

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток для вивчення основ комп'ютерної графіки»
за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студентка групи ІТ-61 Резнікова Аліса Сергіївна

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2020 р.

Науковий керівник

_____ к.т.н., доц., Ващенко С. М.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

_____ Шифрін Д. М.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2020 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Резнікова Аліса Сергіївна

1 Тема роботи Web-додаток для вивчення основ комп'ютерної графіки

Керівник роботи Ващенко Світлана Михайлівна, к.т.н., доцент

затверджені наказом по університету від «14» травня 2020 р. №0576-III

2 Строк подання студентом роботи «1» червня 2020 р.

3 Вхідні дані до роботи технічне завдання на розробку web-додатку, наданий перелік файлів для інтеграції розробки у ресурс e-learning у подальшому

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування веб-додатку, розробка веб-додатку

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність роботи, мета та цілі дипломного проекту, огляд існуючих аналогів, планування та проектування, інструменти розробки, етапи розробки.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____ 01.10.2019 _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	21.10.2019- 30.10.2019	
2	Огляд останніх досліджень	03.11.2019- 03.12.2019	
3	Проектування набору тестів за поданими темами	07.12.2019- 31.12.2019	
4	Вивчення наданої структури	01.02.2020- 29.02.2020	
5	Розробка програмної реалізації тестових завдань	25.03.2020- 07.04.2020	
6	Оформлення документації	11.04.2020- 31.05.2020	

Студент _____
(підпис)

Резнікова А.С.

Керівник роботи _____
(підпис)

к.т.н., доц. Ващенко С.М.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток для вивчення основ комп'ютерної графіки».

Пояснювальна записка складається зі вступу, трьох основних розділів, висновків, списку використаних джерел із 20 найменувань, додатків. Загальний обсяг пояснювальної записки складає 53 сторінок, у тому числі 44 сторінки основного тексту, 2 сторінки списку використаних джерел, 15 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці веб-додатку для вивчення основ програмування комп'ютерної графіки.

В роботі дослідження в області дистанційного навчання та онлайн-тренажерів, існуючих аналогів, виконана постановка завдання.

У роботі було виконано проектування, яке включає розробку IDEF0 та Use Case діаграм, а також планування, етапами якого є створення WBS та OBS діаграм, а також діаграми Ганта.

Результатом проведеної роботи є розроблений веб-додаток для вивчення комп'ютерної графіки, що може бути інтегрованим у систему elearning.

Практичне значення роботи полягає у забезпеченні студентів денної та заочної форм навчання тренажером із дисципліни «Комп'ютерна графіка».

Ключові слова: WEB-ДОДАТОК, ДИСТАНЦІЙНЕ НАВЧАННЯ, ELEARNING, JAVASCRIPT, HTML, CSS, BOOTSTRAP, THREE.JS, JSON, КОМП'ЮТЕРНА ГРАФІКА.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Огляд останніх досліджень і публікацій.....	8
1.2 Аналіз програмних продуктів - аналогів.....	10
1.3 Постановка задачі.....	15
2 ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ	18
2.1 Діаграми нотації IDEF0.....	18
2.2 Use Case Diagram	21
3 РОЗРОБКА ВЕБ-ДОДАТКУ	23
3.1 Структура проекту.....	23
3.2 Розробка тренажеру.....	24
3.3 Приклад роботи.....	32
ВИСНОВКИ	36
ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ	37
ДОДАТОК А	39
ДОДАТОК Б.....	47

ВСТУП

Серед усього розмаїття предметів, що входять до навчального курсу майбутніх програмістів, особливу увагу варто приділити дисципліні програмування комп'ютерної графіки. Курс даної дисципліни повністю забезпечує студентів необхідними теоретичними знаннями в цій галузі: викладаються основні методи роботи з OpenGL (починаючи з підготовки до робочого процесу і закінчуючи створенням інтерактивного додатку), теорія кольорів, геометричні перетворення тощо.

На сьогоднішній день існує великий спектр можливостей для вивчення та практики різних мов програмування. Проте виникла потреба розробити власний ресурс, який би дозволив студентам отримати потрібні навички в програмуванні комп'ютерної графіки.

Таким чином, на основі було сформовано мету дипломної роботи — створення веб- додатку, що у майбутньому буде приєднаний до платформи e-learning[13] та являтиме собою тренажер для роботи із OpenGL для студентів усіх форм навчання.

Для досягнення поставленої мети було поставлено ряд завдань:

- провести аналіз сучасного стану питання та відповідних програмних аналогів;
- ознайомитися з вимогами системи дистанційного навчання СумДУ до віртуальних тренажерів;
- визначити вимоги до тренажеру та скласти технічне завдання;
- обрати засоби реалізації;
- провести моделювання роботи тренажеру;
- виконати програмну реалізацію.

Практичне значення виконаної роботи полягає в тому, що студент отримує зручний інструмент для відпрацювання своїх навичок програмування комп'ютерної графіки.

Результатом роботи є відповідне програмне забезпечення, яке було представлено на конференції «Інформатика, математика, автоматизація ІМА2020» [21]

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

У час стрімкого розвитку інформаційних технологій більша частина повсякденних справ людини переходить у режим онлайн. Починаючи від здійснення покупок і закінчуючи оформленням важливої документації — всі подібні питання можуть бути вирішені одним натисканням миші. У наслідок цього людина проводить у всесвітній мережі значну частину свого часу: працюючи, спілкуючись у соціальних мережах, відпочиваючи за переглядом фільмів або читанням.

Навчання із використанням онлайн сервісів та тренажерів набуло швидкої популярності серед бажаючих зайнятись самоосвітою. На даний момент кількість сфер, у яких онлайн навчання є дієвим дійсно велика. Розглянемо ті напрямки, у яких дистанційне навчання та інтернет-сервіси мають найбільшу ефективність:

- вивчення іноземних мов;
- вивчення мов програмування[14];
- підготовка до шкільних екзаменів (ЗНО, ДПА) та робота за шкільною програмою;
- тренування пам'яті, уваги, швидкості мислення;
- навчання прикладному мистецтву — музиці, малюванню, фотографуванню тощо;

Великим розмаїття представляють і безпосередньо типи сервісів, що надають навчальні послуги. Онлайн ресурси різняться залежно від того, який саме предмет викладається. Можна виокремити три види сервісів за підходами до навчання: тренажери, онлайн-курси, навчальні посібники у комбінації із вправами до них. Найбільш зручними та актуальними є онлайн-тренажери із ряду наступних причин:

- необмеженність у часі;
- незалежність від викладачів;
- наявність усього навчального матеріалу у межах одного сервісу;
- структуровані уроки;
- незалежність від апаратного забезпечення.

Для вивчення мов програмування найкраще працюють тренажери, які надають можливість як вивчити теорію, так одразу і відпрацювати отримані навички. До таких можна віднести веб-ресурс CodeAcademy[9], мобільні додатки серій SoloLearn[11] та freeCodeCamp[10]. Основними перевагами роботи з подібними сервісами є наступні:

- користувач має можливість відпрацьовувати отримані знання безпосередньо після блоку теоретичної інформації;
- користувач має можливість роботи онлайн, тобто все, що потрібно — доступ до мережі Інтернет;
- наочне відображення результату компіляції коду.

Звичайно, робота із такими навчальними додатками має ряд певних недоліків, наприклад:

- урізаний функціонал безкоштовної версії або її відсутність;
- викладання виключно базового матеріалу;
- відсутність курсів саме із OpenGL[6].

Але, навіть з огляду на такі вагомні переваги освітніх сервісів, слід звернути увагу і на їх недоліки:

- обмежений функціонал безкоштовної версії або її відсутність;
- неможливість користування сервісами без реєстрації;
- наявність виключно базових уроків програмування;
- відсутність уроків саме з OpenGL[6].

З огляду на вищеперелічене, розробка даного веб-додатку надасть можливість користувачам безкоштовно навчатись програмуванню комп'ютерної графіки у онлайн-режимі. Структура веб-ресурсу

зумовлюватиме наявність послідовності вправ, які дозволять студентам відпрацьовувати навички роботи із лініями, точками та полігонами використовуючи функціонал OpenGL[15]. Даний проект буде нести соціально-педагогічну цінність, оскільки він спрямований на використання у рамках викладання дисципліни програмування комп'ютерної графіки у Сумському державному університеті. Використання розробленого у рамках дипломної роботи веб-сервісу має значно полегшити процес навчання для студентів усіх форм навчання.

1.2 Аналіз програмних продуктів - аналогів

Одним із перших етапів роботи над будь-яким навіть некомерційним проектом є огляд існуючих аналогів та рішень, що вже застосовуються для вирішення окресленої проблеми. Слід попередньо зазначити, що онлайн сервісів із вивчення саме OpenGL немає: існує достатня кількість книг та відеоруоків, але не тренажерів, ефективність яких є значно вищою. Тому розглянемо онлайн тренажери, які представляють студентам можливість у інтерактивній формі вивчати мови програмування.

Першим у списку аналогів розроблюваного продукту є веб-сервіс Codecademy[9]. Даний ресурс представляє собою інтерактивну онлайн-платформу, що присвячена вивченню таких мов програмування як Python, PHP, jQuery, JavaScript[16], Ruby, C++, Java, а також мов розмітки веб-сторінок HTML[17] та CSS[18]. Основним вирашним елементом даного ресурсу є елемент інтерактивності та певною мірою гейміфікації[12]. На рис. 1.1 представлена початкова сторінка Codecademy, де перш за все користувачеві пропонується авторизація або реєстрація на сервісі.

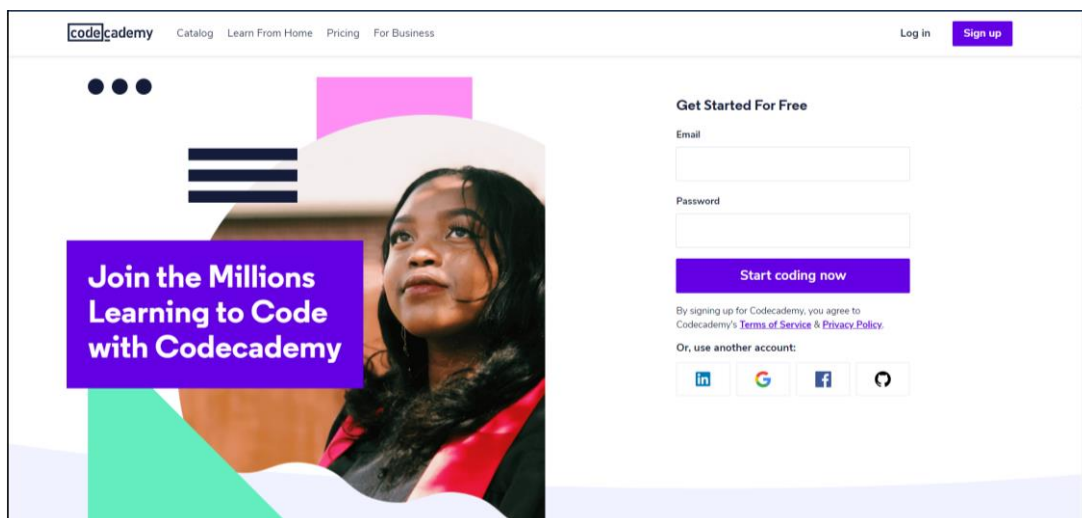


Рисунок 1.1 – Початкова сторінка Codecademy

Основною перевагою даного сервісу є інтерактивність та елемент гейміфікації[12]: користувач виконує покроково невеликі вправи – вивести на консоль повідомлення, створити змінну тощо. Також варто відзначити візуальну структуру процесу навчання. Робоча зона поділяється на три блоки — теоретична частина та список завдань, зона для написання коду та зона імітації консольного виводу. Така побудова робить процес навчання значно зручнішим, адже, по-перше, знання засвоюються дозовано і немає перевантаження, а по-друге є можливість миттєво відтворити вивчене.

