButton($(this)); //dynamically changing
amount of attempts left on check button
        VStep3.validate(); // validate the validators
    });
};

this.mustache = function () {
    return {
        STEP3_INPUT1: new TextInput('step3-input1')
            .render(),
        STEP3_INPUT2: new TextInput('step3-input2')
            .render(),
        STEP3_INPUT3: new TextInput('step3-input3')
            .render(),
        STEP3_INPUT4: new TextInput('step3-input4')
            .render(),
    };
};

```

```

STEP3_INPUT5: new TextInput('step3-input5')
    .render(),
STEP3_INPUT6: new TextInput('step3-input6')
    .render(),
STEP3_INPUT7: new TextInput('step3-input7')
    .render(),
STEP3_LATEX_1: step3_latex1 != null ? step3_latex1.render() : "",
STEP3_LATEX_2: step3_latex2 != null ? step3_latex2.render() : "",
STEP3_LATEX_3: step3_latex3 != null ? step3_latex3.render() : "",
STEP3_LATEX_4: step3_latex4 != null ? step3_latex4.render() : "",
STEP3_LATEX_5: step3_latex5 != null ? step3_latex5.render() : "",
STEP3_LATEX_6: step3_latex6 != null ? step3_latex6.render() : "",
STEP3_LATEX_7: step3_latex7 != null ? step3_latex7.render() : ""
    }
}
};

```

trainer.config.json

```

{
  "LANGUAGES": {
    "ua": "Українська",
    "en": "English"
  },
  "DEFAULT_LANG": "ua",
  "TRAINER_SCORE": 100,
  "TRAINER_AUTHOR": "__",
  "COURSE_AUTHOR": "__",
  "PRODUCTION": false
}

```


trainer.steps.json

```
[
  {
    "filename"      : "introduction",
    "score"         : 0,
    "hasScript"    : true
  },
  {
    "filename"      : "step1",
    "score"         : 30,
    "hasScript"    : true
  },
  {
    "filename"      : "step2",
    "score"         : 40,
    "hasScript"    : true
  },
  {
    "filename"      : "step3",
    "score"         : 30,
    "hasScript"    : true
  },
  {
    "filename"      : "results",
    "score"         : 0,
    "hasScript"    : true
  }
]
```