

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: «Інтерактивний додаток для вивчення побудови геометричних примітивів з курсу "Комп'ютерна графіка"»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проектування»

Виконавець роботи: студент групи ІТ-62 Дуванов Семен Сергійович

Кваліфікаційна робота бакалавра

захищена на засіданні ЕК

з оцінкою

_____ «___» _____ 2020 р.

Науковий керівник

(підпис)

к.т.н., доц., Баранова І. В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без 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

Дуванов Семен Сергійович

1 Тема роботи Інтерактивний додаток для вивчення побудови геометричних примітивів з курсу "Комп'ютерна графіка"

керівник роботи Баранова Ірина Володимирівна, к.т.н., доцент,
затверджені наказом по університету від « 14 » травня 2020 р. № 0576-III

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

3 Вхідні дані до роботи технічне завдання на розробку веб-тренажеру

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, постановка задачі, моделювання роботи інтерактивного додатку, практична реалізація веб-тренажеру з курсу "Комп'ютерна графіка"

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність, постановка задачі, огляд аналогів, функціональні вимоги до веб-додатку, засоби реалізації, технології реалізації, структурно-функціональне моделювання, діаграма ВВ, структурна схема, практична реалізація, висновки (всього слайдів презентації)

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____ 01.10.2019 _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області, огляд існуючих аналогів	20.10.2019 – 25.10.2019	
2	Постановка задачі, функціональні вимоги до веб-додатку	30.10.2019 – 03.11.2019	
3	Вибір засобів і технології реалізації	04.03.2020 – 16.03.2020	
4	Планування робіт, структурно-функціональне моделювання	18.03.2020 – 23.03.2020	
5	Створення прототипу та верстання макету інтерфейсу	24.03.2020 – 04.04.2020	
6	Розробка завдань, переведення їх у веб-представлення, впровадження логіки наповнення вмісту веб-сторінки при переміщенні між розділами	06.04.2020 – 13.04.2020	
7	Створення блоку теорії	15.04.2020 – 21.04.2020	
8	Розробка механізму підрахування результатів	22.04.2020 – 07.05.2020	
9	Впровадження різних варіантів локалізації інтерфейсу	08.05.2020 – 14.05.2020	
10	Перевірка розробленої системи на помилки та їх виправлення	16.05.2020 – 20.05.2020	
11	Оформлення документації	20.05.2020 – 26.05.2020	

Студент _____

(підпис)

Дуванов С. С.

Керівник роботи _____

(підпис)

к.т.н., доц. Баранова І.В.

РЕФЕРАТ

Тема дипломного проекту: «Інтерактивний додаток для вивчення побудови геометричних примітивів з курсу "Комп'ютерна графіка"».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 17 найменувань, 3 додатків. Загальний обсяг роботи – 109 сторінок, в тому числі 41 сторінка основного тексту, 2 сторінки списку джерел та 66 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці інтерактивного додатку у вигляді веб-тренажеру з дисципліни «Комп'ютерна графіка» для студентів дистанційної форми навчання СумДУ.

В роботі проведено аналіз предметної області та існуючих аналогів, розглянуто їх недоліки та переваги. Обрано засоби та технології реалізації проекту. Також виконано моделювання роботи тренажеру з використанням методологій IDEF та UML, проведено планування робіт.

Розроблено структуру додатку, інтерфейс, заповнено контент сторінок теоретичних відомостей та розроблені необхідні завдання. Реалізовано можливість декількох спроб виконання та механізм підрахування результатів. Виконана локалізація інтерфейсу трьома мовами.

Результатом виконання проекту є розроблений інтерактивний веб-тренажер для вивчення побудови геометричних примітивів з курсу «Комп'ютерна графіка» для системи дистанційної освіти СумДУ.

Тренажер буде корисним студентам, що вивчають дисципліну «Комп'ютерна графіка» на початковому етапі.

Ключові слова: КОМП'ЮТЕРНА ГРАФІКА, OPENGL, ПРИМІТИВ, ЗАВДАННЯ, РЕЗУЛЬТАТИ, ТРЕНАЖЕР, ВЕБ-ДОДАТОК, JAVASCRIPT, BOOTSTRAP.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ОБЛАСТІ РОЗРОБКИ ТРЕНАЖЕРІВ ДЛЯ КОМП'ЮТЕРНОЇ ГРАФІКИ	8
1.1 Дослідження актуальності проблеми	8
1.2 Аналіз програмних продуктів - аналогів.....	9
1.3 Огляд технологій реалізації.....	11
1.4 Постановка задачі	12
2 МОДЕЛЮВАННЯ РОБОТИ ДОДАТКУ	14
2.1 Структурно-функціональне моделювання.....	14
2.2 UML моделювання веб-тренажеру	16
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО ДОДАТКУ.....	22
3.1 Структура веб-додатку.....	22
3.2 Реалізація компонентів тренажера.....	30
3.3 Тестування продукту.....	37
ВИСНОВКИ.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41
ДОДАТОК А.....	43
ДОДАТОК Б	52
ДОДАТОК В.....	63

ВСТУП

Багатьом людям іноді важко концентрувати свою увагу під час вивчення матеріалу, зв'язаного із програмуванням (та й не тільки), представленого в текстовому форматі. Людина може весь час відволікатися, й процес навчання розтягується надовго. Теоретичний матеріал в зручному для людського ока графічному представленні та елементи інтерактивності можуть спростити цей процес.

Актуальність роботи саме і полягає в тому, що розроблена система як раз міститиме всі ці елементи в собі. Веб-тренажер допоможе у засвоєнні матеріалу із основ тривимірної графіки у лиці специфікації OpenGL, яка використовується до сих пір. Процес засвоєння матеріалу шляхом виконання різних завдань із елементами керування може бути набагато цікавішим за просте читання текстової інформації із довідників, тому цей додаток матиме вигляд інтерактивного веб-інтерфейсу.

Об'єктом дослідження дипломного проекту є розроблення інтерактивного веб-додатку для вивчення основ побудови геометричних примітивів в специфікації OpenGL та розміщення його на дистанційному курсі університету СумДУ.

Предметом дослідження є інформаційна система, яка перевіряє якість знань із поставленої теми, містить в собі теоретичний матеріал та закріплює знання студента при багаторазовому виконанні завдань.

Метою дипломної роботи є створення інтерактивного додатку у вигляді веб-тренажеру для дистанційного курсу дисципліни «Комп'ютерна графіка», до якого матимуть доступ студенти університету СумДУ дистанційної форми навчання.

Основними задачами досягнення мети кваліфікаційної роботи є:

- провести аналіз предметної області, останніх досліджень та існуючих аналогів з теми роботи;

- розробити структуру додатку;
- створити візуально приємний веб-інтерфейс.
- розробити завдання та наповнити контент додатку;
- реалізувати механізм підрахування результатів проходження сеансу.

Практична цінність роботи полягає у спрощенні процесу вивчення або засвоєння інформації із специфікації OpenGL студентам завдяки наявності інтерактивності, перевірці якості знань із певної теми та (загалом) внесенні різноманіття в процес навчання.

1 АНАЛІЗ ОБЛАСТІ РОЗРОБКИ ТРЕНАЖЕРІВ ДЛЯ КОМП'ЮТЕРНОЇ ГРАФІКИ

1.1 Дослідження актуальності проблеми

Комп'ютерна графіка зараз стрімко розвивається і використовується в багатьох галузях, навіть в медицині [1]. Розповсюджене використання графіки в більших масштабах у майбутньому не підлягає сумніву. Одним із інструментів створення комп'ютерної графіки є специфікація OpenGL, для вивчення основ якої служитиме майбутній веб-додаток. Вивчення OpenGL допомагає зрозуміти, як саме представляється комп'ютерна графіка в програмному коді.

Веб-тренажер є одним із інструментів засвоєння інформації [2]. Завдяки компактній комбінації теоретичного матеріалу та практичних завдань в одній програмі, інформація може засвоюватися краще, аніж при читанні довідників і т.д. У випадку засвоєння інформації, зв'язаної із програмуванням, звичайним способом, людині знадобиться в наявності програмне забезпечення або середовище для перевірки роботи коду (наприклад Visual Studio).

Що стосується веб-тренажеру, для нього потрібна наявність лише браузера із доступом в мережу Інтернет, будь-хто може перейти за посиланням і відкрити веб-тренажер, не потребуючи встановлення додаткових програм. Також веб-тренажер може підказати де була допущена помилка при проходженні завдань (наприклад порядок викликання функцій, правильність значень параметрів функцій), чого іноді не може зробити компілятор.

В розглянутих дослідженнях вчених також наведені методики розробки тренажерів, які визначають порядок їх створення та необхідні компоненти, щоб тренажер відповідав вимогам сучасного освітнього процесу [3 - 5].

Перш за все тренажер дипломного проекту буде корисним студентам

дисципліни «Комп'ютерна графіка». Він вноситиме різноманіття в процес навчання та буде корисним на начальному етапі засвоєння інформації із теми «Примітиви», але насамкінець студентам все ж таки прийдеться встановлювати середовище розробки програмного забезпечення та використовувати отримані знання на реальній практиці.

1.2 Аналіз програмних продуктів - аналогів

Продукт дипломного проекту (веб-тренажер саме з комп'ютерної графіки) не має аналогів, оскільки він є вузькоспеціалізованим (тобто, переглядаючи простір Інтернету, не було знайдено веб-додатків, подібних до розроблюваного).

Виявлені додатки спеціалізуються більшістю в області навчання програмування, як наприклад інтерактивні додатки (рис.1.1-1.2) в рамках вивчення мов програмування під веб тощо [6].

Ці додатки не підходять для нашої мети, бо вони орієнтовані лише на мови програмування, тому не можуть бути використані. Специфікація OpenGL включає в себе унікальні, притаманні їй функції та класи, тому вона потребує окремої уваги.

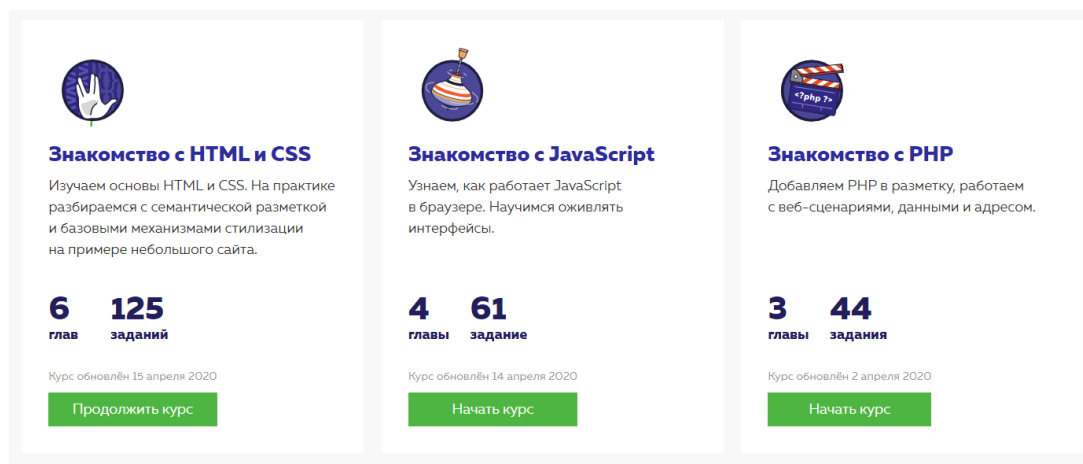


Рисунок 1.1 – Приклад тренажерів для вивчення мов веб-програмування

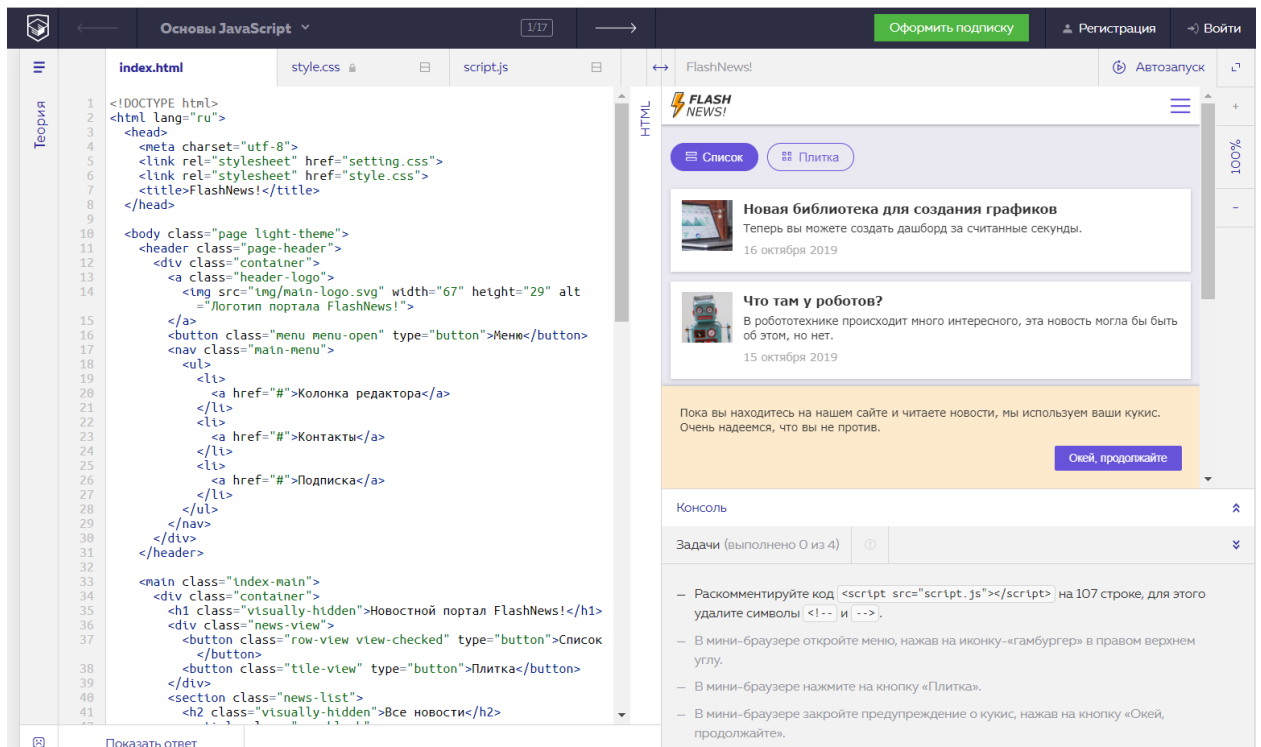


Рисунок 1.2 – Приклад тренажерів для вивчення мов веб-програмування

Існує також багато додатків для вивчення мов, як наприклад [7]. Вказані додатки також не можуть бути використані в якості аналогів, бо зовсім не враховують специфіку області знань «Комп'ютерна графіка» (рис.1.3).

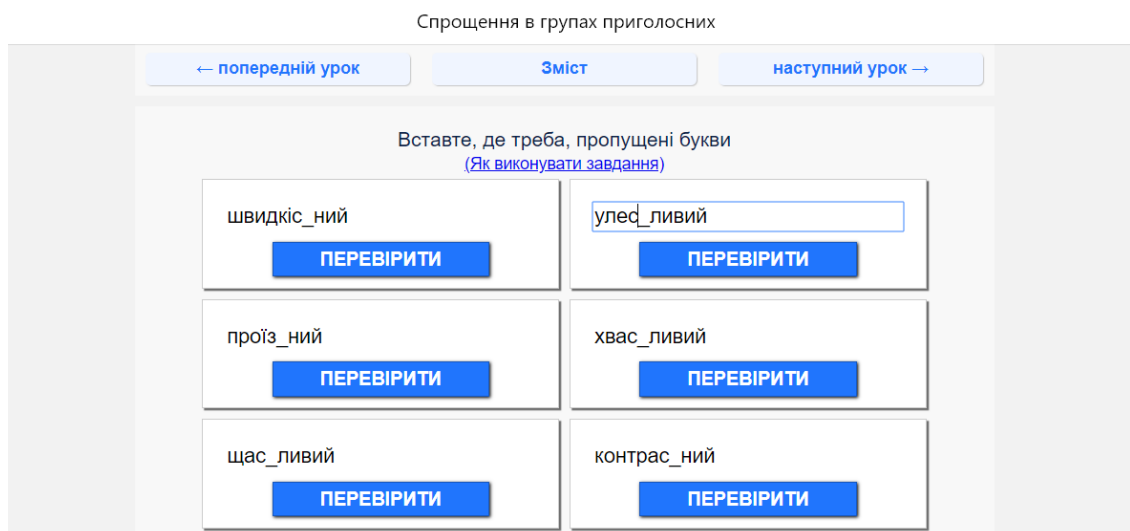


Рисунок 1.3 – Приклад веб-тренажеру для вивчення правопису української мови

Є також додатки у вигляді веб-квесту [8], але вони орієнтовані на школярів та надають знання лише з базових теоретичних понять комп'ютерної графіки, тому не можуть бути застосовані для закріплення практичних навичок студентів вищих навчальних закладів.

1.3 Огляд технологій реалізації

Згідно вимогам замовника, веб-додаток повинен бути реалізованим лише мовою JavaScript із-за обмежень серверної сторони дистанційного середовища університету. Тому вирішено, що веб-тренажер являтиме собою односторінковий веб-додаток без бази даних, що оперує інформацією, яка записана у файли формату JSON, розташованих у піддиректоріях тренажеру.

Односторінкові веб-додатки – це явище, яке стрімко розвивається сьогодні завдяки своїй швидкості роботи, доступності до різних типів пристроїв, що потребують лише наявності браузера [9]. На відміну від сайту, односторінковий веб-додаток складається лише з однієї сторінки, в яку завантажується інформація за допомогою скриптів (тобто скрипти вставляють блоки з HTML-кодом в головну і єдину сторінку та, за потреби, приховують непотрібну інформацію). Вся логіка функціонування прописується засобами мови виконуваних сценаріїв для веб-сайтів JavaScript.