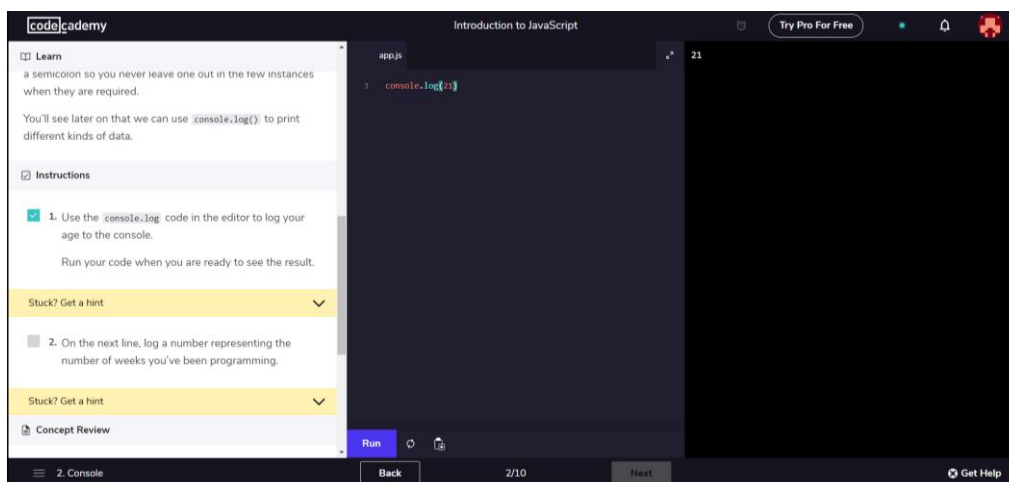


Рисунок 1.2 – Структура вправ на ресурсі Codecademy

Наступним проектом-аналогом є мобільний додаток SoloLearn[11]. Даний ресурс, як і його попередник, надає можливість вивчати широкий спектр мов програмування (рис. 1.3). Навчання побудовано у вигляді гри: потрібно накопичувати досвід, заробляти нагороди і навіть змагатися з іншими членами спільноти. Робити це можна за допомогою не тільки веб-сервісу, а й додатків на Android і iOS. Для запису коду і запуску програми є «пісочниця» - редактор і емулятор компілятора. По мірі просування ви будете заробляти очки і підвищувати свій рівень. Інтерфейс і структура роботи загалом дуже схожі на Codecademy, а тому і мають такі самі переваги.

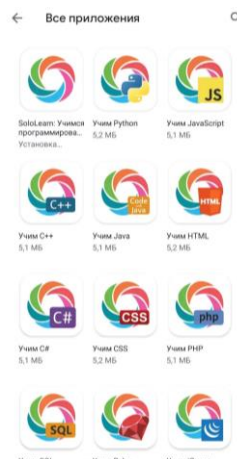


Рисунок 1.3 – Мови програмування, доступні до вивчення на SoloLearn



Рисунок 1.4 – Структура уроків на SoloLearn

Наступний аналог — freeCodeCamp[10]. Він представляє собою некомерційну організацію, яка складається з інтерактивної веб-платформи навчання, форуму в мережі спільноти, чатів, інтернет-видань та місцевих організацій, які мають намір зробити веб-розробки для навчання доступними для всіх. Починаючи з навчальних посібників, які знайомлять студентів з HTML[17], CSS[19] та JavaScript[20], студенти переходять до виконання завдань, які вони повинні виконати окремо або парами. Розглянемо безпосередньо веб-платформу. Структура уроків і інтерфейс у значній мірі співпадають із аналогами у даному переліку. Недоліками (у порівнянні із попередніми варіантами) можна визначити відсутність викладання українською або російською мовою, а також відсутність інтерактивних уроків із інших мов програмування, наприклад C++. Але, звичайно, це у певній мірі компенсується тим, що ресурс є безкоштовним на 100%.



Рисунок 1.5 – Перелік уроків на freeCodeCamp

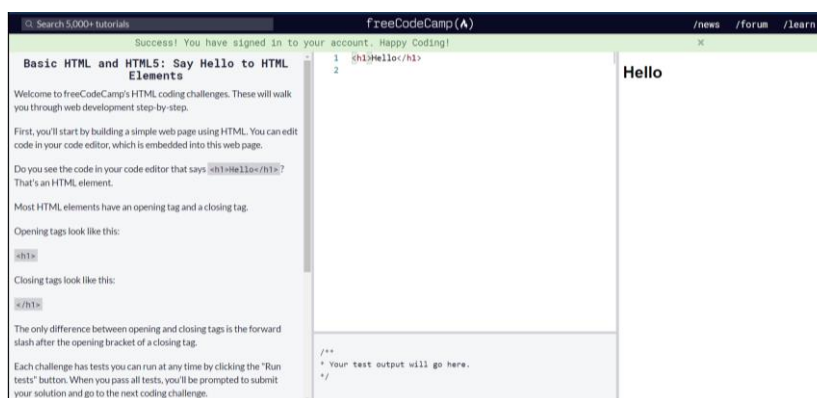


Рисунок 1.6 – Структура уроків на freeCodeCamp

Проаналізувавши переваги та недоліки вищеперелічених ресурсів була створена порівняльна характеристика аналогів та розробленого веб-ресурсу за певним переліком критеріїв і сформована табл 1.1.

Таким чином, вивчивши дану методологію викладання, а також дослідивши ринок на наявність тренажерів для відпрацювання навичок написання коду на OpenGL[14], можна сказати, що майбутня розробка є актуальною. Додаток, у першу чергу, розроблюватиметься для використання студентами та викладачами Сумського державного університету, тому відповідно отриманий продукт є безкоштовним для всіх студентів.

Таблиця 1.1 – Порівняльна характеристика аналогів

Назва критерію	Назва ресурсу			
	Codecademy	SoloLearn	freeCodeCamp	Розроблюваний ресурс
1. Інтерактивність	+	+	+	+
2. Використання повного функціоналу при наявності безкоштовного профілю	-	-	+	+
3. Курс із C++ OpenGL	-	-	-	+
4. Можливість роботи без реєстрації	-	-	-	+
5. Можливість інтеграції у elearning	-	-	-	+

1.3 Постановка задачі

Виходячи із вищепереліченого, можна визначити наступну мету роботи: розробити web-додаток для вивчення основ програмування комп'ютерної графіки. Він спрямований на часткову автоматизацію процесу викладання практичних занять із дисципліни програмування комп'ютерної графіки, а також надасть студентам усіх форм навчання можливість вивчати дисципліну у інтерактивній формі.

Практичне значення даного веб-додатку полягає у його використанні під час навчального процесу. Це дозволить студентам Сумського державного університету відпрацьовувати навички програмування комп'ютерної графіки у зручній та легкій для сприйняття формі. У ході реалізації проекту мають бути розроблені уроки, що включають практичні завдання, а також забезпечена стабільна робота продукту.

Оскільки розроблюваний продукт у першу чергу має бути інтерактивним, то його інтерфейс повинен бути інтуїтивно зрозумілим користувачеві. З огляду на те, що продукт буде інтегрований у систему elearning Сумського державного університету, визначені мови інтерфейсу — українська, англійська, російська. Як раніше було зазначено, необхідно вирішити наступний перелік задач для успішного виконання роботи:

Для досягнення мети необхідно:

- вивчити надану структуру тренажеру;
- виконати модифікацію структури;
- розробити функціонал тренажеру: реалізація вправ, симуляція роботи із кодом на C++, відображення результатів виконання коду.

Як і для будь-якого web-сервісу, розмітка сторінок виконується за допомогою HTML, CSS[19] та сервісів Bootstrap для більшої зручності та швидшого виконання.

Використовуючи мову JavaScript[20], JSON[] а також за допомогою інструментарію Three.js[8], проводиться реалізація головного модулю-симуляції та відтворення уроків.

Three.js[8] — це легка кросбраузерна бібліотека JavaScript, що використовується для створення та відображення 3D графіки у веб-додатках. Дана бібліотека поєднує у собі інтуїтивну легкість використання у порівнянні із WebGL, а також високу ефективність відтворення

зображень, оскільки не потребує підключення плагінів у браузері. До переваг даної бібліотеки можна віднести:

- 1) легкість оволодіння;
- 2) гарна документація та наявність великої кількості прикладів;
- 3) підтримка в усіх сучасних браузерах;
- 4) не потребує жодного клієнта або плагіна для роботи із моделями.

- 5) використовує графічні можливості комп'ютера користувача.

Виходячи із легкості у користуванні, а також схожості із OpenGL[7], Three.js був обраний у якості основного інструменту відображення проробленого користувачем у рамках уроку.

Технічне завдання на розробку веб-додатку наводиться у додатку А.

Планування виконання робіт у додатку Б.

2 ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ

Наступним кроком після проведення огляду останніх досліджень та публікацій, аналізу програмних продуктів-аналогів, а також встановлення мети проекту та засобів реалізації визначено мету проекту та задачі, які необхідні для його реалізації, є проектування веб-додатку. Проектування продукту, у загальноприйнятому розумінні, це процес реалізації абсолютно нового продукту, що базується саме на попиті на нього з боку майбутніх користувачів.

2.1 Діаграми нотації IDEF0

Першим етапом у процесі проектування веб-додатку є розробка контекстної діаграми A-0. Необхідність її створення полягає у тому, що дана діаграма містить зрозумілі пояснення та дає опис системи та її взаємодії з зовнішніми об'єктами.

Основні елементи даної діаграми поділяються на наступні групи:

- Вхідні дані — ті дані, що існують для використання продукту.
- Вихідні дані — результат, отриманий після використання продукту.
- Управління — дані, необхідні під час використання продукту.
- Механізми — актори, що використовують продукт.

З огляду на попередньо проведений аналіз, був сформований перелік головних елементів для контекстної діаграми «Організація навчання із вивчення основ КГ»:

- Вхідні дані: запит користувача до системи.

- Вихідні дані: практичні навички програмування та бали за проходження тесту.
- Управління: методологія викладання дисципліни програмування комп'ютерної графіки, система дистанційного навчання Сумського державного університету.
- Механізми: студенти-користувачі ресурсу, розроблюваний додаток, апаратне забезпечення.

На основі цих даних була розроблена контекстна діаграма, що представлена на рис.2.1.

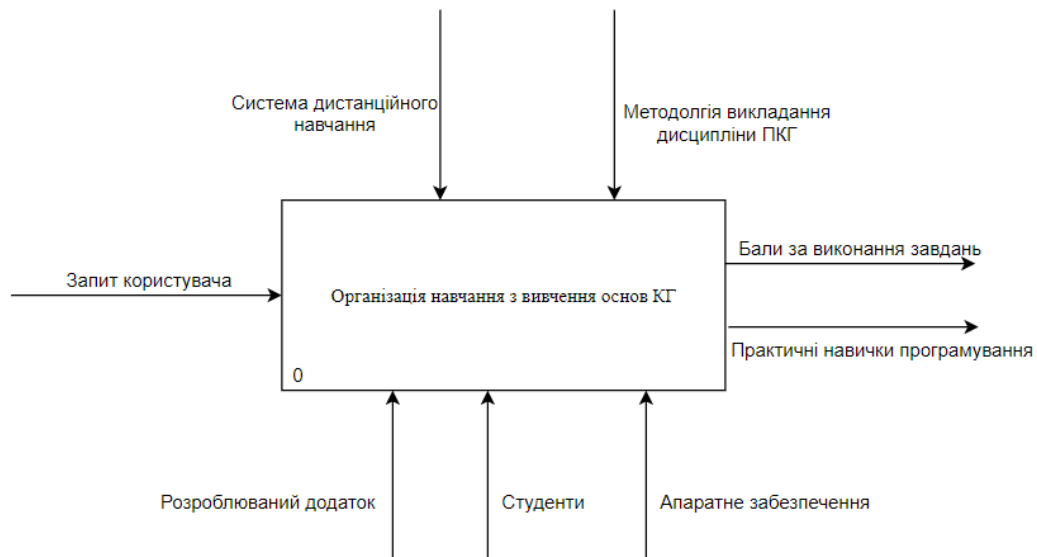


Рисунок 2.1 – Контекстна діаграма нульового рівня

Діаграма нульового рівня дає узагальнений опис системи та її взаємодії із навколишнім середовищем, тому наступним кроком проводимо декомпозицію. Такі дії дозволять зрозуміти логіку процесу розробки і встановити для команди більш чіткі задачі. IDEF0 була декомпована і тепер на діаграмі першого рівня представлено наступні процеси, що включені у проце організації навчання:

- Перший етап: виконання практичних завдань.

1) Вхідні дані: запит користувача до системи.

2) Вихідні дані: практичні навички із дисципліни програмування комп'ютерної графіки та факт проходження тестів.

1) Управління: система дистанційного навчання та методолгія викладання дисципліни КГ.

2) Механізми: студенти, розроблюваний додаток та апаратне забезпечення.

- Другий етап: отримання результатів роботи.

3) Вхідні дані: запит користувача до системи та факт проходження усіх тестів.

4) Вихідні дані: практичні навички із дисципліни програмування комп'ютерної графіки та оцінка.

5) Управління: система дистанційного навчання та методолгія викладання дисципліни КГ.

6) Механізми: студенти, розроблюваний додаток та апаратне забезпечення.

Діаграма другого рівня представлена на рис.2.2.

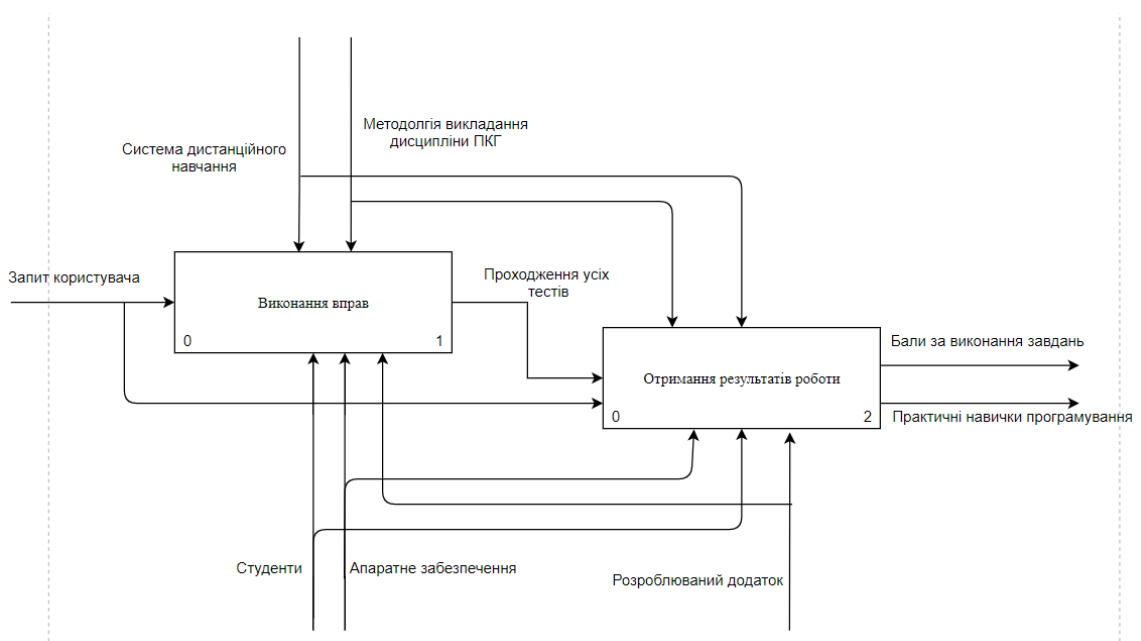


Рисунок 2.2 – Контекстна діаграма першого рівня декомпозиції

2.2 Use Case Diagram

Важливим етапом проектування будь-якого програмного продукту є створення Use Case Diagram (Діаграма варіантів використання). Така діаграма дозволяє зрозуміти та чітко окреслити залежність варіантів використання системи акторами, описати взаємодію веб-додатку та користувача.

У ході роботи над діаграмою були визначені актор системи User – користувач, що бажає віпрацювати навички програмування.

Наступним кроком після визначення акторів системи є формування списку варіантів використання системи. Даний перелік відповідає вимогам, які зазначені у додатку А.

Варіанти використання для веб-ресурсу:

- виконання вправ за розділом;
- отримання оцінки за урок.

На основі сформованих даних про акторів та всі можливі варіанти використання веб-ресурсу, була розроблена Use Case діаграма. Вона представлена на рис.2.3.

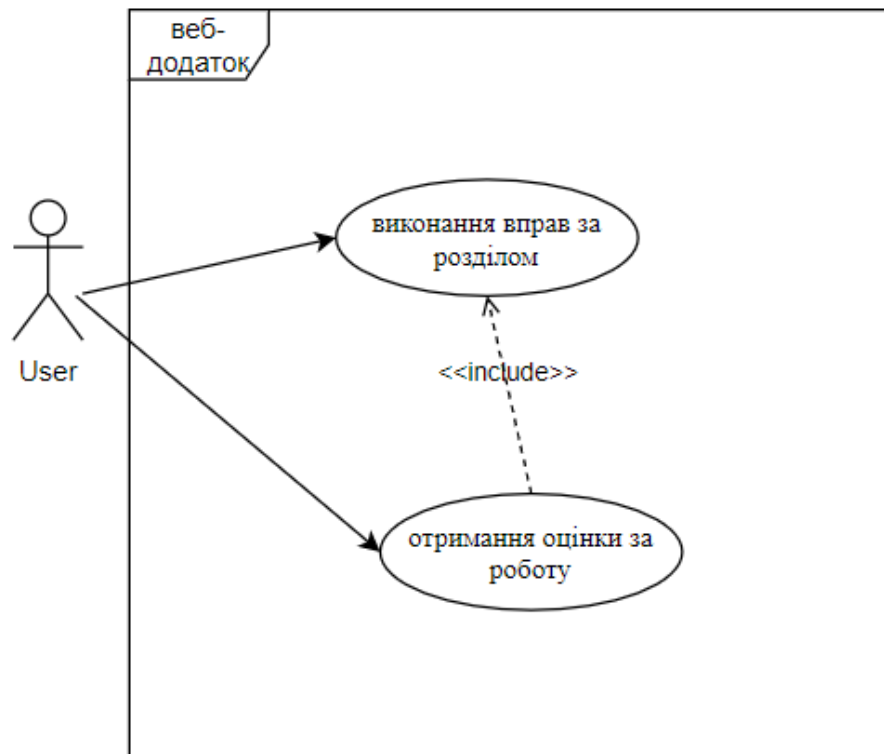


Рисунок 2.3 – Діаграма варіантів використання.

3 РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Структура проекту

Послідовним етапом після завершення проектування є реалізація продукту. Процес розробки програмного продукту проекту був розділений на два основних етапи: вивчення структури системи тренажерів, що була надана як основа Сумським державним університетом, і безпосередньо розробка інтерактивних тестів. Поданий набір файлів дозволяє розробляти будь-які тренажери із різноманітних дисциплін і в подальшому легко інтегрувати розробки у систему e-learning.

Структура наданої заздалегідь системи складається із наступних файлів та каталогів:

1) Каталог `css` та файл `trainer.custom.css` — файл стилів, що дозволяє розробнику вносити власні зміни у зовнішній вигляд сторінок. Окрім цього, зовнішній вигляд та розмітка ресурсу виконані із використанням Bootstrap 3.

2) Каталог `fonts` і файли, що вкладені в нього, визначають шрифти даного веб-додатку.

3) Основна логіка роботи реалізована у `js`-файлах, таких як `trainer.engine.js`, `introduction.js`, `results.js` та `step1.js`. На основі даних файлів, а також завдяки їх модифікації був створений ряд тестів, що представляють собою комбінації `html`-файлів та однойменних `js`-файлів.

4) `JSON`-файли, завдяки яким забезпечується мультимовність веб-додатку, а також зв'язок між кроками у тренажері.

5) Набір `html` сторінок, які є каркасом візуальної частини проекту.

Таким чином, дана структура є зручною не тільки із точки зору розробника для побудови вправ та тестів, але і для користувача, оскільки

інтерфейс є інтуїтивно зрозумілим, а також представлений вибір мов проходження тесту — українська, англійська або російська.

3.2 Розробка тренажеру

Після вивчення вищеописаної структури, були виділені та відтворені наступні кроки розробки власної системи тестів:

- 1) виділення основних підтем та визначення питань за ними.
- 2) створення html та javascript файлів, що визначатимуть наповнення завдання, а також реалізують головні механіки роботи із тестом.
- 3) редагування css-файлів для надання крокам тренажеру гармонійного та цілісного зовнішнього вигляду.