Існують фреймворки та бібліотеки мови JavaScript, які сильно спрощують процес розробки веб-додатків – це React.js, Vue.js та Angular.js. Вони є найпопулярнішими зараз у використанні розробниками [10-12].

Також існує така технологія як Аjax, без якої не обходиться розробка будь-якого сучасного веб-додатку [9]. Вона дозволяє взаємодіяти із сервером у фоновому режимі та обмінюватися з ним даними, не оновлюючи сторінку цілком (цей механізм називається «асинхронним обміном даними») при спробі змінити вміст якогось елемента DOM-представлення. По суті це – обмін

даними із сервером засобами мови JavaScript. Ця технологія набагато пришвидшує роботу системи.

В технології Ajax дані, якими обмінюється клієнтська частина із серверною, можуть бути представлені у таких форматах: JSON (найбільш вживаний) та XML [13]. Такі широко використовувані сервіси, як Gmail та перекладач Google Translate, є прикладами односторінкового веб-додатку (або англійською SPA – Single Page Application) [9].

В нашому випадку база даних відсутня, тому AJAX-запити виконуватимуться лише для зчитування даних із файлів JSON, що розташовані в піддиректоріях тренажеру.

1.4 Постановка задачі

Мета проекту: розробити тренажер для дистанційного курсу навчання дисципліни «Комп'ютерна графіка» Сумського державного університету, що допомагатиме студентам у вивченні специфікації OpenGL та перевірятиме якість знань в темі «Примітиви».

Основними задачами для досягнення мети кваліфікаційної роботи є:

- продумати сценарій взаємодії користувача із додатком;
- проаналізувати дизайн існуючих різних веб-додатків, щоб зрозуміти композицію елементів на сторінці, підібрати до них гармонійні кольори та створити візуально приємний веб-інтерфейс;
- розробити структуру тренажера;
- розробити розділ із теорією, яка буде актуальною в час користування тренажером;
- розробити завдання та наповнити ними додаток;
- реалізувати механізм підрахування результатів проходження сеансу.

Розроблений продукт буде викладено на хостинг СумДУ адміністратором веб-тренажерів дистанційних курсів.

Для роботи продукту на комп'ютері потрібні лише дві вимоги:

- наявність браузеру;
- доступ до мережі Інтернет.

Після відкриття продукту в браузері користувач матиме змогу вибрати локалізації інтерфейсу. Далі слідує розділ із теорією, що буде актуальною під час проходження різноманітних завдань веб-тренажеру. Після проходження завдань користувач отримує результати проходження сеансу (час проходження та кількість правильних відповідей в процентах).

Повну інформацію про вимоги до програмного продукту наведено в технічному завданні (додаток А).

Інформацію щодо планування робіт та оцінки ризиків виконання проекту наведено в додатку Б.

2 МОДЕЛЮВАННЯ РОБОТИ ДОДАТКУ

2.1 Структурно-функціональне моделювання

Для побудови функціональної моделі системи застосуємо методологію IDEF0. Компонентами в методології є прямокутники або блоки (процес, функція, дія або задача) [14]. В кожен блок з різних сторін входять стрілки, які несуть такі ролі:

- вхід (входить в блок з лівої сторони) – вхідна інформація, що оброблятиметься процесом;
- управління (входить в блок зверху) – дані, що є обмеженням до процесу.
- вихід (виходить з блоку в праву сторону) – інформація, що є результатом виконання роботи або завершення процесу.
- механізм (входить в блок знизу) – ресурси або учасники процесу чи роботи.

Після побудови діаграми IDEF0 «Тренування та перевірка якості знань» було визначено такий перелік даних:

- вхідні дані: Перелік питань для засвоєння, Початок сеансу тренування;
- управління процесу: Теоретичний матеріал, Алгоритм роботи системи.
- вихідні дані: Кількість балів, Час проходження сеансу.
- механізми процесу: Студент, Викладач, Програмне забезпечення.

Діаграма IDEF0 представлена на рис. 2.1.

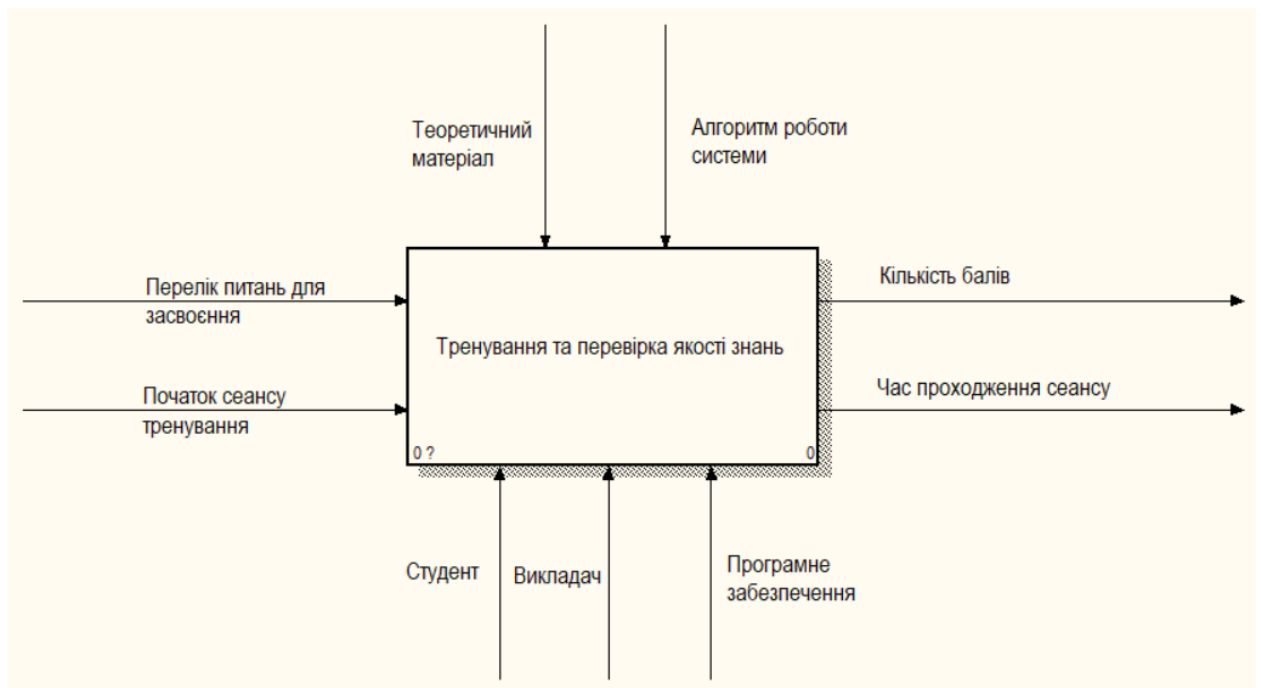


Рисунок 2.1 – Контекстна діаграма основного процесу

Далі процес «Тренування та перевірка якості знань» декомпозується на три процеси: «Перегляд теорії», «Виконання завдань», «Підрахування результатів».

Перший процес має такий перелік даних:

- вхідні дані: Початок сеансу тренування;
- управління процесу: Теоретичний матеріал, Алгоритм роботи системи;
- вихідні дані: Теорія переглянута;
- механізми процесу: Студент, Викладач, Програмне забезпечення.

Другий процес має такий перелік даних:

- вхідні дані: Теорія переглянута, Перелік питань для засвоєння;
- управління процесу: Теоретичний матеріал, Алгоритм роботи системи;
- вихідні дані: Завдання виконано;
- механізми процесу: Студент, Викладач, Програмне забезпечення.

Третій процес має такий перелік даних:

- вхідні дані: Завдання виконано;

- управління процесу: Алгоритм роботи системи;
- вихідні дані: Кількість балів, Час проходження сеансу;
- механізми процесу: Програмне забезпечення.

Діаграму декомпозиції процесу «Тренування та перевірка якості знань» зображено на рисунку 2.2.

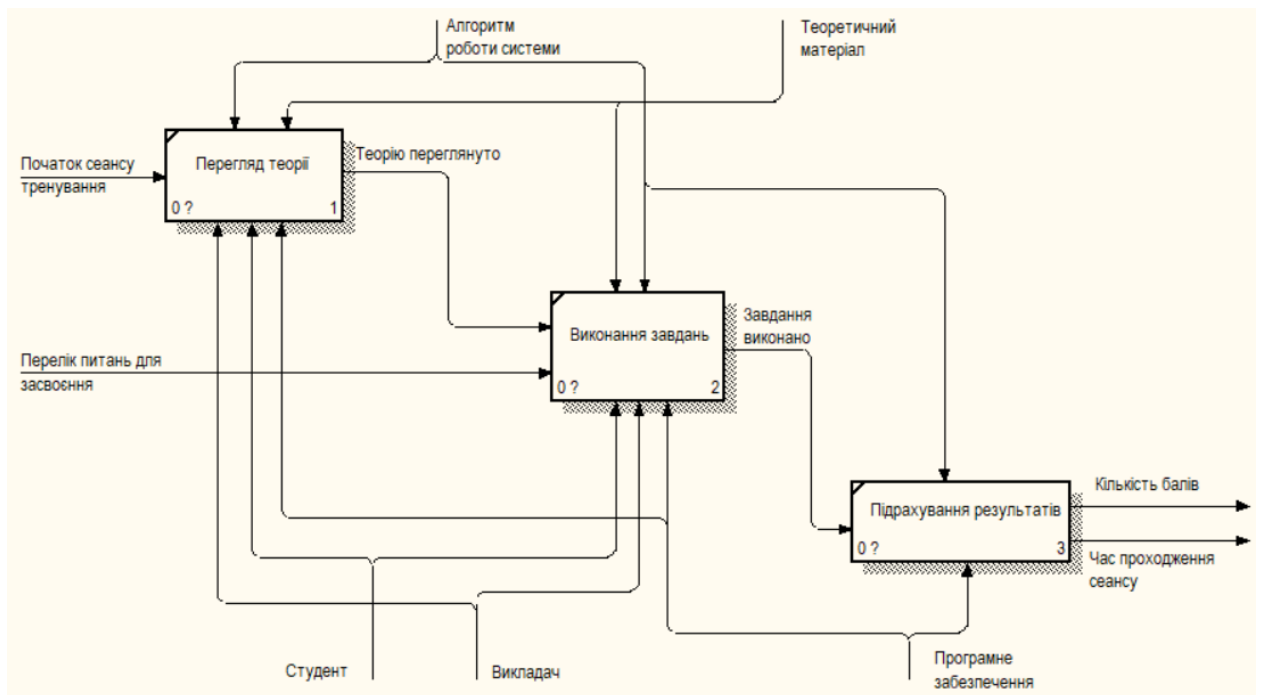


Рисунок 2.2 – Діаграма декомпозиції процесу «Тренування та перевірка якості знань»

2.2 UML моделювання веб-тренажеру

Для проектування системи використаємо UML – мову моделювання та графічного представлення функціонування програмних систем [15].

В першу чергу розробимо діаграму варіантів використання системи. Діаграму UseCase зображено на рис.2.3.

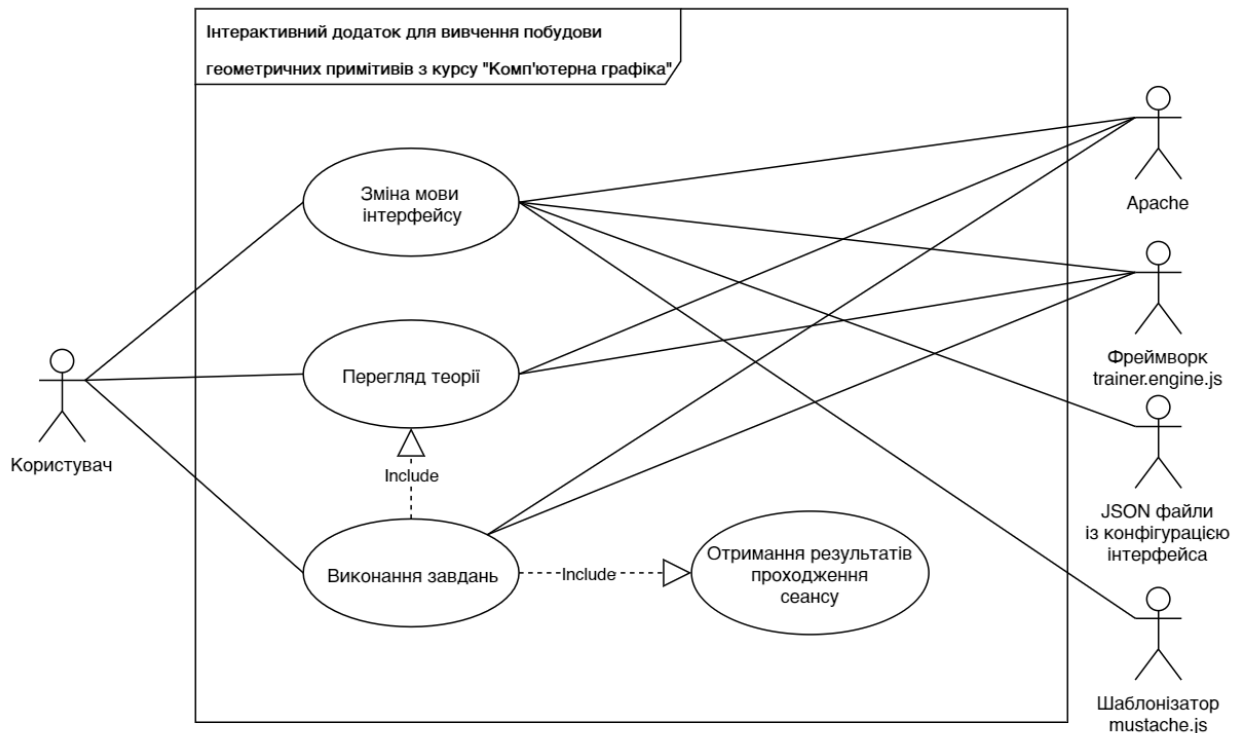


Рисунок 2.3 – Діаграма варіантів використання

В процесі моделювання діаграми було визначено, що акторами системи є користувач, веб-сервер Apache, фреймворк `trainer.engine.js`, JSON файли із конфігурацією інтерфейсу, шаблонізатор `mustache.js`, й система складається із чотирьох процесів. Зміна мови інтерфейсу – це можливість зміни мови локалізації на початку сеансу використання тренажера. Виконання завдань – це безпосередньо процес тренування. Перегляд теорії – процес перегляду довідкової інформації, як на початку сеансу, так і під час виконання завдань. Отримання результатів проходження сеансу – процес підрахування результатів.

Для чотирьох варіантів використання було створено діаграми послідовностей, що зображені на рисунку 2.4 – 2.7.

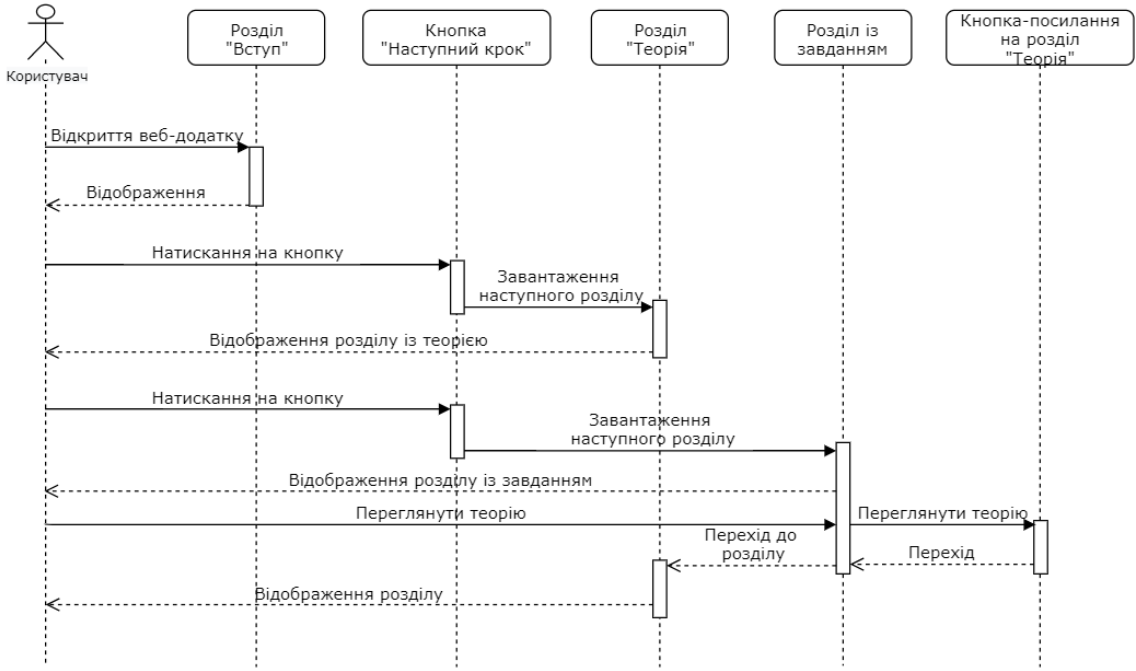


Рисунок 2.4 – Діаграма послідовності «Перегляд теорії»

Діаграми зображують хронологію послідовності виконання дій для отримання очікуваного результату.