3.2.1 Розробка завдань

Під час розробки тренажеру першочергово було продумане та реалізоване наповнення веб-додатку із точки зору навчального процесу. У обраному підрозділі присутні три теми: робота із точками, робота із лініями та робота із полігонами (трикутниками). Кожна із даних тем включає вивчення ряду команд для роботи із геометричними примітивами, а також зумовлює тренування просторового мислення студента. Таким чином сформовано 8 питань тестового формату, що мають різну спрямованність — як для вивчення методів та функцій OpenGL, так і для розвитку логіки. Завдання, розроблені у ході роботи, зумовлюють наявність теоретичної бази у студента за таким переліком питань:

- 1) знання функцій відображення вершин, ліній, створення полігонів;
- 2) встановлення відповідного кольору у моделі rgb;

- 3) знання функцій зміни зовнішнього вигляду лінії;
- 4) робота із буферами;
- 5) робота із координатами у OpenGL.

Завдяки широкому розмаїттю представлених технічних можливостей наданої структури, для кожного завдання підібрані різні типи тестів:

- 1) вибір однієї правильної відповіді;
- 2) вибір декількох правильних варіантів;
- 3) обрання елемента із представленого випадючого списку;
- 4) питання з нечіткою відповіддю, де користувачеві пропонується самостійно ввести відповідь у відповідне поле введення;
- 5) перетягування вірних відповідей у спеціальне поле.

Отже, даний підхід не тільки дозволяє користувачеві перевірити свої знання із обраних тем, а й зробити це у інтерактивній манері. На наступному рисунку зображений початковий набір доступних варіантів взаємодій користувача та тренажеру.

Перевірка деяких елементів

Оберіть одну правильну відповідь: Оберіть елемент зі списку ▾

Введіть в поле текст 'test' и 'Text' через кому: Введіть текст

Перетягніть відповіді 1 та 4:

Answer 2 Answer 3 Answer 1 Answer 4

Введіть числа 1 АБО 4: Введіть текст

plot $3x^2+2x+5$:

(x from -1.6 to 0.9)

`glutVertex();`
 `glVertex3d();`
 `glauxVertex3d();`
 `glVertex();`

Введіть 4.5 або 4,5: Введіть текст

Рисунок 3.1 – Типи тестів

Для посилення ефекту інтерактивності, окрім тестової частини було вирішено розробити наступну схему взаємодії користувача із системою: відповівши вірно, користувачеві стає доступна візуалізація коду на OpenGL та відмальовані примітиви, що відтворюються із використанням Three.js.

3.2.2 Модифікація JSON файлів

Важливою частиною роботи над даним веб-додатком стала модифікація JSON файлів. Саме завдяки роботі із даною технологією забезпечується зв'язок головної сторінки та сторінок-кроків, а також зміна мови веб-додатку. Нижче перелічені файли та відповідні зміни, які були внесені для модифікації веб-додатку.

1) `trainer.config.json`: змінене значення поточного розробника, а також внесене ім'я автора курсу.

2) `trainer.steps.json`: додано нові блоки, які відповідають крокам тренажеру. Кожний блок містить назву html файлу, який відповідає тестовому завданню, значення кількості балів за завдання, а також ідентифікатор наявності відповідного javascript файлу.

3) `en.json`, `ru.json`, `ua.json`: у даних файлах внесені питання та варіанти відповідей, змінені назва тренажеру, заголовки сторінок, назви та кількість кроків тренажеру, тощо. Дана система дозволяє легко і компактно втілити складний механізм переключення мов веб-додатку, а також запобігти ускладненню програмного кода.

3.2.3 Розробка html та javascript файлів

Послідовно на основі отриманої бази було створено 16 нових файлів, що являють собою пари html та javascript. Кожна така пара є кроком у тренажері, на проходження якого дається три спроби.

Також необхідно зазначити, що до всіх файлів JSON та до головної сторінки потрібно внести зміни таким чином, щоб відображення ресурсу на різних мовах було коректним. Вносяться зміни до:

- формулювання питань;
- варіантів відповідей;
- назв кроків на різних мовах;
- елементів навігації.

На рисунку 3.2 наведено приклад внесення змін до файлу, який відповідає за англійську версію ресурсу.

```

50  {
51    "RESULTS_TITLE": "Your scores",
52    "RESULTS_START_TIME": "You started at",
53    "RESULTS_END_TIME": "Finished at",
54    "RESULTS_TIME_DIFF": "Time to complete",
55    "RESULTS_TIME_SEC": "sec.",
56    "RESULTS_YOUR_SCORE": "Your score is",
57
58    "STEP1_NAME": "Introduction",
59    "STEP2_NAME": "Step 1",
60    "STEP3_NAME": "Step 2",
61    "STEP4_NAME": "Step 3",
62    "STEP5_NAME": "Step 4",
63    "STEP6_NAME": "Step 5",
64    "STEP7_NAME": "Step 6",
65    "STEP8_NAME": "Step 7",
66    "STEP9_NAME": "Step 8", |
67    "STEP10_NAME": "Results"
68  }
69 }

```

Рисунок 3.2 – Внесення змін до JSON файлу

Переходимо до відповідного каталогу та створимо файли step1.html та step1.js. Дані файли представляють собою перший крок (завдання) тесту.

Створюється блок, у який вміщується тестове завдання — питання та варіанти відповіді, а також блок із сценою виконання коду на Three.js. Для створимо декілька нових блоків: канва для відмалювання сцени (що є новим елементом із появою HTML5), група радіо кнопок, безпосередньо питання та кнопка перевірки. На рисунку 3.3 нижче наведено приклад модифікації розмітки сторінки із першим завданням.



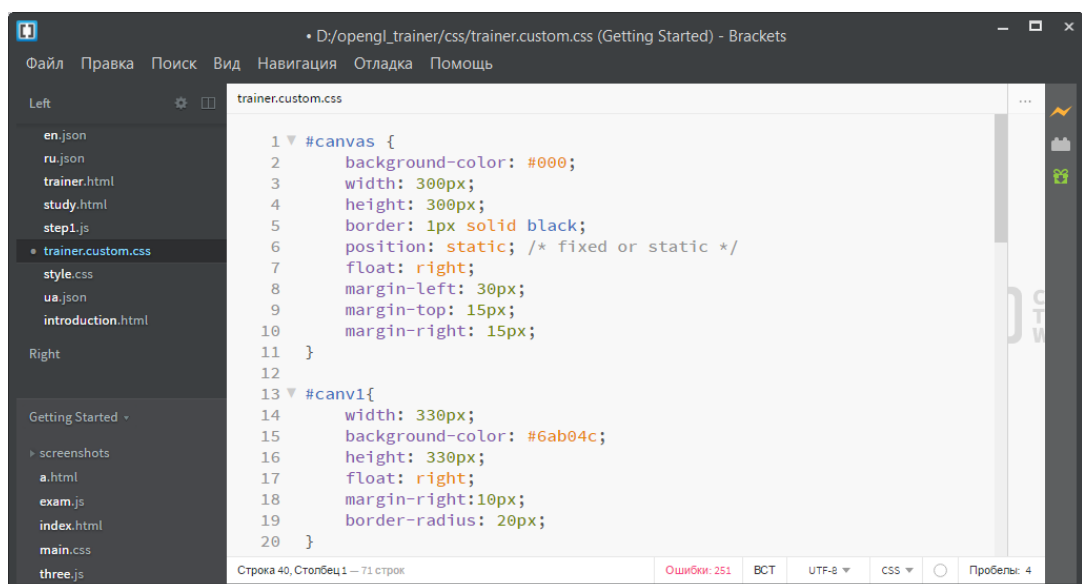
```

1 <div class="jumbotron page1">
2   <h1>{{ELEMENTSTEST_TITLE1}}</h1>
3
4   <div class="formcontainer">
5     <div class="formcontainer-sub">
6       <h4> <div id="canv1">
7         <div id="canvas" width="200" height="200">
8           </div>
9         </div>
10        <div id="tasktitle">
11          {{ELEMENTSTEST_CHOOSE_WRITE_OPTION}}:</div>
12        </h4>
13        <div class="form1">{{STEP1_RADIOS}}</div>
14        <button class="btn btn-success pull-right check">
15          {{CHECK}}</button>
16      </div>
17    </div>
18  </div>
19  <div id="chart" style="width: 600px; height: 400px;">
20    {{DIAG}}</div>
21 </div>

```

Рисунок 3.3 – Модифіковане представлення розмітки першого кроку тесту

Наступним кроком слід модифікувати вищезазначений файл стилів таким чином, щоб ергономічно та адаптивно розмістити елементи тесту, а також надати сторінці більш естетичного вигляду. На рисунках 3.4 та 3.5 представлено процес та результат модифікації CSS файлів. Кольори, використані для оформлення, підібрані згідно із кольоровою палітрою платформи дистанційного навчання SumDU MIX.



```

1 #canvas {
2   background-color: #000;
3   width: 300px;
4   height: 300px;
5   border: 1px solid black;
6   position: static; /* fixed or static */
7   float: right;
8   margin-left: 30px;
9   margin-top: 15px;
10  margin-right: 15px;
11 }
12
13 #canv1{
14   width: 330px;
15   background-color: #6ab04c;
16   height: 330px;
17   float: right;
18   margin-right: 10px;
19   border-radius: 20px;
20 }

```

Рисунок 3.3 – Приклад модифікації файлу стилів

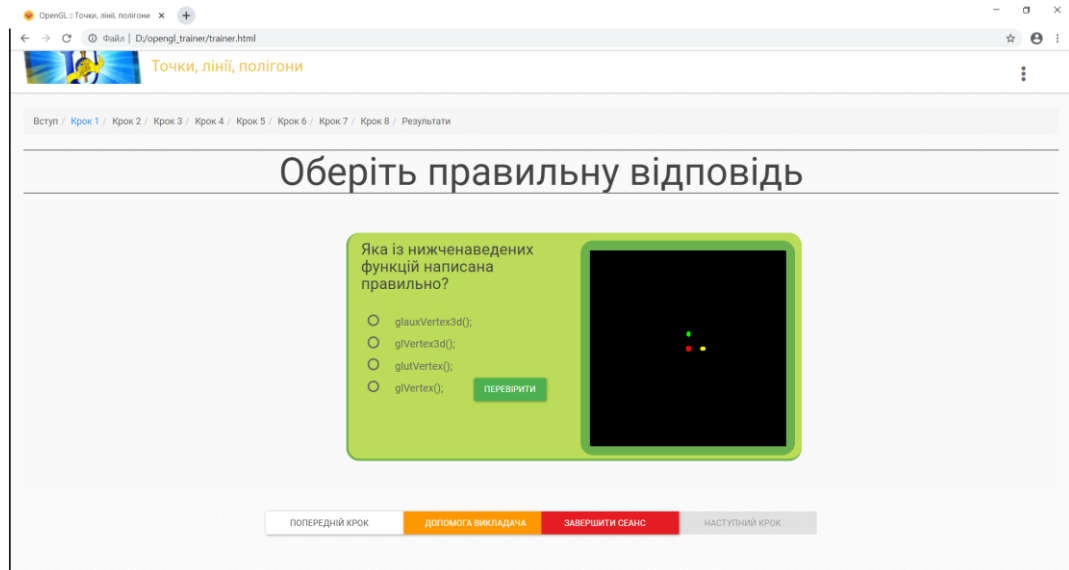


Рисунок 3.4 – Результат модифікації файлу стилів

Для коректного та цілісного функціонування тестів необхідно провести модифікації файлів javascript. Перейшовши до них, залишаємо рендеринг потрібних елементів (у даному випадку це група радіо кнопок), функцію валідації відповідних варіантів питань, а також встановлюємо значення валідатора на бажаний варіант відповіді. Таким чином, відповівши правильно, користувачеві нараховуватиметься 12,5 балів за кожну відповідь тесту, що може бути змінено у файлі JSON, який відповідає за конфігурацію балів за кожне завдання. Також у даному файлі можна змінити те, у якій послідовності виконуються кроки тесту. У випадку, якщо користувач зробить помилку, то він отримає відповідне повідомлення та підсвітку варіантів червоним кольором. Якщо відповідь вірна, то кнопка переходу на наступний крок стане доступною.



Рисунок 3.5 – Приклад модифікації файлу javascript

Наступним кроком було створення візуального представлення графічних примітивів як елементу інтерактивності та наочності. Як було зазначено у розділі 1.3, для цього скористаємось бібліотекою Three.js.

Для цього попередньо завантажений файл three.js додаємо до відповідного каталогу, що містить усі файли javascript, необхідні для функціонування веб-додатку. Дана бібліотека дозволить відтворювати будь-які графічні елементи, доступні при роботі із OpenGL.

У раніше створений блоковий елемент канви впишемо сцену. Для цього у файлі step1.js у функції, що відповідає за рендеринг елементів сторінки створимо нову сцену, а також встановимо камеру та розміри рендеру.

Нижче наведено приклад створення сцени, а також описано, яким чином вона була утримана у блоці.

```

container = document.getElementById( 'canvas' );
var scene = new THREE.Scene();
var camera = new THREE.PerspectiveCamera(160,
window.innerWidth/window.innerHeight, 0.1, 100 );
var renderer = new THREE.WebGLRenderer();
renderer.setSize( 300, 300 );

```

```
container.appendChild( renderer.domElement );
```

Після цього (у прикладі із відмальовкою точок) створюються масив вершин, встановлюються відстані між точками, точкам присвоюється матеріал та колір, а після цього вони додаються на сцену.

У частину коду, наведеній нижче, даний процес наведено із практичної точки зору. Завершальним елементом відмальовки сцени є встановлення камери у бажану позицію і виконання команди рендерингу.

```
var vertices = [];  
    for ( var i = 0; i < 3; i ++ ) {  
        var x = THREE.MathUtils.randFloatSpread( 500 );  
        var y = THREE.MathUtils.randFloatSpread( 300 );  
        var z = THREE.MathUtils.randFloatSpread( 0 );  
        vertices.push( x, y, z );  
    }  
var geometry = new THREE.BufferGeometry();  
geometry.setAttribute('position', new  
THREE.Float32BufferAttribute(vertices, 3 ) );  
var material = new THREE.PointsMaterial( { color: 0xff0000 } );  
var points = new THREE.Points( geometry, material );  
scene.add( points );  
camera.position.z = 6;  
renderer.render( scene, camera );
```

3.3 Приклад роботи

На наступних зображеннях наведені приклади роботи додатку. Інтерфейс тренажеру простий, інтуїтивно зрозумілий, не вимагає особливих навичок.

На головній сторінці користувач має можливість обрати мову інтерфейсу. На рисунках 3.6-3.8 представлено приклади встановлення різних мов проходження тесту.

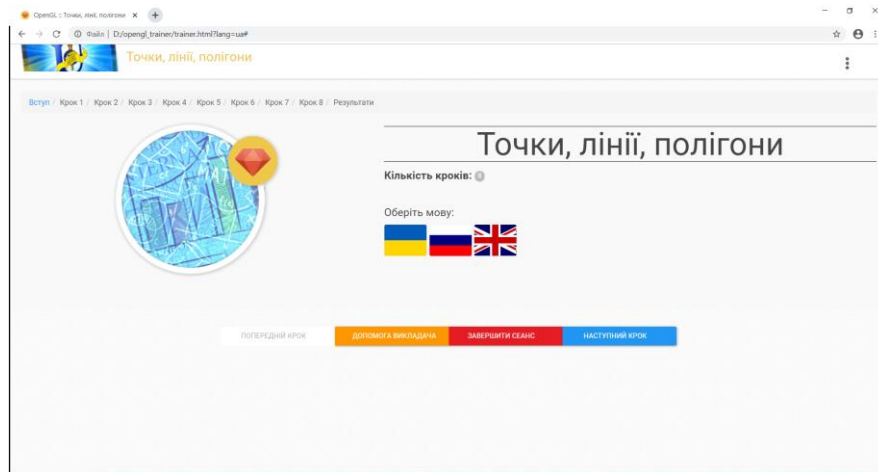


Рисунок 3.6 – Приклад встановлення української мови

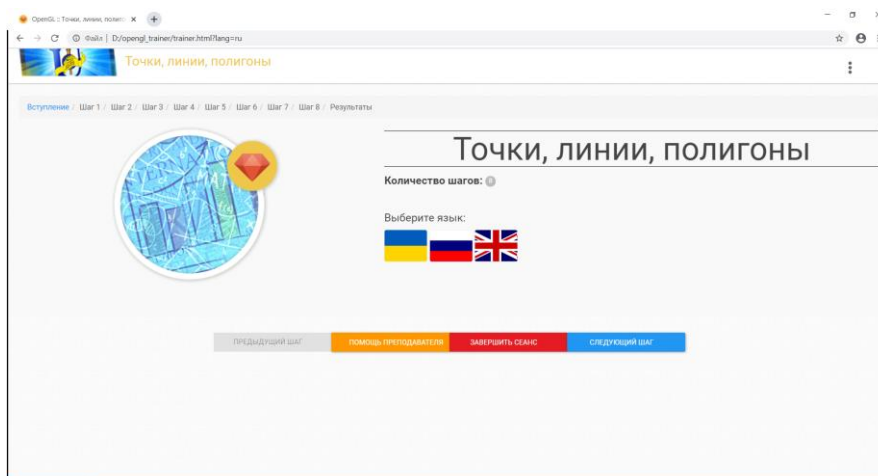


Рисунок 3.7– Приклад встановлення російської мови

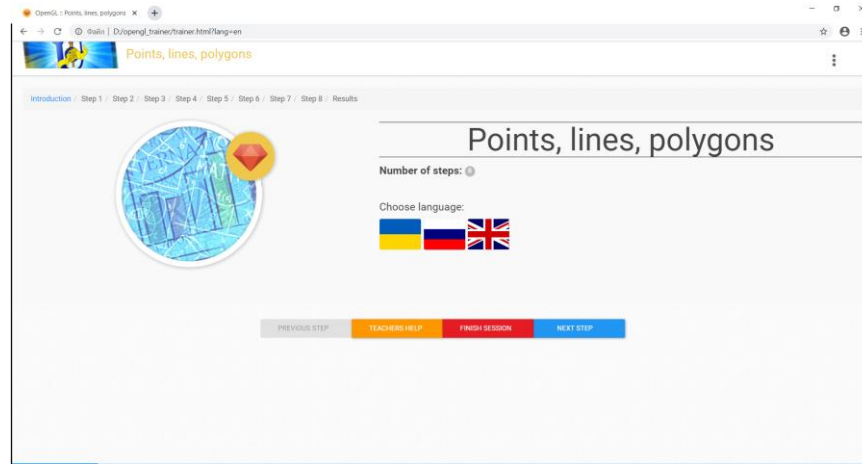


Рисунок 3.8 – Приклад встановлення англійської мови

Для зручності роботи користувача, а також втілення мінімалістичного дизайну було розроблено наступний дизайн сторінок тесту. Як було зазначено раніше, завдання мають тестову форму і на рисунку 3.9 представлено приклад завдання із вибором однієї правильної відповіді.

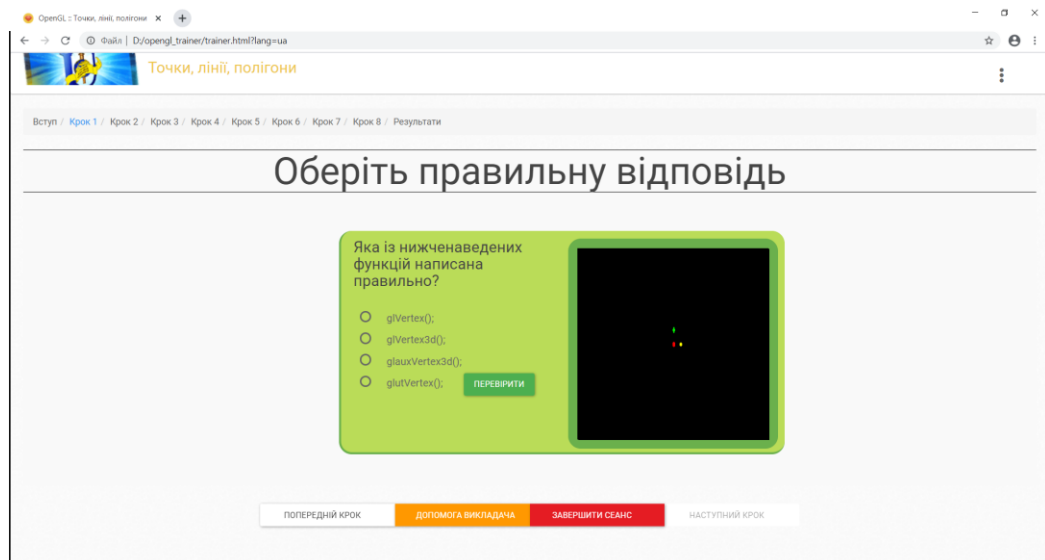


Рисунок 3.9 – Приклад тестового завдання

Важливо, щоб користувачеві надавалась допомога при проходженні тесту, тому при виборі невірної відповіді з'являється підкреслення червоним кольором і знімається кількість спроб.