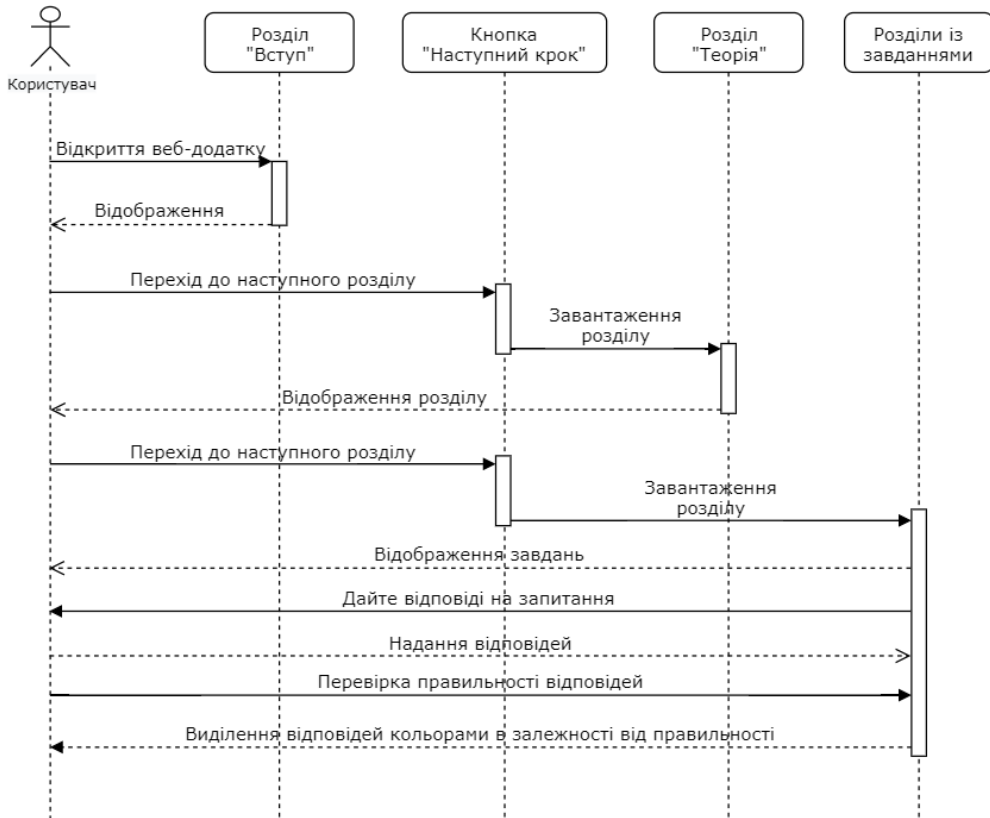


Рисунок 2.5 – Діаграма послідовності «Виконання завдань»

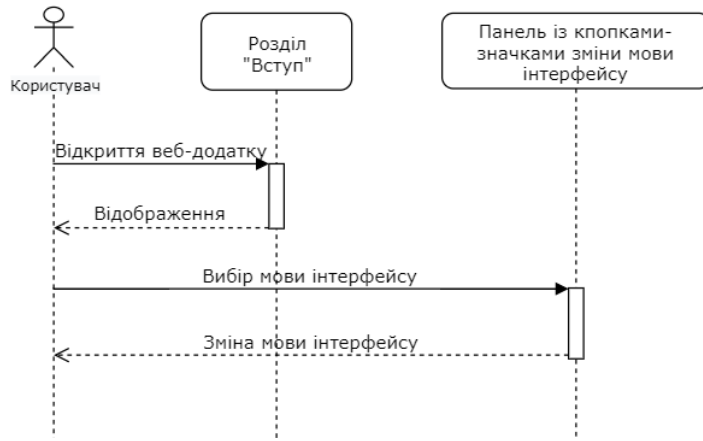


Рисунок 2.6 – Діаграма послідовності «Зміна мови інтерфейсу»

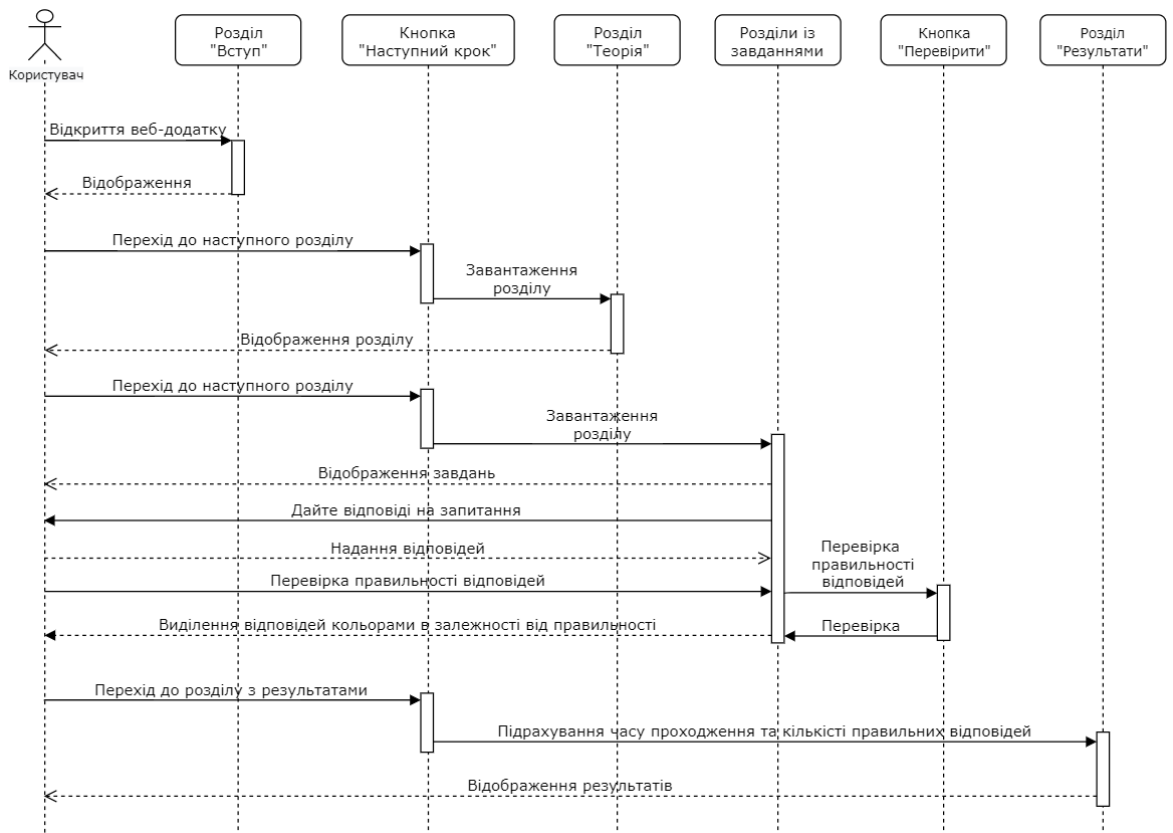


Рисунок 2.7 – Діаграма послідовності «Отримання результатів проходження сеансу»

Для представлення того, з яких компонентів складається система, створимо діаграму компонентів [16]. У нашому випадка вона показує ієрархічний взаємозв'язок файлів системи (переважно скриптів) та компонентів, що забезпечують функціонування системи.

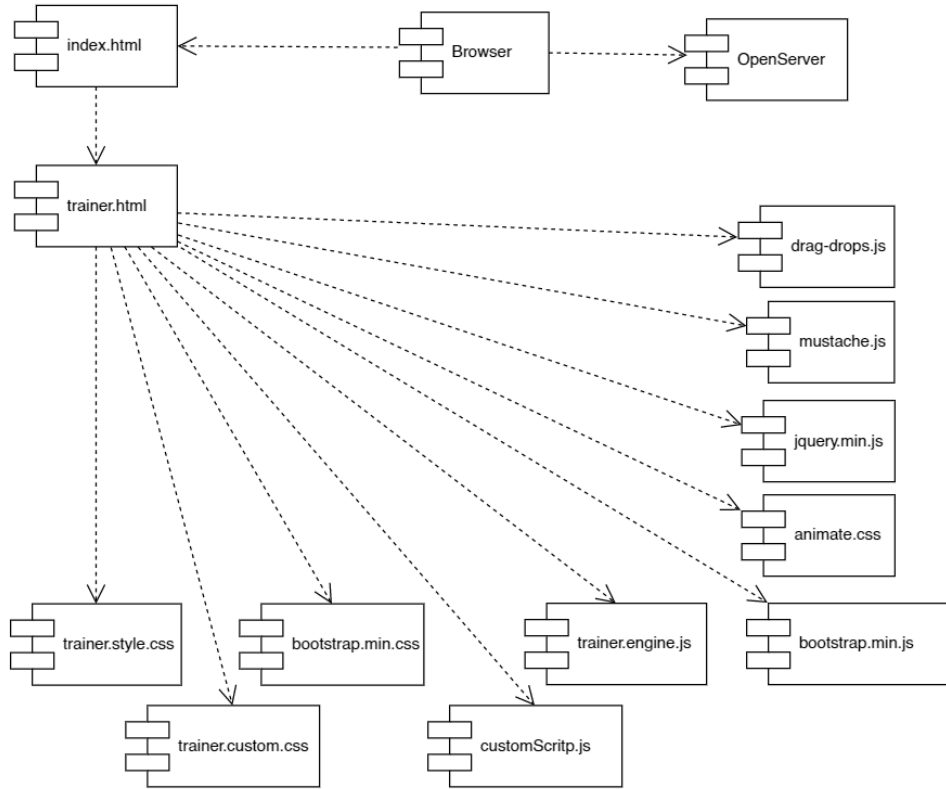


Рисунок 2.8 – Діаграма компонентів веб-тренажеру

Розробимо абстрактну діаграму класів нашої системи, яку наведено на рисунку 2.9.

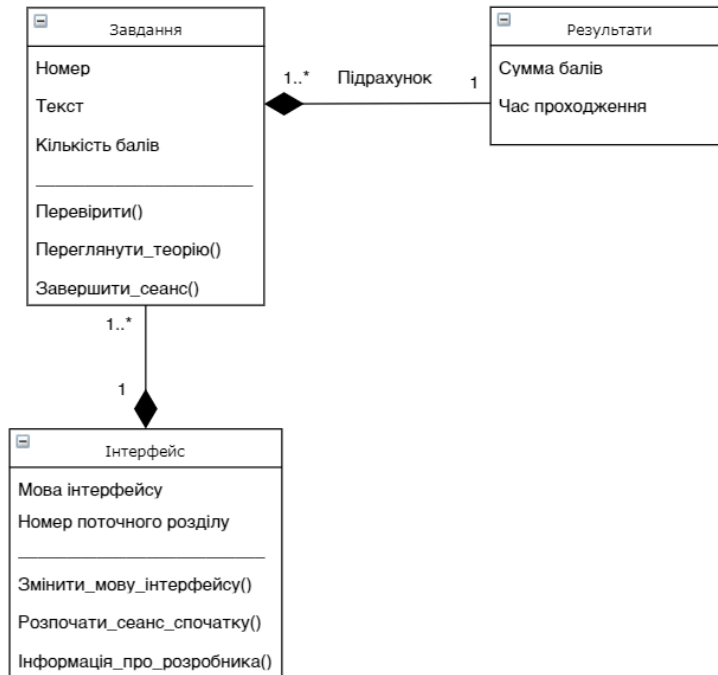


Рисунок 2.9 – Діаграма класів веб-тренажеру

В такому представленні система складається з трьох класів: завдання, інтерфейс, результати. Оскільки існування потенційних екземплярів всіх класів, крім «Інтерфейс», залежить від існування попередніх класів – то між ними стоїть зв'язок композиції (ромб розташовується на стороні класу-контейнеру).

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО ДОДАТКУ

3.1 Структура веб-додатку

Інтерактивний тренажер у вигляді веб-додатку було створено на основі фреймворку для тренажерів дистанційного середовища навчання університету СумДУ.

Стартове вікно містить логотип майбутнього тренажеру та кнопки зміни мови інтерфейсу (рис. 3.1).

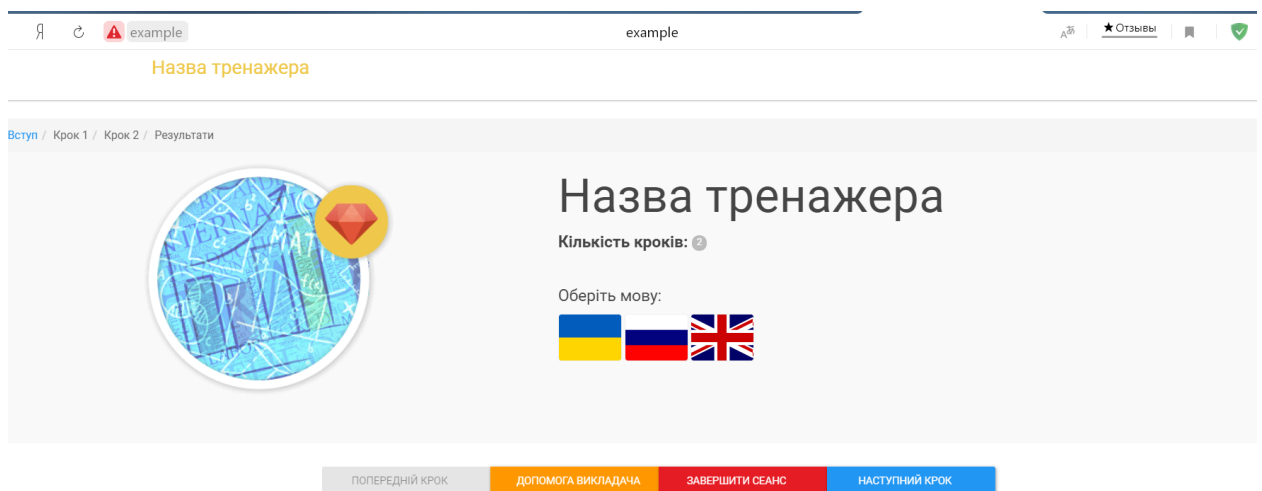


Рисунок 3.1 – Стартове вікно фреймворку (розділ «Вступ»)

Наступний розділ містить набір із прикладів елементів керування, з яких можна створювати завдання (рис. 3.2-3.3).

Назва тренажера

Вступ / [Крок 1](#) / Крок 2 / Результати

Перевірка елементів...

Оберіть коректний елемент зі списку:

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

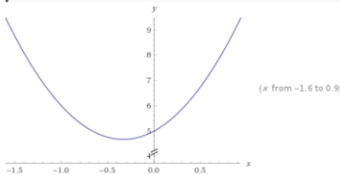
Перетягніть відповіді 1 та 4:

Введіть числа 1 АБО 4:

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

Рисунок 3.2 – Набір елементів керування для завдань (розділ «Крок1»)

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



Радіо №3
 Радіо №2
 Радіо №4 (правильний)
 Радіо №1

Введіть 4.5 або 4,5:

Правильна відповідь
 Неправильна відповідь

$G_{\mu\nu} = 8\pi G(T_{\mu\nu} + \rho\Lambda g_{\mu\nu})$

Рисунок 3.3 – Набір елементів керування для завдань (розділ «Крок1»)

Розділ «Результати», до якого користувач потрапляє після виконання всіх завдань, має наступний вигляд (рис. 3.4). Окрім текстового представлення результатів, він також містить діаграму Google Chart, що відображає статистику проходження завдань на кожному кроці (максимальна можлива кількість балів за крок та набрана кількість).

Фреймворк містить в собі скрипт *trainer-engine.js* (головний скрипт-рушій), що забезпечує перевірку правильності відповідей при натисканні на кнопку «Перевірити» та підрахування результатів проходження сеансу.

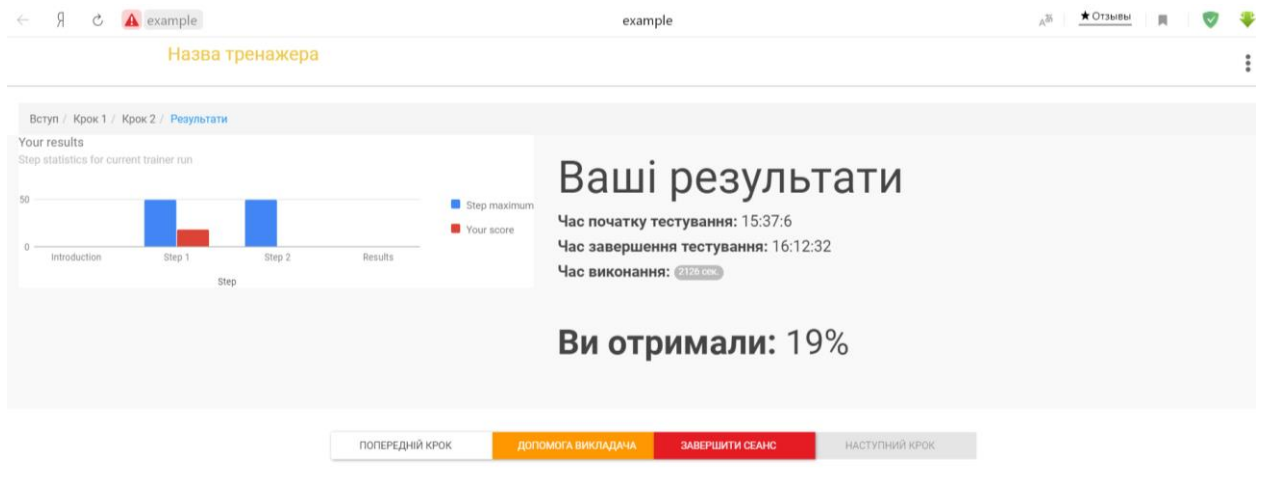


Рисунок 3.4 – Розрахунок результатів проходження сеансу

Основна директорія веб-додатку містить усі папки та файли, необхідні для його коректної роботи (рис. 3.5).