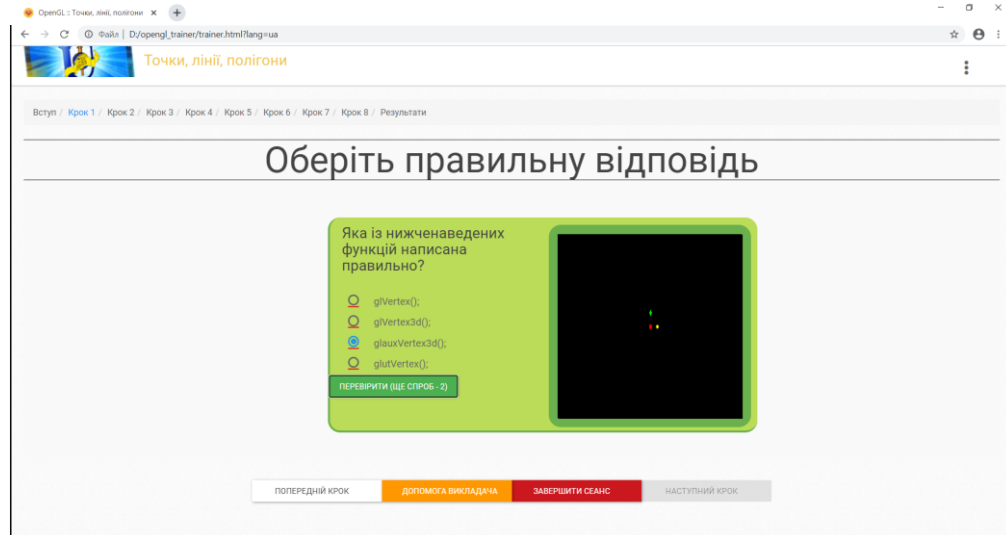


Рисунок 3.10 – Приклад встановлення невірної відповіді

У випадку, якщо користувач відповідає на питання вірно, йому стає доступним перехід до наступного кроку, а також відповідне повідомлення.

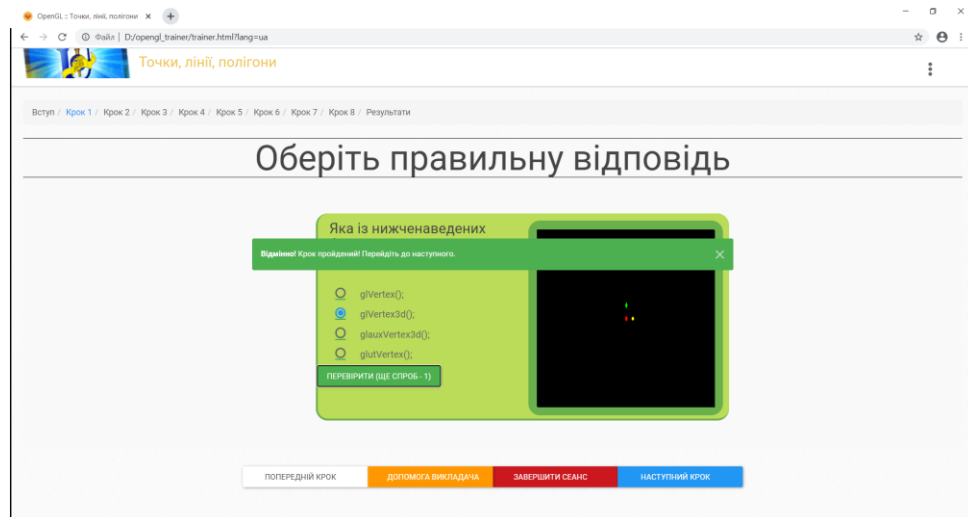


Рисунок 3.11 – Повідомлення про вірні результати

При проходженні всіх кроків тесту користувачеві представляється статистика його проходження: оцінка, час початку та закінчення роботи, а також період проходження.

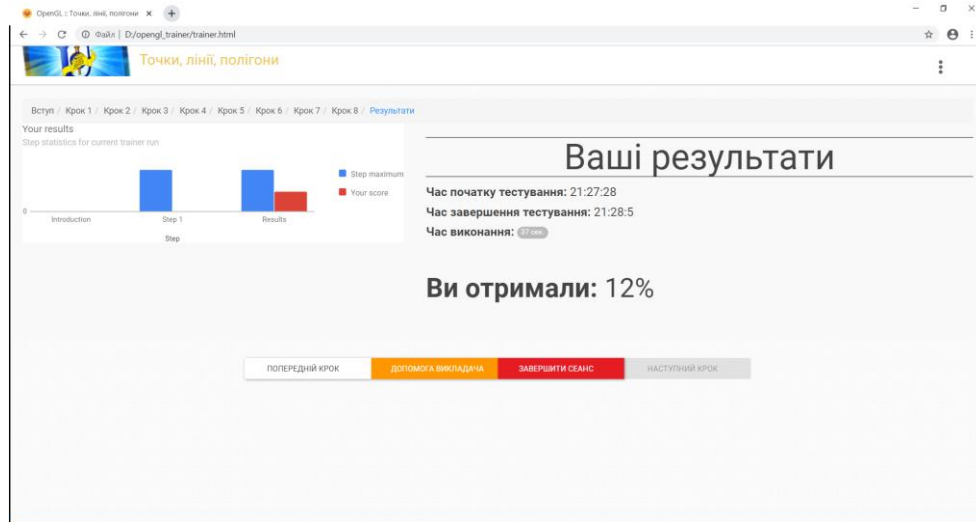


Рисунок 3.12 – Приклад проходження тесту

При натисканні на відповідну іконку у правому верхньому кутку сторінки можна переглянути інформацію про розробників та автора курсу.

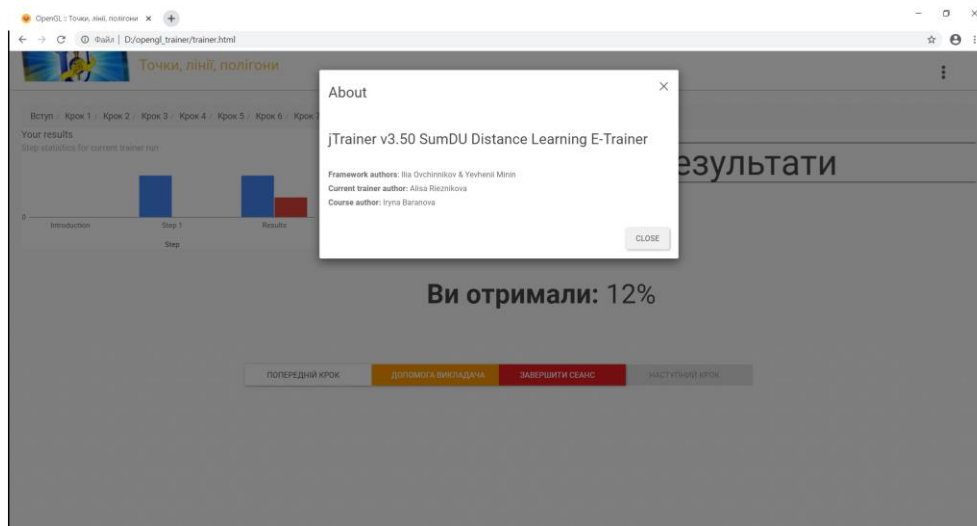


Рисунок 3.13 – Інформація про розробників

ВИСНОВКИ

У ході проектування та розробки веб-додатку кваліфікаційної роботи було досліджено доцільність розробки майбутнього додатку шляхом проведення аналізу предметної області. Було виявлено, що на даний момент існує достатньо багато сервісів для вивчення мов програмування, але вивчення OpenGL досі залишається поза цим колом. Таким чином було доведено, що майбутня розробка буде затребуваною. Також в результаті аналізу аналогів сформовано уявлення про роботу продукту.

Наступним кроком була проведена деталізація та чітко встановлені задачі, які будуть реалізовані у ході роботи над дипломним проектом. Було розроблено технічне завдання проекту. Даний документ дає чітку постановку завдань для виконання під час роботи над проектом, встановлює методи та інструменти розробки. Після цього було проведено моделювання майбутнього додатку.

Після проектування додатку була виконана його реалізація. У ході роботи було вивчено систему тестування, створену для ресурсу дистанційного навчання у Сумському державному університеті.

Результатом роботи став продукт, що буде інтегруватися у систему eLearning СумДУ та використовуватись студентами у якості тренажеру із дисципліни “Комп’ютерна графіка”. Даний проект може розвиватись у майбутньому шляхом внесення більшої кількості елементів інтерактивності.

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Руководство к своду знаний по управлению проектами (PMBOK Guide 2004). - М.: Project Management Institute.
2. Управление проектами с использованием Microsoft Project 2013: лабораторный практикум / Е. В. Кузьмин. - Самара: ПГУТИ, 2016. - 151 с.
3. Просницкий А. В. Самоучитель «Microsoft Project 2013. Управление проектами» [Электронный ресурс] / А. В. Просницкий. - Электрон. текстовые дан. - Киев: 2013. - Режим доступа: <http://leoconsulting.com.ua/library/nashi-samouchiteli>. - Загл. с экрана.
4. Стандарт по управлінню проектами British standard 6079-1:2000 PM.
5. Project management [Электронный ресурс] – Режим доступа до ресурсу: http://en.wikipedia.org/wiki/Project_management.
6. Шрайнер Д. OpenGL Redbook / Д. Шрайнер., 2015. – 936 с. – (Nine). – (ISBN: 0321552628).
7. OpenGL Programming Guide / S.Dave, G. Sellers, J. Kessenich, B. Licea-Kane., 2016. – 935 с. – (Nine).
8. threejs.org [Электронный ресурс] – Режим доступа до ресурсу: <https://threejs.org/>.
9. Codecademy [Электронный ресурс] – режим доступа: www.codecademy.com/learn
10. freeCodeCamp [Электронный ресурс] – режим доступа: www.freecodecamp.org
11. SoloLearn [Электронный ресурс] – режим доступа: www.sololearn.com
12. Почему геймификация работает и как её применить в обучении? [Электронный ресурс] – режим доступа: <https://bitly.su/D937Xi>

13. E-learning SSU [Электронный ресурс] – Режим доступа до ресурсу:
<https://elearning.sumdu.edu.ua/>
14. Какой язык программирования стоит выучить первым?
[Электронный ресурс] – режим доступа: <https://bitly.ru/ccrb59>
15. Wright Richard R. OpenGL SuperBible: Comprehensive Tutorial and Reference / R. Wright Richard, N. Haemel, G. Sellers., 2017. – 1008 с. – (Eighth). – (ISBN: 0-32-171261-7)
16. Что такое JavaScript? Презентация JS для начинающих
[Электронный ресурс] – режим доступа: <https://bit.ly/2TTxaVh>
17. Учебник HTML и CSS от Трепачева Дмитрия [Электронный ресурс]
– режим доступа: <https://bit.ly/2YMTFLV> 19.
18. Что такое CSS. Преимущества CSS. Версии CSS и их разработчики
[Электронный ресурс] – режим доступа: <https://bit.ly/2D4Hpx1> 20.
19. Что такое CSS, преимущества и недостатки [Электронный ресурс] –
режим доступа: <https://bit.ly/2IgKT3d>
20. Что такое JavaScript? [Электронный ресурс] – режим доступа:
<https://mzl.la/2LXeisy>
21. Rieznikova A., Vashchenko S. Web-based application for learning the basics of computer graphics programming. / A. Rieznikova, S. Vashchenko // Informatics, mathematics, automatics: materials and the program of the science-technology conference, Sumy, April 20-24, 2020 - Sumy: Sumy State Universitet, 2020. - P. 108.

ДОДАТОК А

**ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи «Web-додаток з вивчення основ
програмування комп'ютерної графіки»**

Суми 2020

1 Призначення й мета створення веб-додатку для вивчення основ комп'ютерної графіки

1.1 Призначення інформаційної системи

Web-додаток повинен представляти тренажер (набір вправ, пов'язаних спільною темою - робота з точками, лініями, полігонами) із дисципліни програмування комп'ютерної графіки, що викладається у Сумському державному університеті. Даний додаток призначений для використання під час навчального процесу студентами різних форм навчання для кращого засвоєння матеріалу, вивченого під час лекційних занять або шляхом самонавчання.

1.2 Мета створення інформаційної системи

Метою даного проекту є забезпечення студентів денної і заочної форми навчання тренажерами із дисципліни комп'ютерна графіка. Тести мають бути різноманітними, спрямованими на відпрацювання практичних навичок програмування комп'ютерної графіки, а також на розвинення просторового мислення шляхом виконання вправ.

1.3 Цільова аудиторія

У цільовій аудиторії інформаційної системи можна виділити наступні групи:

1. Студенти денної форми навчання.
2. Студенти заочної форми навчання.

У першу чергу додаток спрямований саме на використання студентами Сумського державного університету.

2 Вимоги до веб-додатку

2.1 Вимоги до структури й функціонування веб-додатку

Web-додаток повинен бути реалізований у вигляді набору сторінок, що будуть завантажені у мережу Інтернет як частина сервісу e-learning. Набір сторінок має бути послідовно пов'язаний між собою, переходячи від однієї вправи до наступної. Сторінки мають бути реалізовані на базі даного розробленого набору файлів.

2.2 Вимоги до персоналу

Для підтримки та експлуатації додатку від персоналу не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп'ютером і стандартним веб-браузером (наприклад, Google Chrome).

2.3 Вимоги до розмежування доступу

Інформація, розташовувана на ресурсі, є загальнодоступною.

Для даного тренажеру передбачено лише один вид користувачів - відвідувачі. **Відвідувачі** мають доступ до вправ тренажеру, що мають виконуватись у певній послідовності.

3 Основні вимоги

3.1 Структура Web-додатку

Web-додаток повинен складатися з наступних розділів:

- Головна сторінка – перелік розділів: точки, лінії, полігони; блок із обранням мови проходження тесту.
- Сторінка із вправою – частина із вибором варіантів, візуалізація коду і візуалізація результату роботи коду.

- Сторінка із результатами проходження тесту – оцінка, час проходження завдань.

3.2 Навігація

Інтерфейс тренажеру повинен забезпечувати інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів і сторінок. Навігаційні елементи повинні забезпечувати однозначне розуміння користувачем їх змісту: посилання на сторінки повинні бути мати заголовок, умовні позначки відповідати загальноприйнятим. Система повинна забезпечувати навігацію за всіма доступними користувачеві ресурсами і відображати відповідну інформацію. Для навігації повинна використовуватися система контент-меню. Меню повинне являти собою список гіперпосилань у верхній частині сторінки.

3.3 Наповнення веб-додатку (контент)

Сторінки розділів веб-додатку представляють собою заздалегідь сформовані вправи, які доступні користувачам.

3.4 Вимоги до функціональних можливостей

Система має надавати користувачеві можливість переглядати теоретичні відомості для виконання обраних вправ у відповідному розділі Теорія. Також після невірної виконання вправ користувачеві надаються повідомлення про невірний результат і візуальна, а після правильного – візуалізація проробленого у вигляді імітації коду на C++ OpenGL і відтворення графіки.

3.5 Функціональні можливості розділів

На головній сторінці будуть представлені наступні елементи:

- Назва тренажеру, кількість кроків тесту;
- Блок вибору мови ресурсу;
- Блок переходу до наступного кроку, допомоги викладача та завершення сеансу;

3.6 Загальні вимоги

Стиль зовнішнього вигляду додатку – сучасний і мінімалістичний. Таке рішення дозволяє студентам концентруватись на виконанні завдань.

У якості кольорової схеми рекомендується використовувати кольори головного сайту секції Інформаційних технологій проектування.

Оформлення додатку загалом має дотримуватись норм гармонійності як і з точки зору кольорової схеми, так і з погляду на розмітку, розташування блоків та функціональних зон.

Розташування елементів на сторінці із вправою схематично показано на рис. А.1.

Точки. Вправа 1.

Головна > Точки > Вправа 1

1. Оберіть правильний варіант:

- Radio button
 Selected radio button
 Radio button

2. Оберіть правильні варіанти:

- Checkbox
 Selected checkbox
 Checkbox

3. Оберіть правильний варіант:

- Radio button
 Selected radio button
 Radio button

The screenshot shows a web interface for an exercise. At the top, there's a navigation breadcrumb: 'Головна > Точки > Вправа 1'. Below that, there are three sections of radio button questions. The first section asks to choose the correct variant, with the second option 'Selected radio button' being selected. The second section asks to choose correct variants, with the second option 'Selected checkbox' being selected. The third section asks to choose the correct variant, with the second option 'Selected radio button' being selected. To the right of these questions is a visual problem area containing a small red dot and a larger green dot. Below the visual problem is a code editor with the text 'Імітація функцій(C++)'. At the bottom of the interface are two buttons: 'Виконати код!' and 'Наступна вправа'.

Рисунок А.1 – Типова сторінка із вправами

3.7 Типові навігаційні й інформаційні елементи

- Шапка сайту: назва ресурсу, логотип, про розробників
- Основне поле контенту: кількість та послідовність кроків, безпосередньо поле тесту, кнопки допомоги викладача та навігації між вправами;
- Футер, що показує прогрес у проходженні тесту.

3.8 Шапка сайту

Шапка сайту повинна містити логотип, а також перехід до перегляду інформації про розробника.

3.9 Основне поле контенту

Основне поле контенту повинне розташовуватися в центрі сторінки. У цьому полі відображається основний зміст обраного розділу.

Стильове оформлення матеріалів і їх елементів (посилань, заголовків, основного тексту, зображень, форм, таблиць і т.п.) повинне бути єдиним для всього веб-додатку.

4 Вимоги до видів забезпечення

4.1 Вимоги до інформаційного забезпечення

Реалізація сайту відбувається з використанням Three.js, PHP, HTML, CSS, JSON, Mustache.js і JavaScript.

4.2 Вимоги до лінгвістичного забезпечення

Веб-додаток повинен надавати користувачеві можливість обирати мову проходження тестів, тому слід забезпечити реалізацію українською, англійською та російською мовами.

4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 9.0 і вище, або Firefox 62.0 і вище, або Opera 66.0 і вище, або Chrome 78 і вище;
- Включена підтримка javascript і Flash.

4.4 Вимоги до апаратного забезпечення

Апаратне забезпечення клієнтської частини повинне забезпечувати підтримку програмного забезпечення клієнтської частини, зазначеного в п. 4.3

4.5 Склад і зміст робіт зі створення сайту

Докладний опис етапів роботи зі створення сайту наведено в табл.

А.1.

Таблиця А.1 – Етапи створення веб-додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Ознайомлення зі наданою структурою	2 дні
2	Точки: Проектування та реалізація сторінок розділу вправ Точки.	3 дні
3	Лінії: Проектування та реалізація сторінок розділу вправ Лінії.	3 дні
4	Полігони: Проектування та реалізація сторінок розділу вправ Полігони.	3 днів
	Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту	11

4.5 Вимоги до складу й змісту робіт із введення сайту в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного веб-додатку вимогам ТЗ і можливість його ефективної роботи, сторона Замовника має внести створений комплекс сторінок на ресурс e-learning та під'єднати їх до усього переліку тренажерів.

ДОДАТОК Б ПЛАНУВАННЯ РОБІТ

1 Планування робіт проекту (WBS)

WBS структура декомпозиції робіт є важливою частиною проектування процесу розробки програмного продукту. Декомпозиція робіт представляє ієрархічну декомпозицію робіт, що має бути виконаною командою проекту.

У даному випадку на вершині ієрархії стоїть задача реалізації проекту, а саме розробка веб-тренажеру. Цей рівень декомпозується на чотири рівні: встановлення вимог, проектування та дизайн, реалізація, документування.

Описання рівнів декомпозиції:

1. Встановлення вимог - збирання вимог, аналіз, специфікація вимог, пріорітизація вимог.
2. Проектування та дизайн включає в себе розробку дизайну інтерфейсу, роботу над вмістом уроків, а також обрання стеку технологій реалізації.
3. Реалізація продукту проекту – розробка бекенду та фронтенду. Проведення тестування продукту.
4. Документація - етап, на якому проводиться компіляція проектної та програмної документації, а також розробка документації для користувачів.

WBS-структура проекту наведена на рисунку Б.1.

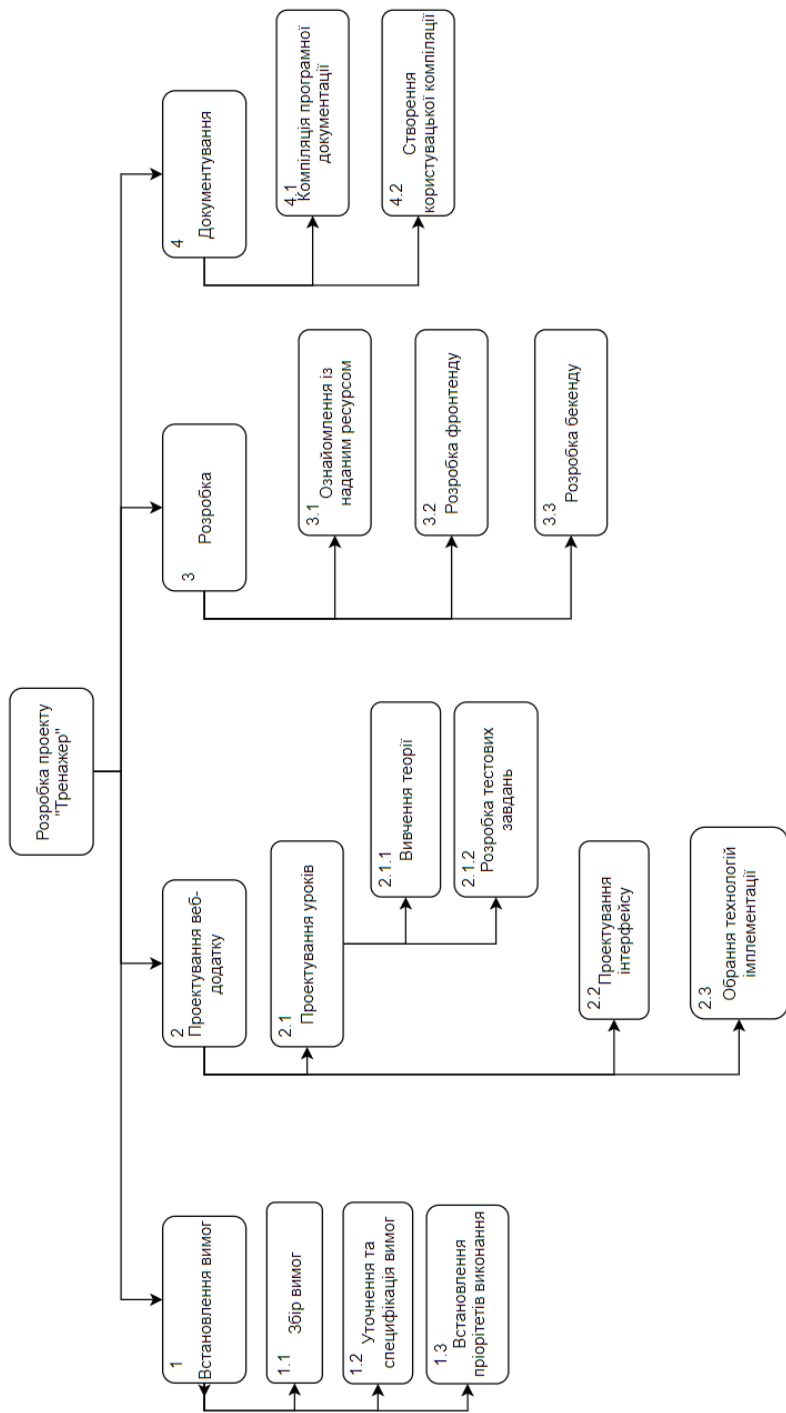


Рисунок Б.1 – WBS-структура проекту

2 Планування структури організації (OBS).