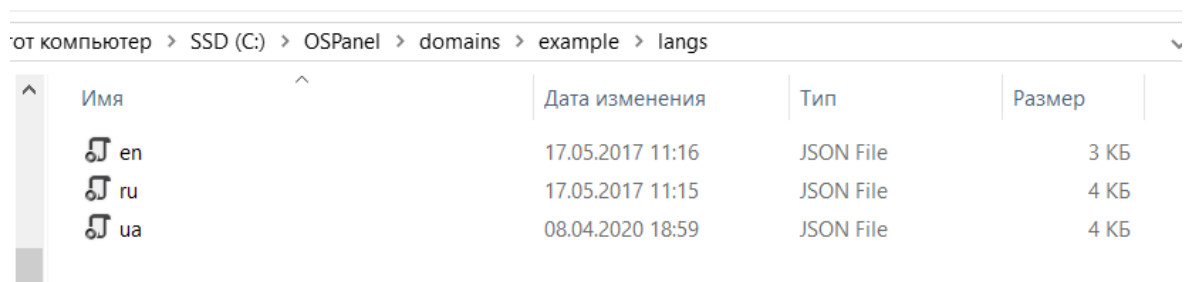
Имя	Дата изменения	Тип	Размер
cdn	08.04.2020 21:18	Папка с файлами	
css	17.01.2017 10:43	Папка с файлами	
fonts	17.01.2017 10:43	Папка с файлами	
img	17.05.2017 11:17	Папка с файлами	
langs	17.01.2017 10:43	Папка с файлами	
trainer	08.04.2020 18:52	Папка с файлами	
index	24.02.2016 19:14	Yandex Browser H...	1 КБ
report	24.02.2016 19:14	Yandex Browser H...	1 КБ
trainer	08.04.2020 21:18	Yandex Browser H...	8 КБ

Рисунок 3.5 – Директорія із файлами веб-тренажеру

Папка `cdn` містить JavaScript скрипти (`jquery.js`, `bootstrap.js`, `mustache.js`, `trainer.engine.js`), `css`-стилі `bootstrap`. Файли із цієї папки неможна змінювати, оскільки кожен тренажер дистанційного навчання на сервері використовує спільну директорію із стилями, бібліотеками, фреймворками, та головним скриптом-рушієм (фреймворком), тому, щоб уникнути проблем при перенесенні веб-додатка на сервер, у ході розробки стилі перевизначалися в новому файлі стилів – `css/trainer.custom.css`. Оскільки всі тренажери на сервері використовують стилі `bootstrap 3` зі спільної папки та була необхідність у

використанні елементів більш нових версій css-фреймворку, було створено файл *css/bootstrap-adds.css*, що містить деякі елементи із версії bootstrap 4.

Папка *langs* містить json-файли із всією інформацією (теоретична інформація, назви розділів, тексти питань, текст із повідомлень) для різних варіантів локалізації інтерфейсу (рис. 3.6-3.7). Json-файли виконують роль «бази даних» та зберігаються в піддиректоріях додатка. Головний скрипт додатка зчитує дані з цих файлів та підставляє їх в головний html-файл *trainer.html*.



Имя	Дата изменения	Тип	Размер
en	17.05.2017 11:16	JSON File	3 КБ
ru	17.05.2017 11:15	JSON File	4 КБ
ua	08.04.2020 18:59	JSON File	4 КБ

Рисунок 3.6 – Директорія із файлами локалізації інтерфейсу



```

1  {
2    "local": "Українська",
3    "lang": {
4      "TRAINER_NAME": "Назва тренажера",
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
28     "RADIO_TEXT_1": "Радіо №1",
29     "RADIO_TEXT_2": "Радіо №2",
30     "RADIO_TEXT_3": "Радіо №3",
31     "RADIO_TEXT_4": "Радіо №4 (правильний)",
32   }
  
```

Рисунок 3.7 – JSON-файл українського варіанту локалізації інтерфейсу

Папка *trainer* містить html-сторінки кожного розділу із супутнім javascript-кодом в папці «scripts», який забезпечує підтримку механізму перевірки відповідей, підрахування балів на кроці і т.д. (рис. 3.8-3.9).

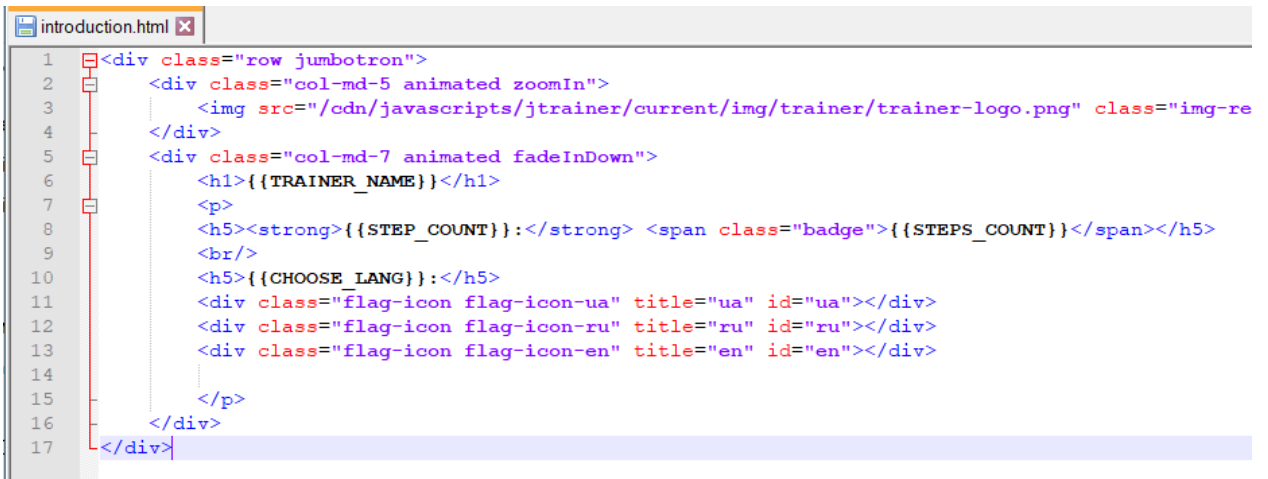
Имя	Дата изменения	Тип	Размер
scripts	08.04.2020 18:52	Папка с файлами	
settings	17.01.2017 10:43	Папка с файлами	
introduction	08.04.2020 21:18	Yandex Browser H...	1 КБ
results	31.01.2017 10:29	Yandex Browser H...	1 КБ
step1	05.02.2018 10:32	Yandex Browser H...	1 КБ
step2	08.04.2020 21:23	Yandex Browser H...	1 КБ

Рисунок 3.8 – Файли розділів тренажеру

Имя	Дата изменения	Тип	Размер
introduction	31.01.2017 10:14	файл JavaScript	1 КБ
results	08.04.2020 21:14	файл JavaScript	2 КБ
step1	05.02.2018 10:35	файл JavaScript	5 КБ
step2	08.04.2020 21:21	файл JavaScript	2 КБ

Рисунок 3.9 – Файли скриптів для розділів тренажеру

Інформація у файлах розділів тренажеру представлена у вигляді конструкції: `{{STEPS_COUNT}}` (рис. 3.10). У фігурних дужках розташовується назва змінної із JSON-файлу, значення якої підставлятиметься замість вище згаданої конструкції скриптом-шаблонізатором `mustache.js`. Скрипт-шаблонізатор допомагає спростити процес створення різних варіантів локалізації інтерфейсу (не треба створювати окремі файли для кожної локалізації, оскільки у нас є лише один файл, в який підставляються необхідні дані) та процес редагування текстових даних інтерфейсу (усі дані структурно та компактно розташовані в одному json-файлі необхідного варіанту локалізації).



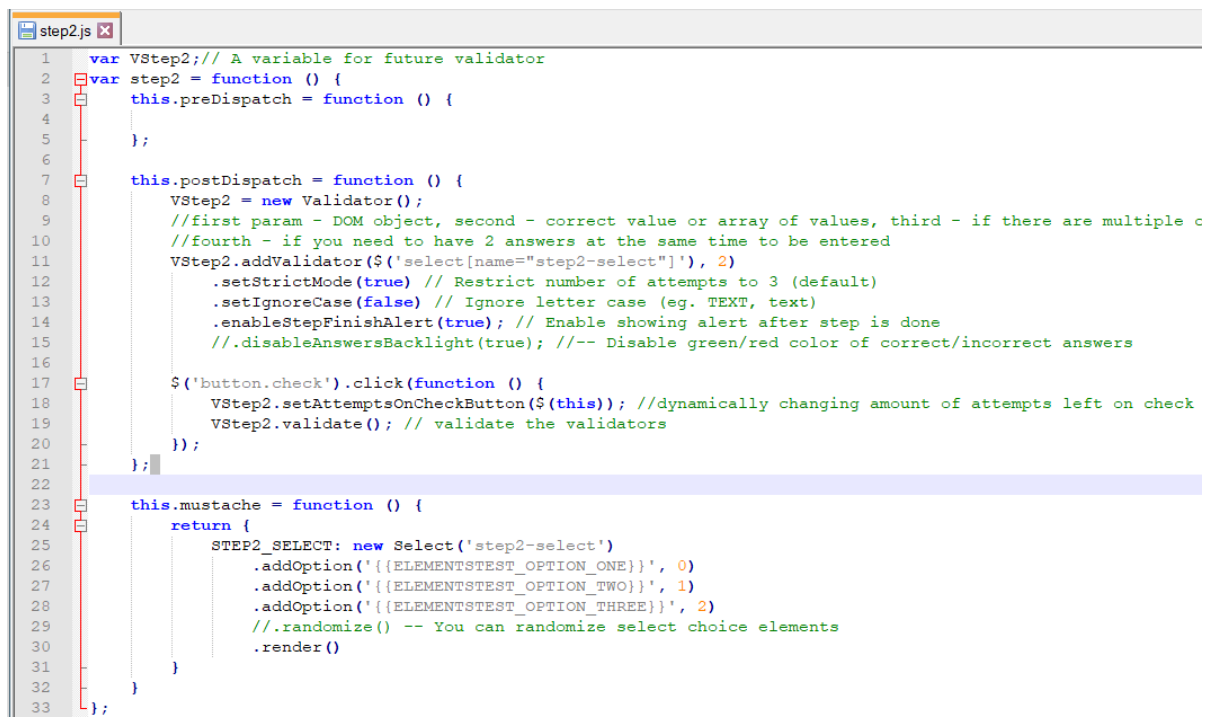
```

1 <div class="row jumbotron">
2   <div class="col-md-5 animated zoomIn">
3     
5   <div class="col-md-7 animated fadeInDown">
6     <h1>{{TRAINER_NAME}}</h1>
7     <p>
8     <h5><strong>{{STEP_COUNT}}</strong> <span class="badge">{{STEPS_COUNT}}</span></h5>
9     <br/>
10    <h5>{{CHOOSE_LANG}}:</h5>
11    <div class="flag-icon flag-icon-ua" title="ua" id="ua"></div>
12    <div class="flag-icon flag-icon-ru" title="ru" id="ru"></div>
13    <div class="flag-icon flag-icon-en" title="en" id="en"></div>
14
15    </p>
16  </div>
17 </div>

```

Рисунок 3.10 – HTML-файл розділу «Вступ»

JavaScript-файл розділу додає елементи керування в html-файл, налаштовує їх (вказуються правильні варіанти відповідей, налаштовуються властивості полів – наприклад ігнорування регістру букв та ін.) та містить функцію обробки подій для кнопки «Перевірити» (рис. 3.11). При переході на наступні розділи головний скрипт додає код html-файлу розділу та супутній скрипт в основний html-файл веб-додатку.



```

1 var VStep2; // A variable for future validator
2 var step2 = function () {
3   this.preDispatch = function () {
4
5   };
6
7   this.postDispatch = function () {
8     VStep2 = new Validator();
9     //first param - DOM object, second - correct value or array of values, third - if there are multiple c
10    //fourth - if you need to have 2 answers at the same time to be entered
11    VStep2.addValidator($('select[name="step2-select"]'), 2)
12    .setStrictMode(true) // Restrict number of attempts to 3 (default)
13    .setIgnoreCase(false) // Ignore letter case (eg. TEXT, text)
14    .enableStepFinishAlert(true); // Enable showing alert after step is done
15    //.disableAnswersBacklight(true); //-- Disable green/red color of correct/incorrect answers
16
17    $('button.check').click(function () {
18      VStep2.setAttemptsOnCheckButton($(this)); //dynamically changing amount of attempts left on check
19      VStep2.validate(); // validate the validators
20    });
21  };
22
23  this.mustache = function () {
24    return {
25      STEP2_SELECT: new Select('step2-select')
26        .addOption('{{ELEMENTSTEST_OPTION_ONE}}', 0)
27        .addOption('{{ELEMENTSTEST_OPTION_TWO}}', 1)
28        .addOption('{{ELEMENTSTEST_OPTION_THREE}}', 2)
29        // .randomize() -- You can randomize select choice elements
30        .render()
31    }
32  }
33 };

```

Рисунок 3.11 – JavaScript-файл розділу «Крок2»

Також в папці *scripts* розташовується файл *customScript.js*, який є надбудовою до основного скрипта фреймворка та містить декілька функцій маніпулювання вмістом інтерфейсу (активація, деактивація, приховування деяких посилань, кнопок) (рис. 3.12).



```

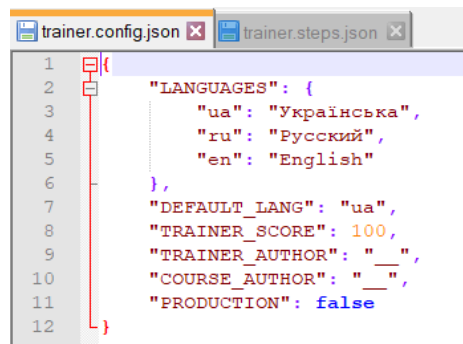
1  var enableTheoryLinks = function () {
2      $('a.book-style').removeClass('isDisabled');
3      $('a.book-style').attr('onclick', 'Rotator.switchStep(1)');
4  };
5
6  var disableTheoryLinks = function () {
7      $('a.book-style').addClass('isDisabled');
8      $('a.book-style').removeAttr('onclick');
9  };
10
11 var hideReturnFromTheoryButton= function () {
12     $('a.return-style').hide();
13 };
14
15 var showReturnFromTheoryButton= function () {
16     $('a.return-style').show();
17 };
18
19 var enableEndButton = function () {
20     $('a#endTrainer').removeAttr('disabled','disabled').attr('data-target','#endModal');
21     return this;
22 };
23
24 var disableEndButton = function () {
25     $('a#endTrainer').attr('disabled','disabled').removeAttr('data-target');
26     return this;
27 };
28
29 var disableInputCache = function () {
30     $('input[type="text"]').attr('autocomplete', 'off');
31 }

```

Рисунок 3.12 – Скрипт-надбудова

У папці «settings» розташовується два json-файли із налаштуваннями.

Перший містить загальну інформацію та налаштовує мову інтерфейсу за замовчуванням (рис. 3.13).



```

1  {
2      "LANGUAGES": {
3          "ua": "Українська",
4          "ru": "Русский",
5          "en": "English"
6      },
7      "DEFAULT_LANG": "ua",
8      "TRAINER_SCORE": 100,
9      "TRAINER_AUTHOR": "",
10     "COURSE_AUTHOR": "",
11     "PRODUCTION": false
12 }

```

Рисунок 3.13 – Файл конфігурації тренажеру

Другий – інформацію про назви файлів розділів та про максимальну кількість балів за розділ (рис. 3.14). У сумі всі бали розділів повинні давати 100.



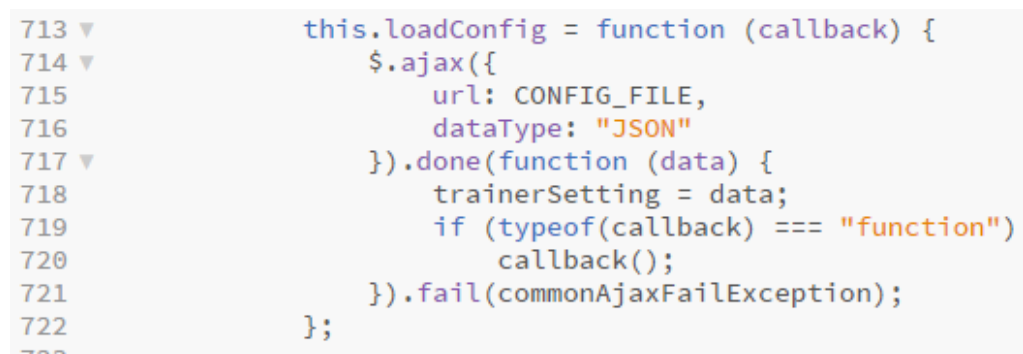
```

1  [
2  {
3      "filename"    : "introduction",
4      "score"      : 0,
5      "hasScript"  : true
6  },
7  {
8      "filename"    : "step1",
9      "score"      : 50,
10     "hasScript"   : true
11  },
12  {
13     "filename"    : "step2",
14     "score"      : 50,
15     "hasScript"   : true
16  },
17  {
18     "filename"    : "results",
19     "score"      : 0,
20     "hasScript"   : true
21  }
22 ]

```

Рисунок 3.14 – Файл конфігурації розділів тренажера

Усі дані з json-фалів зчитуються скриптом *trainer-engine.js* (головним скриптом) за допомогою аїах-запитів (рис. 3.15).



```

713  this.loadConfig = function (callback) {
714  $.ajax({
715      url: CONFIG_FILE,
716      dataType: "JSON"
717  }).done(function (data) {
718      trainerSetting = data;
719      if (typeof(callback) === "function")
720          callback();
721  }).fail(commonAjaxFailException);
722  };
723

```

Рисунок 3.15 – Приклад аїах-запиту в головному скрипті тренажера для зчитування даних конфігурації тренажера

3.2 Реалізація компонентів тренажера

Першою чергою було створено логотип тренажеру в програмі Adobe Illustrator та задано дані у файлах конфігурації тренажеру (назва тренажеру, інформація про розробника) (рис. 3.16). Також було перевизначено початкові CSS-стилі елементів інтерфейсу (змінився зовнішній вигляд кнопок, оформлення поточного розділу в панелі зверху та деяких інших елементів).