Організаційна структура представляє собою графічне відображення учасників проекту та відповідальних осіб, які задіяні в реалізації проекту. Верхній рівень OBS представляє всю команду проекту. На наступних рівнях проходить розбиття на під-команди, які стануть відповідальними за виконання певних елементарних робіт.

OBS структура наведена на рисунку Б.2.

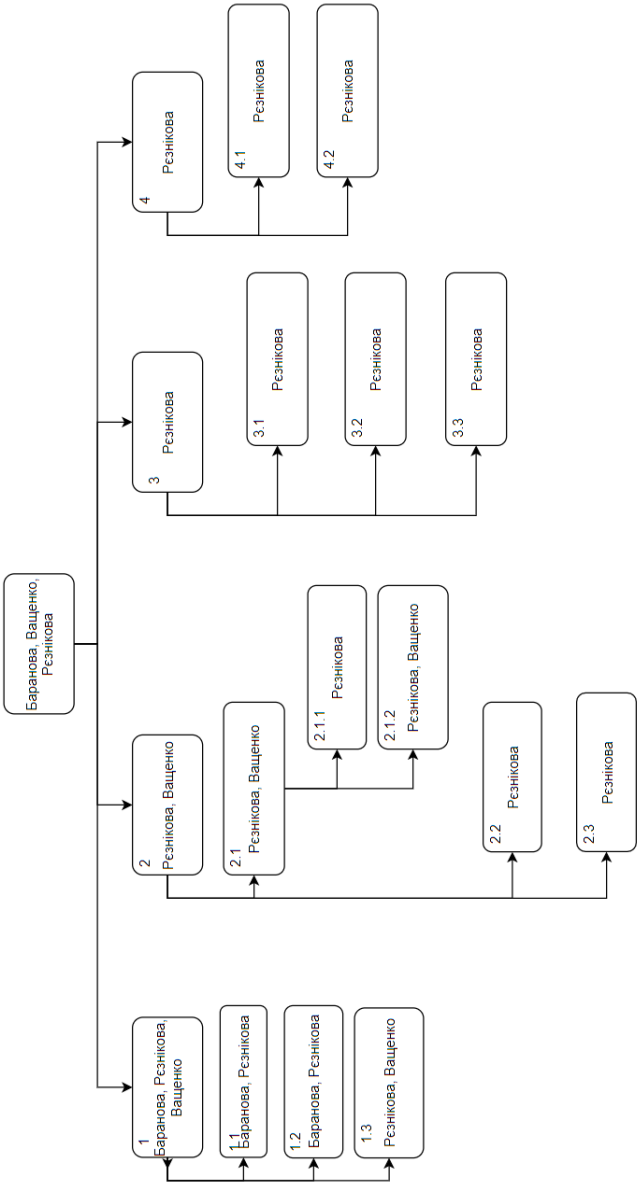


Рисунок Б.2 – OBS-структура проекту

У таблиці Б.1 представлено перелік стейкхолдерів.

Таблиця Б.1 - Представлення зацікавлених сторін

	Аліса Резнікова	Світлана Ващенко	Ірина Баранова
Контактна інформація	a.rznkva@gmail.com +380953345925	sveta@opm.sumdu.edu.ua	gloria@opm.sumdu.edu.ua
Внесок (1-5)	5	3	3
Вплив (1-5)	5	3	4
Що важливо для сторони?	Висока залученість інших зацікавлених сторін; Раціональний розподіл часу; Достатнє знання використовуваних технологій;	Висока продуктивність з боку команди розробників; Своєчасне спілкування і звітність;	Своєчасна доставка якісного продукту, відповідного вимогам.
Як зацікавлені сторони можуть внести свій вклад в проект?	Висока продуктивність у поєднанні з правильним управлінням часом.	Грамотна координація роботи команди; Надання консультацій та допомоги команді.	Складання чіткого списку вимог; Основні консультації.
Як зацікавлена особа може заблокувати проект?	Вимога додаткового часу для вивчення нових технологій; Недотримання термінів; Брак спілкування з іншими зацікавленими сторонами.	Низький рівень спілкування з командою.	Висувати суперечливі або неясні вимоги до продукту; Скоротити фінансування; Закрити проект.
Роль у проекті	Розробник	Скрам майстер	Власник

3 Побудова календарного графіку виконання ІТ-проекту.

Для того щоб мати чітку візуалізацію використання ресурсів та часу на певну послідовність робіт, найбільш доцільно сформувати діаграму Ганта. Дане представлення є реальним розподілом робіт зі сформованої структури декомпозиції робіт за календарними датами, тобто зрозуміло візуалізованим розкладом виконання робіт. Побудова діаграми відбувається наступним чином. На горизонтальній лінійці проводиться визначення термінів робіт у годинах/днях/тижнях тощо, а зліва вертикально розташовують найменування всіх робіт. Після цього в утвореному просторі відмічають роботи за тривалістю та встановлюють відповідні зв'язки.

Діаграма Ганта представлена на рисунку Б.3.

4 Управління ризиками.

Ризик за одним із визначень — це теоретично виникаюча подія, яка у випадку своєї появи негативно або позитивно впливає на проект. Управлінням ризиком називається процес реагування на події та зміни ризиків у процесі виконання проекту. Моніторинг ризиків включає контроль ризиків протягом всього життєвого циклу проекту. Якісний моніторинг ризиків забезпечує управління інформацією, яка допомагає приймати ефективні рішення до настання ризикових подій. Найбільш розповсюдженою характеристикою ризику є загроза або небезпека виникнення невдач у тій чи іншій діяльності.

Управління ризиками включає в себе наступне:

- 1) Ідентифікація ризиків
- 2) Оцінювання ризиків
- 3) Заходи реагування на ризики
- 4) Моніторинг ризиків

Відомі чотири основних методи управління ризиками: скасування, запобігання та контролювання, страхування та поглинання ризиків.

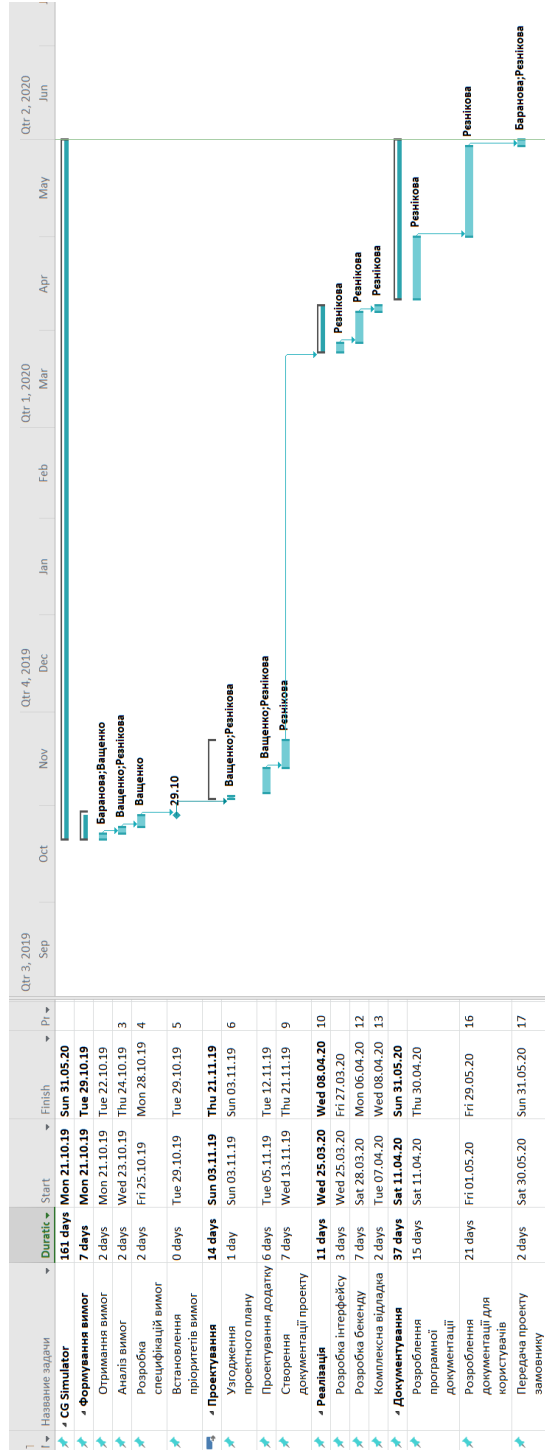


Рисунок Б.3 – Діаграма Ганта

Скасування ризику означає відмову від певної діяльності чи таку істотну (радикальну) її трансформацію, у результаті якої ризик зникає.

Поглинання ризику - це такий спосіб діяльності, коли при матеріалізації ризику збитки повністю несе його учасник (учасники). Цей метод управління ризиками застосовують тоді, коли можливість ризику невелика чи збитки в разі його настання неістотно впливають на учасників проектної діяльності.

У даному випадку використана модель пом'якшення та прийняття. Ризики у даному випадку визначаються як ті, які неможливо омайнути, або як ті, що потрібно певним чином передбачити та запобігти їм.

Таблиця Б.2 – Ризики проекту

№	Опис ризику	Вплив	Ймовірність	Рівень небезпеки	Стратегія
1	Низький рівень деталізації вимог	4	2		Пом'якшення впливу шляхом налаштування ефективної комунікації.
2	Низька кваліфікація розробників	5	4		Пом'якшення шляхом додавання більшої кількості часу на розробку.
3	Низький рівень порозуміння у команді.	2	2		Створення комфортних умов співпраці.
4	Неоптимальний розподіл часу роботи	3	4		Пом'якшення шляхом встановлення
5	Помилки розробки	4	4		Пом'якшення впливу шляхом вивчення спеціальної літератури.

Таблиця Б.4 – Ризики проекту (зведеня)

			4	5	2
		3		1	
P/I	1	2	3	4	5