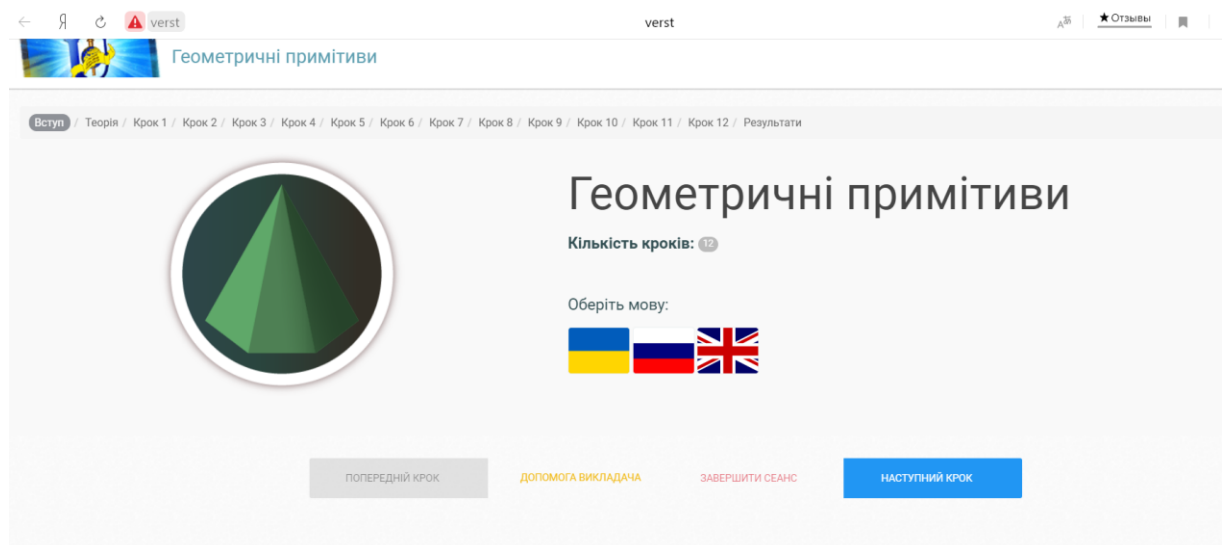


Рисунок 3.16 – Стартове вікно

Мова інтерфейсу за замовчуванням – «Українська» (рис. 3.17).

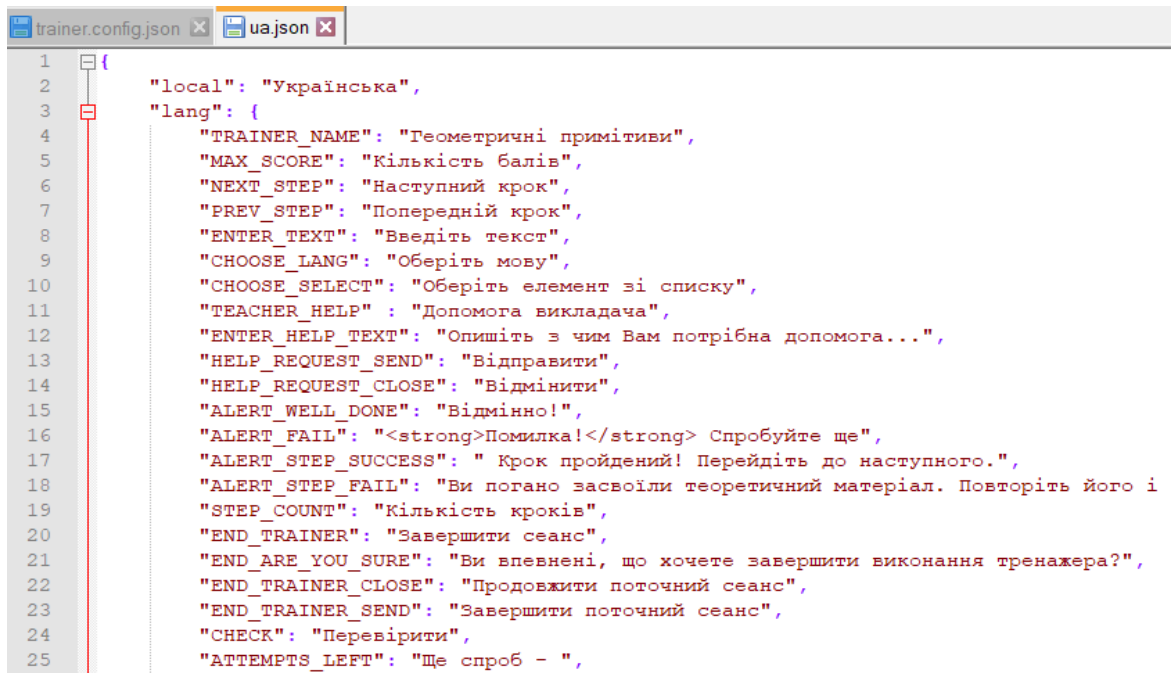
```

trainer.config.json
1  {
2    "LANGUAGES": {
3      "ua": "Українська",
4      "ru": "Русский",
5      "en": "English"
6    },
7    "DEFAULT_LANG": "ua",
8    "TRAINER_SCORE": 100,
9    "TRAINER_AUTHOR": "Semen Duvanov",
10   "COURSE_AUTHOR": "Irina Baranova",
11   "PRODUCTION": false
12 }

```

Рисунок 3.17 – Файл конфігурації

Уході розробки веб-додатку спочатку було реалізовано український варіант інтерфейсу, а потім – інші (рис. 3.18).



```

1 {
2   "local": "Українська",
3   "lang": {
4     "TRAINER_NAME": "Геометричні примітиви",
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    "ALERT_WELL_DONE": "Відмінно!",
16    "ALERT_FAIL": "<strong>Помилка!</strong> Спробуйте ще",
17    "ALERT_STEP_SUCCESS": " Крок пройдений! Перейдіть до наступного.",
18    "ALERT_STEP_FAIL": "Ви погано засвоїли теоретичний матеріал. Повторіть його і",
19    "STEP_COUNT": "Кількість кроків",
20    "END_TRAINER": "Завершити сеанс",
21    "END_ARE_YOU_SURE": "Ви впевнені, що хочете завершити виконання тренажера?",
22    "END_TRAINER_CLOSE": "Продовжити поточний сеанс",
23    "END_TRAINER_SEND": "Завершити поточний сеанс",
24    "CHECK": "Перевірити",
25    "ATTEMPTS_LEFT": "Ще спроб - ",

```

Рисунок 3.18 – Дані українського варіанту локалізації

Наступний розділ («Теорія») містить блок, що є модифікацією «bootstrap» елемента «nav-bar» або «навігаційна панель». При натисканні на посилання змінюються блоки з темами теорії.

Блок містить теорію із шести зв'язаних тем, що знадобляться під час виконання завдань.

Майже кожен рисунок із фігурами теоретичного блоку містить підказки з кодом функцій OpenGL та її параметрами, які задавались при побудові примітиву в середовищі розробки Visual Studio 2010 (рис. 3.19).



Рисунок 3.19 – Розділ «Теорія»

Усі зображення з фігурами в теорії та наступних завданнях були створені шляхом компіляції коду функцій OpenGL в середовищі Visual Studio 2010 C++ проекту (рис. 3.20).

```

void CALLBACK display(void)
{
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );

    glClearColor(0.14, 0.15, 0.23, 1.0f);

    glColor3d(0.8,0.1,1);

    auxWireIcosahedron(5); // 1-task

    //glutSolidSphere(4, 7, 8); 2-task

    //glPushMatrix(); 3-task

    //auxSolidCube(4);

    //glPopMatrix();

```

Рисунок 3.20 – Приклад OpenGL коду

З наступного розділу починаються завдання, що мають назви «Крок+ номер кроку». В кожному завданні містяться посилання на розділ «Теорія», щоб користувач міг читати теорію та виконувати завдання одночасно, тим самим закріплюючи знання (рис. 3.21). Якщо перейти до

розділу «Теорія», то з'явиться посилання у вигляді стрілки для зручного переходу назад на розділ із завданням (рис. 3.22).

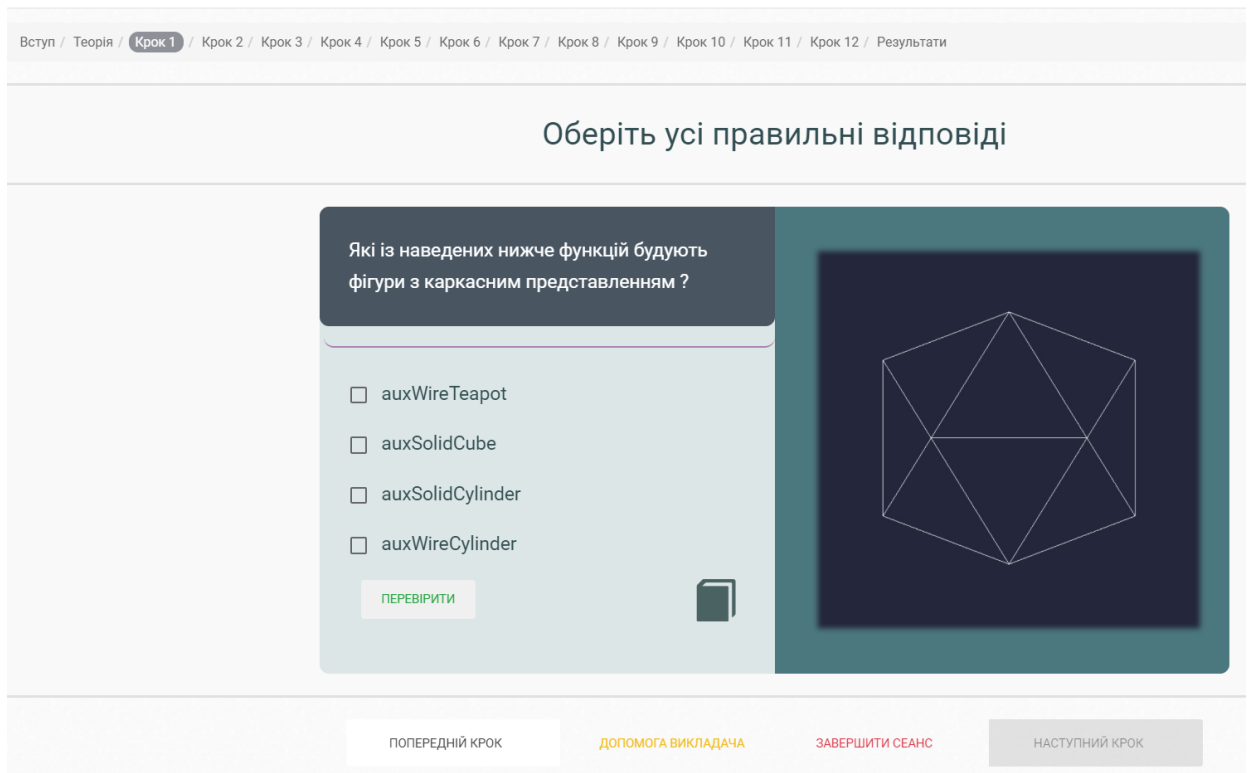


Рисунок 3.21 – Розділ «Крок1»

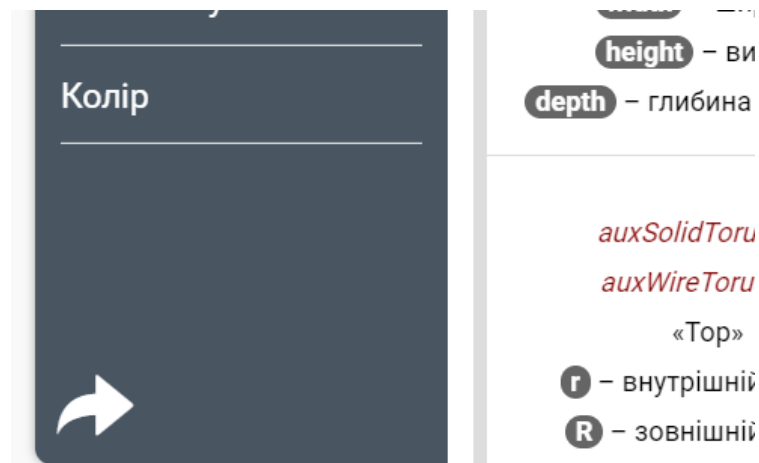


Рисунок 3.22 – Значок повернення назад в розділі «Теорія»

Далі слідує ще одинадцять різноманітних кроків із завданнями (рис. 3.23).

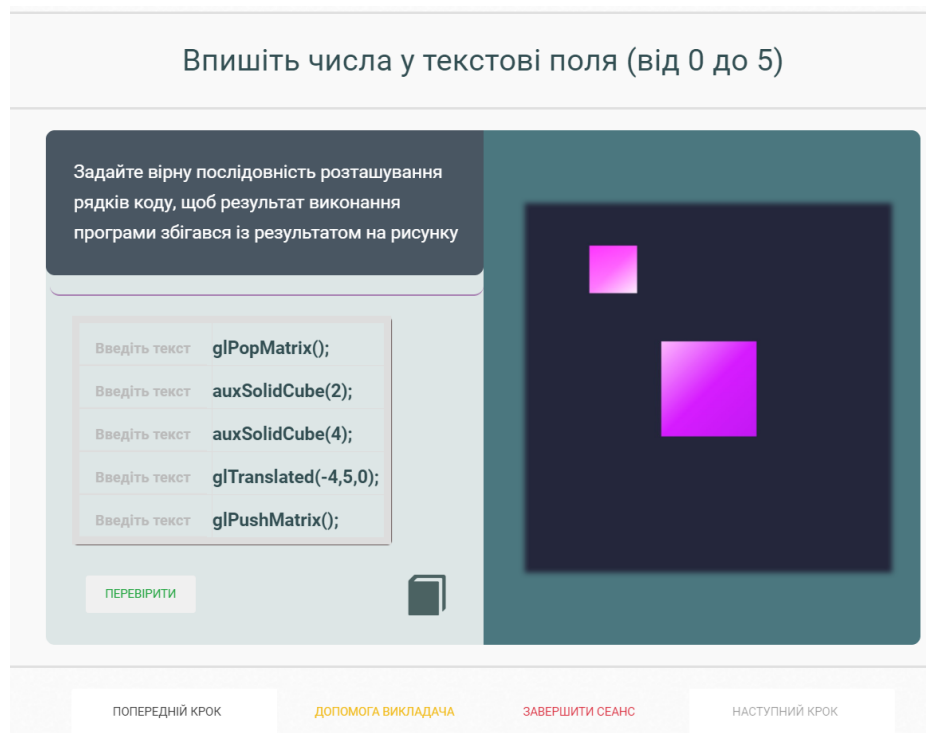


Рисунок 3.23 – Завдання з наступних розділів (або кроків)

При неправильній відповіді на запитання у перші два рази, на кожен раз з'являється повідомлення про помилку (рис. 3.24).



Рисунок 3.24 – Повідомлення про неправильну відповідь на запитання

При неправильній відповіді в третій раз з'являється повідомлення про завершення поточного сеансу тренажеру (рис. 3.25) та користувача перекидає на початок тренажеру (сеанс починається заново).

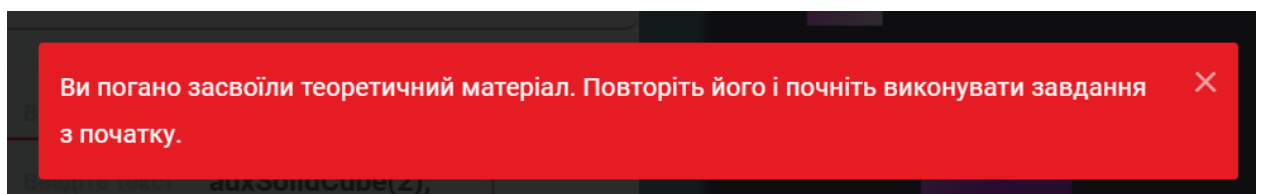


Рисунок 3.25 – Повідомлення про неправильну відповідь на запитання

З будь-якого кроку користувач може завершити сеанс і він буде направлений в розділ «Результати». В цьому вікні користувач зможе побачити результат поточних набраних балів за пройдені кроки, але не зможе переглядати попередні розділи. З цього вікна можливо лише розпочати сеанс заново. При цьому кнопка «Завершити сеанс» змінюється на кнопку «Розпочати знову» (рис. 3.26).

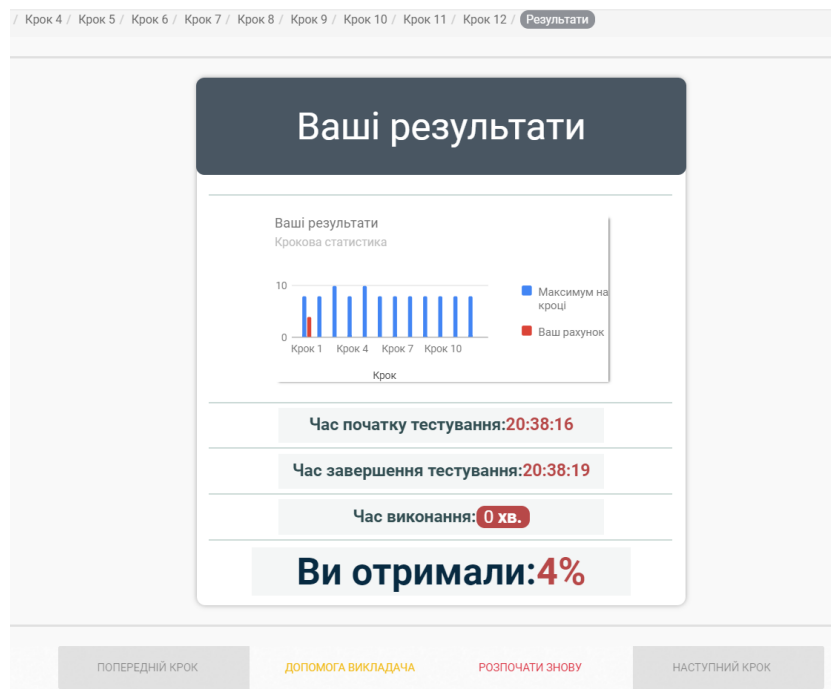


Рисунок 3.26 – Розділ «Результати» після передчасного завершення сеансу

При успішному проходженні всіх кроків із завданнями і переходу в розділ «Результати» кнопка «Попередній крок» буде доступною і користувач зможе передивитися пройдені завдання або теорію. Формат часу виконання було змінено відносно шаблону фреймворку із секунд на хвилини (рис. 3.27).

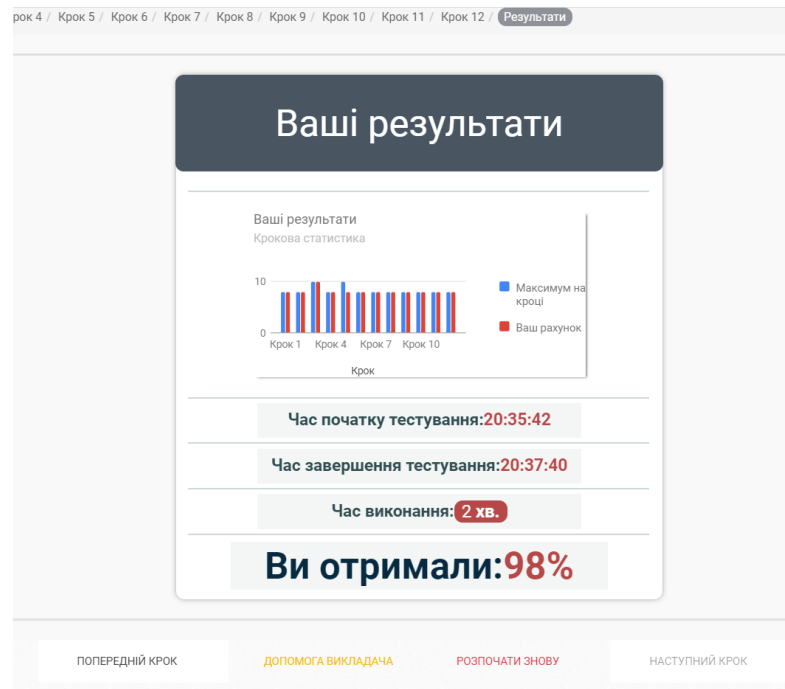


Рисунок 3.27 – Розділ «Результати» після вдалого проходження всіх кроків

Після створення всіх завдань і розділів директорія із файлами розділів виглядає наступним чином (рис. 3.28).

Имя	Дата изменения	Тип	Размер
scripts	16.04.2020 15:37	Папка с файлами	
settings	10.04.2020 13:39	Папка с файлами	
introduction	01.05.2020 20:37	Yandex Browser H...	1 КБ
results	01.05.2020 15:25	Yandex Browser H...	2 КБ
step1	05.05.2020 16:02	Yandex Browser H...	2 КБ
step2	05.05.2020 16:04	Yandex Browser H...	2 КБ
step3	05.05.2020 16:04	Yandex Browser H...	3 КБ
step4	01.05.2020 20:51	Yandex Browser H...	2 КБ
step5	05.05.2020 16:04	Yandex Browser H...	2 КБ
step6	05.05.2020 16:04	Yandex Browser H...	2 КБ
step7	05.05.2020 16:04	Yandex Browser H...	3 КБ
step8	05.05.2020 16:04	Yandex Browser H...	3 КБ
step9	05.05.2020 16:04	Yandex Browser H...	3 КБ
step10	05.05.2020 16:04	Yandex Browser H...	3 КБ
step11	05.05.2020 16:04	Yandex Browser H...	3 КБ
step12	05.05.2020 16:05	Yandex Browser H...	3 КБ
theory	06.05.2020 10:53	Yandex Browser H...	24 КБ

Рисунок 3.28 – Директорія із html-файлами розділів

У файлі конфігурації розділів прописано, що за всі кроки (окрім 3-го та 5-го) можна отримати максимально 8 балів (рис. 3.29). Кількість балів

призначено з урахуванням складності завдань та щоб у сумі за тренажер при безпомилковому проходженні можна було отримати 100 балів.

```

1  [
2  {
3      "filename"    : "introduction",
4      "score"      : 0,
5      "hasScript"  : true
6  },
7  {
8      "filename"    : "theory",
9      "score"      : 0,
10     "hasScript"   : true
11  },
12  {
13     "filename"    : "step1",
14     "score"      : 8,
15     "hasScript"  : true
16  },
17  {
18     "filename"    : "step2",
19     "score"      : 8,
20     "hasScript"  : true
21  },
22  {
23     "filename"    : "step3",
24     "score"      : 10,
25     "hasScript"  : true
26  },
27 ]

```

Рисунок 3.29 – Файл конфігурації розділів

Код файлів проекту – файлу загальної конфігурації веб-інтерфейсу, файлу конфігурації розділів, файлів різних варіантів локалізації, файлу каскадних стилів та файлу-доповнення до стилів bootstrap 3, html-файлів розділів із супутнім javascript-скриптом до них, скрипту-надбудови до фреймворка, головного файлу (головної сторінки) веб-додатку наведено у додатку В.

3.3 Тестування продукту

Розроблений тренажер було розміщено адміністратором ресурсу в матеріалах дистанційного курсу «Комп'ютерна графіка» для перевірки працездатності (рис.3.30).

Перевірка показала, що всі компоненти тренажеру працюють коректно, формують та відображають відповідні результати проходження тренажеру.

The screenshot shows a web interface for a distance learning course. The browser address bar displays `dl.sumdu.edu.ua/studio/tree/3308/show/723065#anchor723065`. The course title is "Комп'ютерна графіка - [2019-2020][ua][ІТР]". The main content area is titled "Інтерактивне практичне завдання 'Геометричні примітиви'". Below the title, there is a progress indicator showing 0.0/0.00. A "Перегляд" (View) button is present, along with a download link for a file named "primitives.zip" (3052521 bytes). A "Запускити" (Launch) button is also visible. At the bottom, a table shows the user's results for this task.

Ваші результати по данному завданню					
пройдено	розпочато	закінчено	усього балів	репліка від користув	
	2020.05.24 12:18		2020.05.24 12:18	0%	
	2020.05.24 12:11		2020.05.24 12:17	97%	YES - 97%
	2020.05.24 12:11		2020.05.24 12:11	0%	

Рисунок 3.30 – Розміщення тренажера в дистанційному курсі

Також було перевірено працездатність локалізованих іншомовних версій тренажера. Інтерфейс відображався коректно і для російськомовного (рис.3.31) і для англкомовного інтерфейсу (рис.3.32).

The screenshot shows the Russian localization of the simulator. The browser address bar displays `dl.sumdu.edu.ua/semipub/htz/723065/index.html?wrap=swf`. The page title is "Геометрические примитивы". The main heading is "Теория". The content describes the GLAUX library functions for building geometric primitives in OpenGL. It mentions the GLAUX (OpenGL Auxiliary Library) and GLUT (GL Utility Toolkit) libraries. It states that each primitive has two types of representation: Solid (Твердотельное) and Wire (Каркасное). Below the text, there are examples of functions: `auxSolidBox(width, height, depth)` and `auxWireBox(width, height, depth)`. The text explains that `width` is the width and `height` is the height. The interface also includes a navigation bar at the bottom with buttons for "ПРЕДЫДУЩИЙ ШАГ", "ПОМОЩЬ ПРЕПОДАВАТЕЛЯ", "ЗАВЕРШИТЬ СЕАНС", and "СЛЕДУЮЩИЙ ШАГ".

Рисунок 3.31 – Перевірка коректної локалізації тренажера

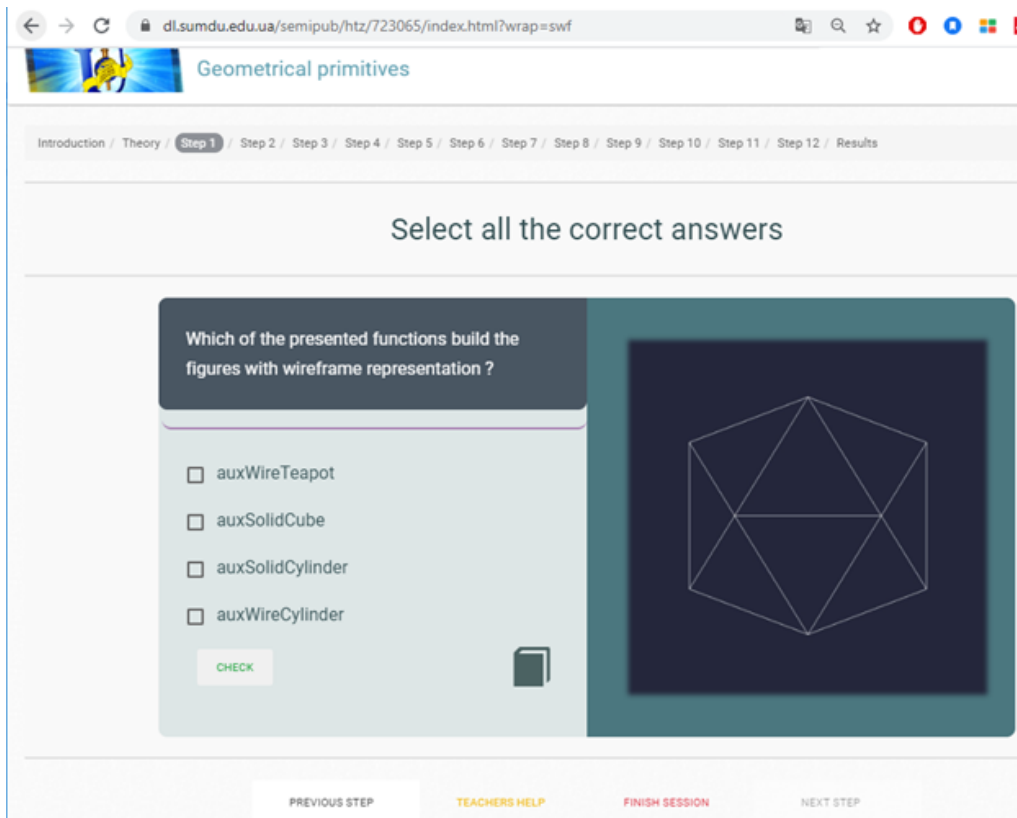


Рисунок 3.32 – Перевірка коректної локалізації тренажера

Таким чином, можна зробити висновок, що поставлені задачі проекту виконані. В результаті роботи розроблено інтерактивний тренажер для дистанційного курсу «Комп’ютерна графіка», який дозволяє студентам поліпшити теоретичні знання та отримати закріпити практичні навички з теми побудови геометричних примітивів даного курсу.

ВИСНОВКИ

У ході виконання дипломної роботи було зроблено аналіз предметної області проекту, досліджено актуальність майбутнього продукту, сформована мета та практична цінність проекту, була сформульована постановка задачі.

При дослідженні подібних програмних продуктів було виявлено, що близьких аналогів тренажер не має. Розглянуті існуючі тренажери з інших галузей, визначено їх недоліки для врахування у власному проекті.

Також було розглянуто технології, які використовуються при створенні додатків подібного роду.

Робота тренажеру була змодельована мовою графічного описання UML.

У ході планування проекту було описано зміст робіт, організаційну структуру, календарний план та проведено аналіз ризиків проекту.

На основі фреймворку для тренажерів дистанційної середовища навчання було розроблено працюючий варіант веб-тренажеру для вивчення побудови геометричних примітивів з курсу «Комп'ютерна графіка».

Після успішної розробки продукту, його було викладено на сервер дистанційного середовища навчання та протестовано.

Створений продукт допомагатиме студентам перевіряти та закріплювати знання з побудови геометричних примітивів з курсу «Комп'ютерна графіка».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Застосування комп'ютерної графіки. URL: <https://www.sites.google.com/site/informatika324/zastosuvanna-komp-uternoie-grafiki> (дата звернення: 14.04.2020).
2. Комп'ютерні технології в освіті : навч. посібн. / Ю. С. Жарких, С. В. Лисоченко, Б. Б. Сусь, О. В. Третяк. Київ: Видавничополіграфічний центр "Київський університет", 2012. 239 с.
3. Методика створення навчальних комп'ютерних інтерактивних тренажерів для вивчення загальнотехнічних дисциплін в умовах інформаційно-освітнього середовища: навч. видання / укладачі: Бабенко Д. В., Доценко Н. А. Миколаїв: Вид-во Миколаївського національного аграрного університету, 2019. 63 с.
4. Доценко Н. А. Дослідження стану використання навчальних комп'ютерних інтерактивних тренажерів в умовах інформаційно-освітнього середовища при підготовці бакалаврів з агроінженерії. Освітній дискурс: збірник наукових праць. Випуск 15 (7-8). 2019. URL: [http://journal-discourse.com/files/pdf/2019_15\(7-8\)-7.pdf](http://journal-discourse.com/files/pdf/2019_15(7-8)-7.pdf) (дата звернення: 16.04.2020).
5. Савельєв А.О., Матвєєва Н.М. Програма-тренажер як форма реалізації електронних дидактичних засобів для самостійної роботи іноземних студентів на етапі довузівської підготовки. Праці Одеського політехнічного ун-ту. 2015. Вип.2 (46). URL: <http://pratsi.opu.ua/app/webroot/articles/207-212.pdf> (дата звернення: 16.04.2020).
6. Интерактивные курсы – HTML Academy. URL: <https://htmlacademy.ru/courses> (дата звернення: 25.04.2020).
7. Тренажер з правопису української мови. URL: <https://webpen.com.ua/> (дата звернення: 25.04.2020).

8. Использование веб-квестов при обучении компьютерной графике. URL: <https://cyberleninka.ru/article/n/ispolzovanie-veb-kvestov-pri-obuchenii-kompyuternoj-grafike/viewer> (дата звернения: 25.04.2020).
9. Разработка одностраничных веб-приложений. URL: <https://avada-media.ua/services/single-page-application-development/> (дата звернения: 17.04.2020).
10. Основные библиотеки и фреймворки JavaScript, которые вы должны знать. URL: <https://code.tutsplus.com/ru/articles/essential-javascript-libraries-and-frameworks-you-should-know-about--cms-29540> (дата звернения: 17.04.2020).
11. Топ-5 JS-фреймворков для фронтенд-разработки в 2020 году. Часть 1. URL: <https://habr.com/ru/company/ruvds/blog/476286/> (дата звернения: 17.04.2020).
12. Обзор и сравнение фреймворков и библиотек JavaScript. URL: <https://oracle-patches.com/web/3899-obzor-i-sravnenie-frejmvorkov-i-bibliotek-javascript> (дата звернения: 17.04.2020).
13. Введение в AJAX и COMET. URL: <https://learn.javascript.ru/ajax-intro> (дата звернения: 17.04.2020).
14. Методология IDEF0 - Учебная и научная деятельность Анисимова Владимира Викторовича. URL: https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2 (дата звернения: 24.04.2020).
15. UML-диаграммы классов: сущности, связи, интерфейсы. URL: <https://prog-cpp.ru/uml-classes/> (дата звернения: 24.04.2020).
16. Проектирование программного обеспечения. URL: <https://habr.com/ru/post/74330/> (дата звернения: 24.04.2020).
17. Диаграмма Ганта — инструмент для тех, кто не любит срывать сроки. URL: <https://lifehacker.ru/diagramma-ganta/> (дата звернения: 22.04.2020).

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

1 Призначення й мета створення інтерактивного веб-додатку

1.1. Призначення інтерактивного веб-додатку

Інтерактивний додаток призначений для перевірки та закріплення теоретичних знань та практичних навичок з побудови геометричних примітивів специфікації OpenGL.

1.2. Мета створення інтерактивного веб-додатку –

Спрощення процесу вивчення та підвищення якості засвоєння матеріалу із основ побудови геометричних примітивів специфікації OpenGL шляхом впровадження інтерактиву.

1.3. Цільова аудиторія

Цільовою аудиторією інформаційної системи є студенти.

2 Вимоги до інтерактивного веб-додатку

2.1. Вимоги до інтерактивного веб-додатку в цілому

2.1.1. Вимоги до структури й функціонування інтерактивного веб-додатку

Веб-додаток повинен бути реалізований у вигляді веб-сторінки, доступної в мережі Інтернет в дистанційному курсі дисципліни «Комп'ютерна Графіка» сайту дистанційного навчання СумДУ – <https://elearning.sumdu.edu.ua>, мовою програмування JavaScript, яка буде змінювати вміст сторінки (приховувати пройдені завдання та додавати нові). Тобто веб-тренажер повинен бути односторінковим веб-додатком. Після проходження всіх завдань користувач повинен мати змогу побачити свій

результат (час проходження та кількість правильних відповідей) та переглянути на які питання було надано неправильну відповідь.

Весь час додаток повинен мати свій первісний вигляд, наповнення та функціонал, який впровадиться розробником перед першим розгортанням.

2.1.2. Вимоги до персоналу

Веб-тренажер не передбачає постійної підтримки, тому немає необхідності залучати персонал. Весь необхідний набір завдань повинен бути складений розробником. Після перенесення завдань у веб-представлення веб-тренажер (сукупність усіх веб-файлів) лежатиме статично у виділеній під нього директорії на сервері СумДУ.

2.1.3. Вимоги до збереження інформації

Вся загальна інформація, якою оперує веб-тренажер, повинна зберігатися в певній директорії у форматі JSON, – це налаштування локалізації, назви всіх кроків із кількістю балів за них, загальна інформація про тренажер і т.д, яка завдяки мові JavaScript зчитуватиметься із файлів і вставлятиметься в DOM-представлення веб-сторінки тренажеру.

2.1.4. Вимоги до розмежування доступу

Веб-додаток є доступним лише для студентів СумДУ дистанційної форми навчання, що вивчають дисципліну «Комп'ютерна Графіка». Доступ до тренажеру надаватиметься системою дистанційної освіти.

Користувачами веб-додатку можуть бути лише студенти та викладачі. Веб-додаток не повинен містити розмежування ролей, наприклад: адміністратор, відвідувач. Усі користувачі веб-тренажеру є відвідувачами, оскільки веб-тренажер не передбачає наявності адміністрування.

2.2. Вимоги до функцій, виконуваних веб-додатком

2.2.1. Основні вимоги

2.2.1.1. Структура веб-додатку

Веб-додаток повинен мати вигляд веб сторінки, всередину якої засобами JavaScript додається вміст (модулі із HTML та JavaScript кодом). Кожне

додавання нового змісту та приховування попереднього повинно бути переходом на новий *розділ*.

Перший *розділ* – це «Вступ», який містить інформацію про назву тренажеру, панель вибору локалізації (мови веб-інтерфейсу) та логотип тренажеру (рисунок 1).

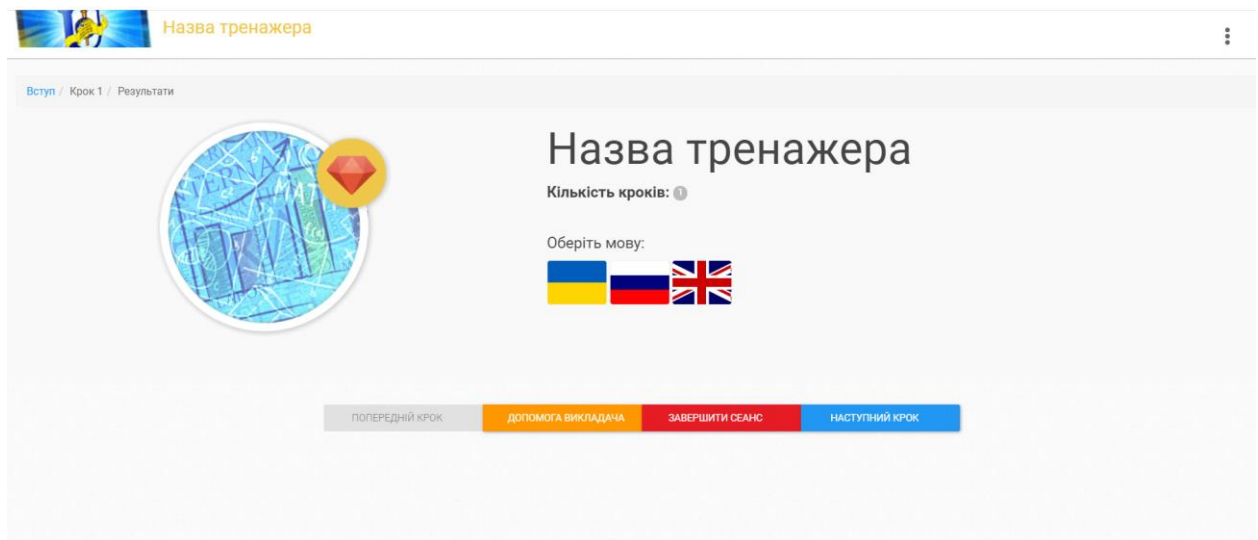


Рисунок 1 – Приклад розділу «Вступ»

Другий *розділ* – це розділ «Теорія», що містить усю необхідну під час проходження сеансу теорію.

Третій *розділ* – це перший крок тренажеру «Крок 1» (перше завдання). Усього кроків повинно бути 12 (тобто тренажер повинен містити 12 типів завдань). Кожен крок повинен містити посилання на розділ «Теорія», щоб користувач зміг передивлятися теорію в процесі проходження. Проходження тренажеру має бути лінійним.

Останнім *розділом* є «Результати», де користувач зможе дізнатися кількість правильних відповідей та час проходження сеансу. Після отримання результатів, користувач має змогу потрапити на попередні *розділи* (кроки), щоб побачити, на які питання було надано неправильні відповіді. Також користувач може потрапити на *розділ* «Результати» передчасно (пройшовши не всі кроки), натиснувши кнопку «Завершити сеанс», але він не матиме

можливість переглядати попередні розділи (пройдені кроки та розділ із теорією).

2.2.1.2. Навігація

Оскільки веб-додаток складається із розділів, які йдуть один після одного, панель навігації повинна мати вигляд рядка в кожному розділі, що знаходиться зверху по типу «breadcrumb». Рядок повинен містити імена всіх розділів. Поточний розділ, на якому зараз знаходиться користувач, повинен виділятися іншим кольором. Але ця панель повинна відігравати лише інформаційну роль та не містити посилань.

Також логотип університету у шапці веб-додатку відіграє роль навігаційного елемента – при натисканні на нього відображається перший розділ («Вступ»), але при цьому сеанс починається спочатку і користувач починає виконувати завдання спочатку.

Кнопка «Завершення сеансу» повинна перекидати користувача на розділ «Результати».

Кожен крок повинен містити значок-посилання на розділ «Теорія», а розділ «Теорія» повинен містити значок-посилання на крок, з якого ми перейшли в цей розділ (зворотне переміщення).

2.2.1.3. Наповнення веб-додатку (контент)

Кожен розділ веб-додатку повинен формуватися програмним шляхом засобами JavaScript. Тобто попереднє пройдене завдання (блок із текстовою інформацією та елементами керування) приховується, запам'ятовується відповідь, та на ту саму сторінку додається наступне завдання (блок із наступним завданням перекриває блок із попереднім завданням). Така схема роботи стосується кожного розділу.

2.2.1.4. Система навігації (карта веб-додатку)

Взаємозв'язок між розділами веб-додатку (карта веб-додатку) представлено на рисунку 2.

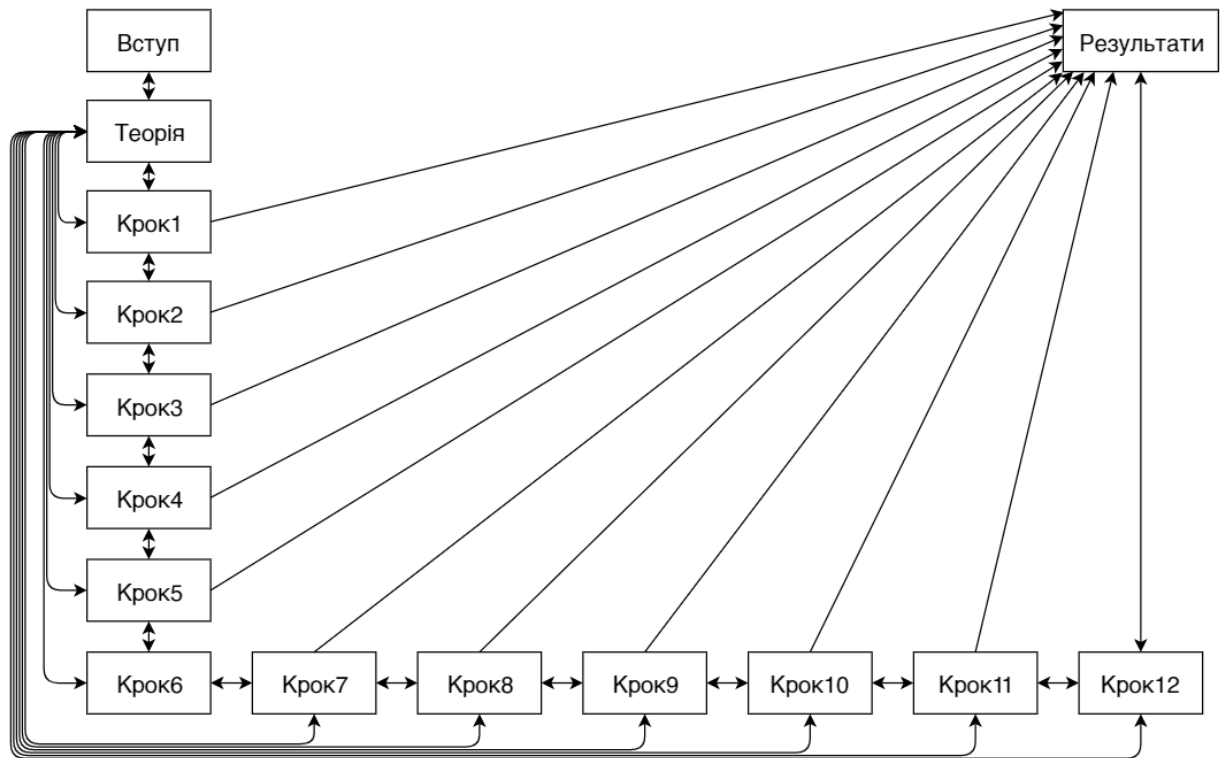


Рисунок 2 – Карта веб-додатку

2.2.1.5. Вимоги до функціональних можливостей

Веб-інтерфейс повинен містити 3 режими локалізації в розділі «Вступ»: українська, російська та англійська мови.

Кожен розділ повинен містити кнопки переходу на наступний розділ (крім останнього) та на попередній (крім першого).

На кожному кроці повинна бути наявною кнопка «Перевірити», яка перевіряє відповідь перед переходом на наступний крок. Після перевірки неправильні відповіді завдань повинні виділятися червоним кольором, а правильні - зеленим. Користувач повинен мати три спроби на крок. При кожній страченій спробі максимальна кількість балів за крок повинна зменшуватись на десять відсотків. Якщо користувач стратив всі три спроби, то сеанс починається спочатку. Також користувач повинен мати змогу на будь-якому кроці (розділі із завданням) завершити сеанс. Також кожен крок повинен містити посилання на розділ «Теорія», щоб користувач збільшив

користь свого сеансу чергуючи проходження завдань із переглядом теоретичного матеріалу.

Після проходження всіх кроків або після натискання «Завершити сеанс» користувачу потрібно показати його результати: час початку проходження сеансу, час закінчення сеансу, кількість часу проходження (різниця між часом закінчення та часом початку тренування у хвилинах) та результат проходження завдань в процентах (відношення кількості правильних відповідей до загальної кількості завдань).

2.2.1.6. Загальні вимоги

Стиль веб-тренажеру можна описати як мінімалістичний. У якості фонового кольору рекомендується використовувати кольори сайту дистанційного навчання СумДУ або кольори, гармонійні до нього.

2.2.1.7. Типові навігаційні й інформаційні елементи

- Шапка веб-тренажеру
- Основне поле контенту

2.2.1.8. Шапка інтерактивного веб-додатку

Шапка веб-тренажеру повинна містити логотип СумДУ. До шапки треба додати інформаційне вікно, спадаюче при натисканні на значок, що покаже інформацію про розробника тренажеру. Логотип СумДУ є посиланням на перший розділ – «Вступ» (сторінка оновлюється та сеанс проходження починається заново).

2.2.1.9. Основне поле контенту

Основне поле контенту повинне розташовуватися в центрі сторінки. У цьому полі відображається зміст кожного розділу. Розділ «Вступ» повинен містити логотип тренажеру, його назву, інформацію про кількість кроків та значки локалізації інтерфейсу. Розділ «Теорія» повинен містити блок із темами теоретичного матеріалу (посиланнями), при натисканні на які на передній план відображається блок із інформацією, відповідною темі. Розділи із кроками (завданнями) повинні містити блок з текстом питання, елементи вибору (радіо кнопки, кнопки і т.д.) та картинку, що відображає результат

виконання тієї чи іншої функції побудови примітивів специфікації OpenGL (рисунок 3). Розділ «Результати» повинен містити інформацію про результати проходження сеансу (рисунок 4). Незалежно від розділу під основним полем контенту тренажер повинен постійно містити однаковий набір кнопок: «Наступний крок», «Попередній крок», «Завершити сеанс», а також кнопка «Перевірити» (крім першого, другого та останнього розділу). Стильове основного тексту повинно бути єдиним для всього веб-додатку.

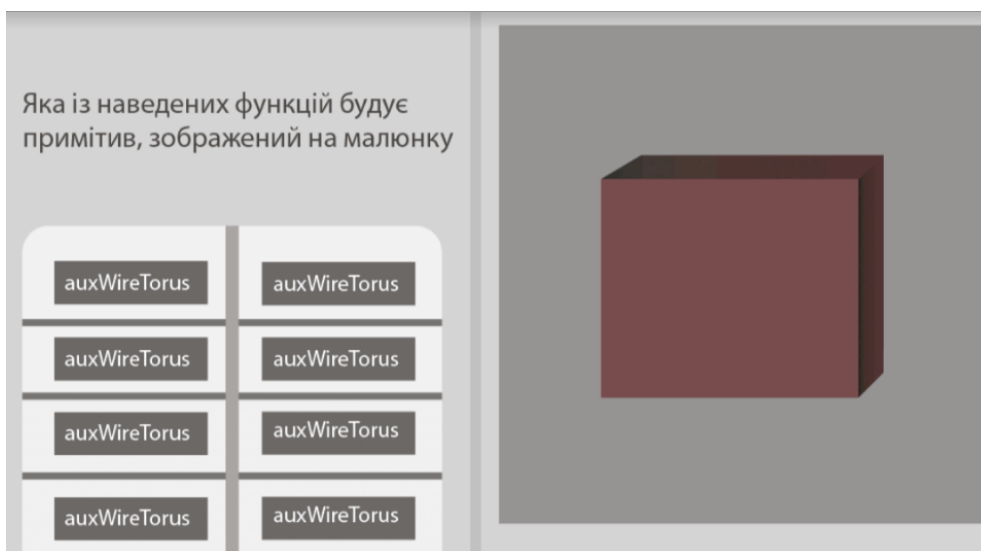


Рисунок 3 – Приклад блоку із завданням

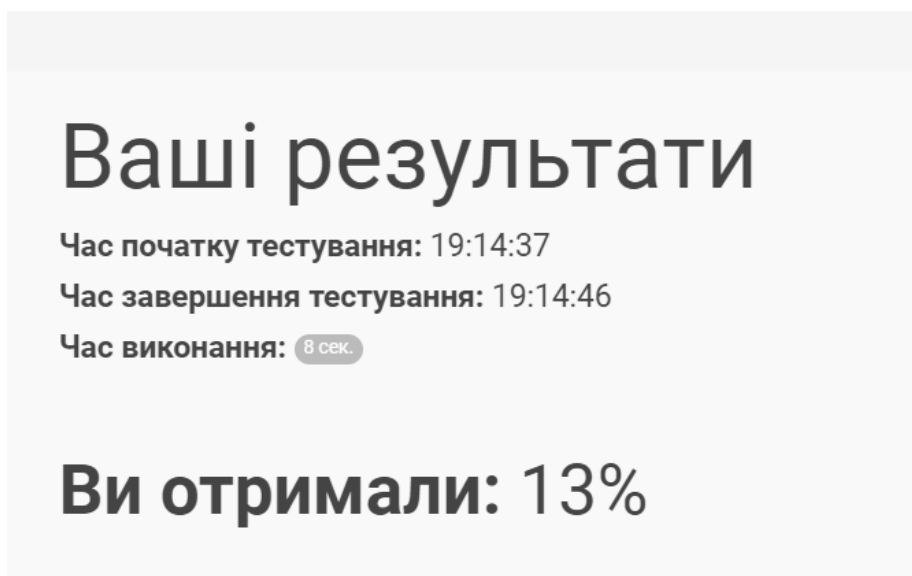


Рисунок 4 – Приклад розділу «Результати»

2.3. Вимоги до видів забезпечення

2.3.1. Вимоги до інформаційного забезпечення

Реалізація веб-тренажеру відбувається з використанням:

- Apache 2.4
- JavaScript
- HTML
- CSS

2.3.2. Вимоги до лінгвістичного забезпечення

Веб-тренажер повинен мати українську, російську та англійську локалізації (варіанти мови інтерфейсу). Потрібно запровадити можливість перемикання між наявними мовами у вигляді значків із прапорами країн у розділі «Вступ».

2.3.3. Вимоги до програмного забезпечення

Клієнтська частина повинна мати наступне програмне забезпечення:

- Веб-браузер: Google Chrome, Yandex, Internet Explorer, Firefox, Opera;
- Включена підтримка JavaScript.

2.3.4. Вимоги до апаратного забезпечення

Серверна частина повинна мати наступне апаратне забезпечення:

- Веб-сервер Apache;
- Не менш 100 МБ вільного місця на диску.

3 Склад і зміст робіт зі створення веб-додатку

Докладний опис етапів роботи зі створення веб-тренажеру наведено в табл.1.

Таблиця 1 – Етапи створення веб-тренажеру

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Створення прототипу макету інтерфейсу	3 дні
2	Верстання макету	4 дні
3	Складання завдань	3 дні
4	Переведення завдань у веб-представлення	4 дні
5	Впровадження різних варіантів локалізації інтерфейсу	3 дні
6	Впровадження логіки наповнення-вивантаження вмісту веб-сторінки при переміщенні між розділами	5 дні
7	Додавання механізму спроб	3 дні
8	Створення механізму підрахування результатів проходження сеансу	4 дні
9	Створення блоку теорії, який з'являтиметься при натисканні на кнопку	3 дні
10	Перевірка розробленої системи на помилки та їх виправлення	5 дні
11	Виправлення недоліків зовнішнього вигляду (дизайну)	2 дні
12	Оформлення супутньої документації	12 днів
	Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту	51

4 Вимоги до складу й змісту робіт із тренажеру в експлуатацію

Для успішного функціонування веб-тренажеру на хостингу необхідно, щоб вся логіка тренажеру була цілком описана мовою JavaScript (серверні мови програмування виключаються). Веб-додаток повинен розроблятися слідуючи вимогам, висунутим адміністратором веб-тренажерів СумДУ на основі обмежень серверної сторони.

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

Деталізація мети проекту методом SMART. Мета проекту – розробити продукт, що матиме вигляд веб-тренажеру, для вивчення основ тривимірної графіки в основі специфікації OpenGL за темою «Примітиви» (прості тривимірні фігури, з яких найчастіше починається 3d-моделювання). Продукт вноситиме різнобарв'я в процес навчання студента на дистанційному курсі дисципліни «Комп'ютерна графіка» та надасть можливість студенту перевірити свій рівень знань. Результати деталізації методом SMART розміщені у табл. Б.1.

Планування змісту структури робіт. Планування змісту структури робіт виконується із використанням такого інструменту, як WBS-діаграма – графічне представлення пакету робіт, які були сформульовані завдяки декомпозиції загальних робіт або робіт верхнього рівня (тобто загальні роботи розбиваються на елементарні роботи, які розташовуються на найнижчому рівні структури). Роботи в структурі розташовуються згідно запланованої хронології розробки продукту (тобто згідно передбачуваної послідовності виконання робіт). Діаграма WBS зображена на рис. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити веб-додаток для вивчення та засвоєння інформації із теми «Примітиви» специфікації OpenGL для дистанційного курсу дисципліни «Комп'ютерна графіка».
Measurable (вимірювана)	Результатом роботи проекту є висновки замовника щодо виконаної роботи та успішне розміщення продукту в середовищі дистанційного навчання університету.
Achievable (досяжна)	Створення продукту здійснюється за допомогою браузера Yandex, тестового редактора Adobe Brackets та локального сервера Open Server.
Relevant (реалістична)	Для розробки веб-тренажера потрібні знання мови веб-розмітки HTML, мови каскадних стилів CSS, мови веб-програмування JavaScript. Також для спрощення верстання потрібно мати знання фреймворка Bootstrap, який пришвидшить процес створення інтерфейсу. Розробник веб-тренажеру має достатньо знань для реалізації поставлених завдань.
Time-framed (обмежена у часі)	Для успішної презентації продукту вся робота повинна бути виконана згідно строків календарного плану.

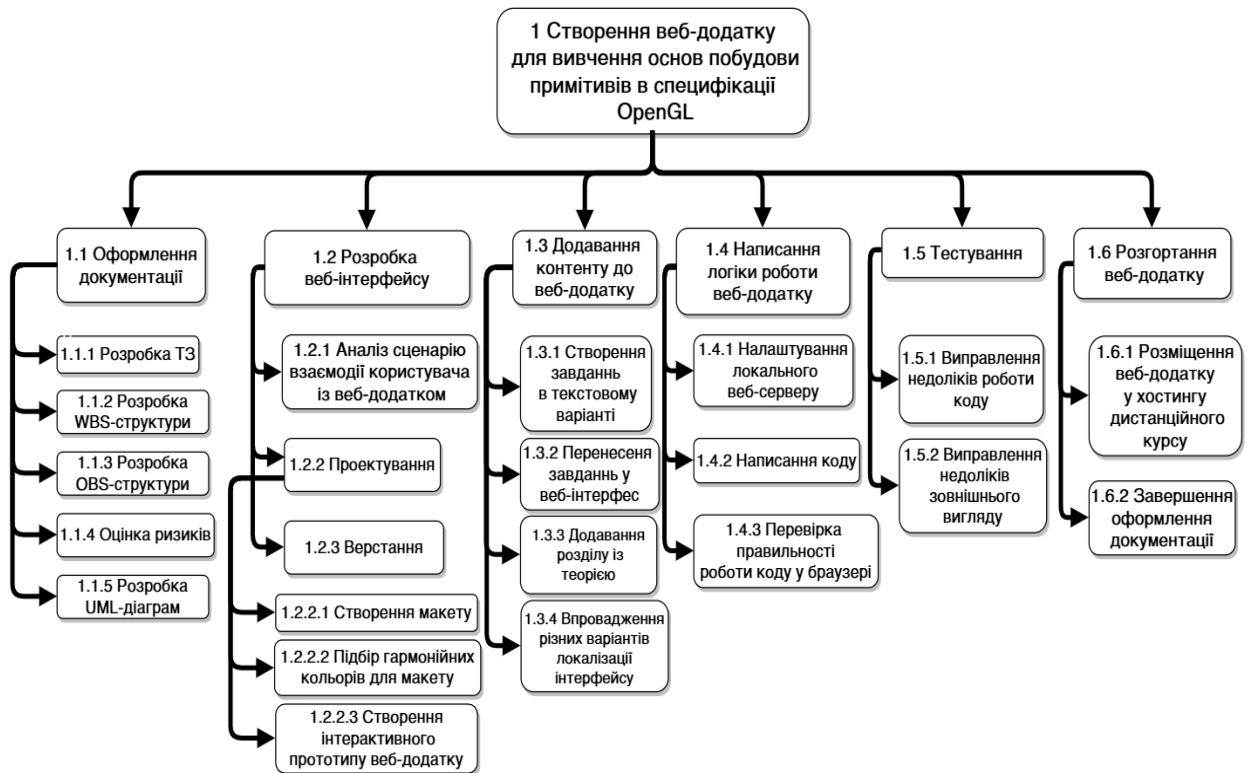


Рисунок Б.1 – WBS-структура робіт проекту

Планування структури організації, для впровадження готового проекту (OBS). Для графічного представлення внутрішньої організаційної структури виконання робіт проекту використаємо діаграму OBS. Діаграма OBS представлена на рис. Б.2.

Список виконавців, що функціонують в проекті наведено в табл. Б.2.

Діаграма Ганта. Для графічного представлення календарного плану виконання робіт використаємо діаграму Ганта, що ілюструє задачі в залежності від послідовності їх виконання [17]. Тривалість виконання робіт представлена в днях (загально). На день виділяється приблизно 3-6 годин робочого часу. Загальна кількість днів є різницею між датою початку проекту та датою розгортання продукту. Діаграма Ганта та список робіт зображені на рис. Б.3-Б.6.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Дуванов С.С.	Розробляє основний функціонал проекту та дизайн продукту.
Тестувальник	Дуванов С.С.	Відповідає за тестування функціоналу, перевірку дизайну та наявності помилок системи.
Керівник проекту	Баранова І.В.	Формує завдання на розробку проекту та контролює строки виконання робіт.
Консультант	Кучкіна Л.	Надає рекомендації щодо розробки функціоналу веб-додатку.
Розгортувач продукту	Кучкіна Л.	Виконує розміщення додатку на хостингу дистанційного курсу дисципліни «Комп'ютерна графіка».

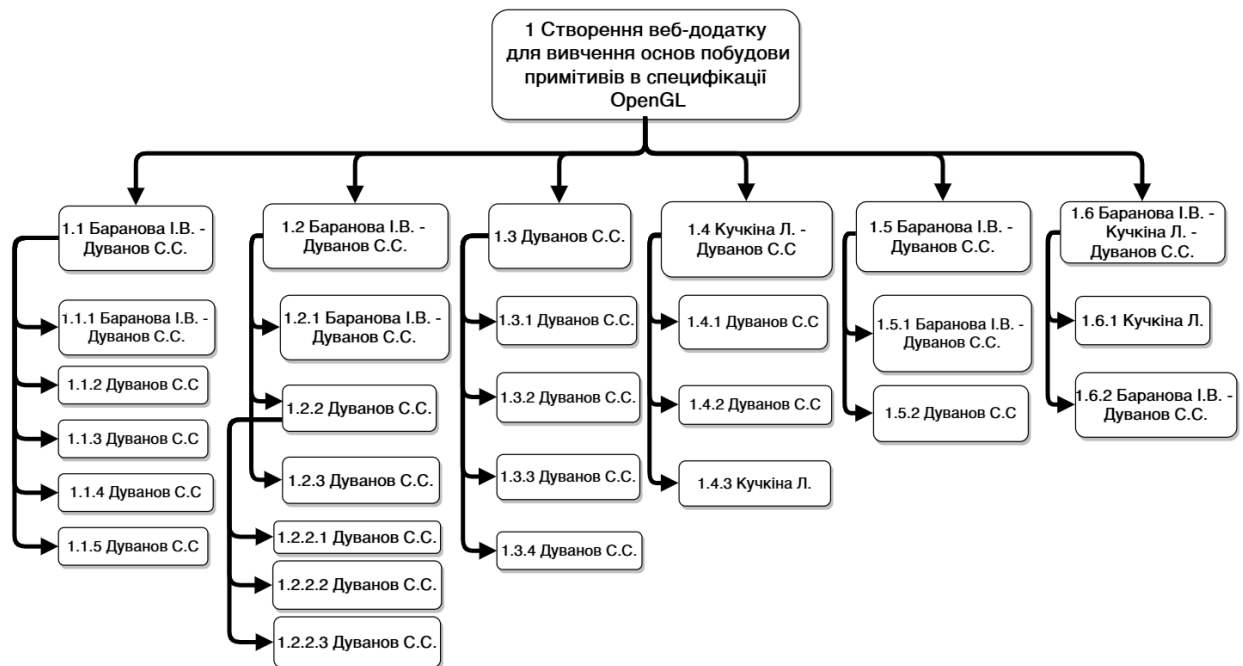


Рисунок Б.2 – Організаційна структура проекту (OBS)

№	Задачі	Дата початку	Дата кінця	Кількість днів
1	Створення веб-додатку для вивчення основ побудови примітивів в специфікації OpenGL	04.04.2020	25.05.2020	51
1.1	Оформлення документації	04.04.2020	16.04.2020	12
1.1.1	Розробка ТЗ	04.04.2020	07.04.2020	3
1.1.2	Розробка WBS-структури	08.04.2020	09.04.2020	1
1.1.3	Розробка OBS-структури	08.04.2020	09.04.2020	1
1.1.4	Оцінка ризиків	10.04.2020	12.04.2020	2
1.1.5	Розробка UML-діаграм	13.04.2020	16.04.2020	3
1.2	Розробка веб-інтерфейсу	16.04.2020	23.04.2020	7
1.2.1	Аналіз сценарію взаємодії користувача із веб-додатком	16.04.2020	18.04.2020	2
1.2.2	Проектування	16.04.2020	20.04.2020	4
1.2.2.1	Створення макету	16.04.2020	18.04.2020	2
1.2.2.2	Підбір гармонійних кольорів для макету	16.04.2020	18.04.2020	2
1.2.2.3	Створення інтерактивного прототипу веб-додатку	18.04.2020	20.04.2020	2
1.2.3	Верстання	20.04.2020	23.04.2020	3
1.3	Додавання контенту до веб-додатку	24.04.2020	04.05.2020	10
1.3.1	Створення завдань в текстовому варіанті	24.04.2020	28.04.2020	4
1.3.2	Перенесення завдань у веб-інтерфес	24.04.2020	28.04.2020	4
1.3.3	Додавання розділу із теорією	28.04.2020	30.04.2020	2
1.3.4	Впровадження різних варіантів локалізації інтерфейсу	01.05.2020	04.05.2020	3
1.4	Написання логіки роботи веб-додатку	05.05.2020	12.05.2020	7
1.4.1	Налаштування локального веб-серверу	05.05.2020	06.05.2020	1
1.4.2	Написання коду	05.05.2020	12.05.2020	7
1.4.3	Перевірка правильності роботи коду у браузері	05.05.2020	12.05.2020	7
1.5	Тестування	13.05.2020	20.05.2020	7
1.5.1	Виправлення недоліків роботи коду	13.05.2020	18.05.2020	5
1.5.2	Виправлення недоліків зовнішнього вигляду	19.05.2020	20.05.2020	1
1.6	Розгортання веб-додатку	21.05.2020	25.05.2020	4
1.6.1	Розміщення веб-додатку у хостингу дистанційного курсу	21.05.2020	22.05.2020	1
1.6.2	Завершення оформлення документації	21.05.2020	25.05.2020	4

Рисунок Б.3 – Список робіт діаграми Ганта

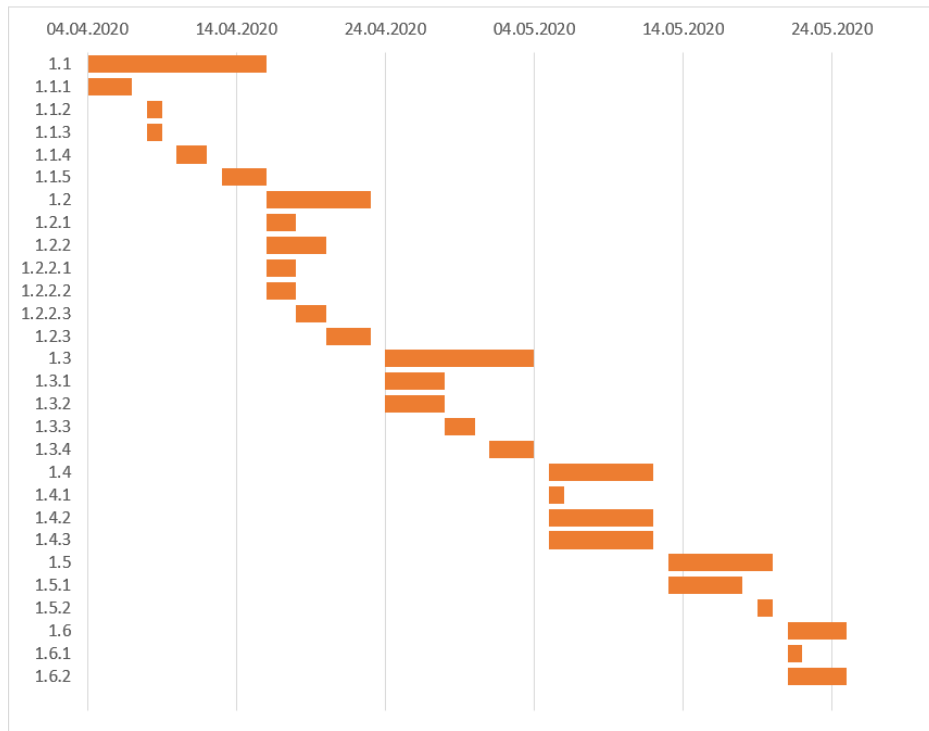


Рисунок Б.4 – Діаграма Ганта



Рисунок Б.5 – Продовження діаграми Ганта

Аналіз ризиків. Виконаємо якісне і кількісне оцінювання ризиків роботи. При якісному оцінюванні визначимо ризики, які потребують невідкладного реагування. Завдяки якісному оцінюванню визначаються ступені важливості ризиків і способи реагування на ці ризики. Кількісне оцінювання ризиків буде проведено для більш повної ідентифікації ризиків та ступеня їхнього впливу на виконання робіт проекту. Оцінено ризики за показниками, що знаходяться в табл. Б.3.

Таблиця Б.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

Після оцінювання ризиків виконаємо планування реагування на ризики — це розробка методів зменшення негативного впливу ризиків на проект. На основі оцінки побудуємо матрицю ймовірності виникнення та впливу ризиків, що зображена на рис. Б.6.

Ймовірність виникнення	3			RS_5, RS_10
	2			RS_2, RS_6
	1	RS_7	RS_3, RS_4, RS_8	RS_1, RS_9
		1	2	3
		Вплив ризику		

Рисунок Б.6 – Матриця ймовірності виникнення та впливу ризиків

- зелений колір – прийнятні ризики;
- жовтий колір – виправдані ризики;

- червоний колір – неприпустимі ризики.

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. Б.4.

Таблиця Б.4 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	$1 \leq R \leq 2$	3,4,7,8
2	Виправдані	$3 \leq R \leq 4$	1,9
3	Недопустимі	$6 \leq R \leq 9$	2,5,6,9

У табл. Б.5 знаходиться класифікація ризиків за показниками ймовірності виникнення ризику та величин втрат.

Таблиця Б.5 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Складність розгортання продукту на хостингу	Низька	Високий	3	<p>1. Дізнатися в адміністраторів дистанційного курсу університету вимоги щодо веб-тренажерів для вдалого, в майбутньому, розміщення продукту на хостинг.</p> <p>2. Слідування вимогам підчас розробки.</p>	Попередження	
RS_2	Відкритий	Марне витрачання часу при розробці	Середня	Високий	6	Дослідження технологій, що прискорюють процес верстання та розробки для запобігання «вигадування чогось з нуля».	Пом'якшення	Чергування робіт між собою задля розуміння реального складу справ та оптимального розподілу часу між роботами.

Продовження таблиці Б.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_3	Відкритий	Неузгодженість концепції веб-додатку з керівником	Низька	Середній	2	1. Формування з керівником, напочатку, загального представлення функціонування продукту, його дизайну. 2. Постійне обговорення з керівником поточних результатів після коротких етапів розробки продукту.	Попередження	
RS_4	Відкритий	Збої в роботі серверу університету	Низька	Середній	2	Чекати доки технічний персонал налаштує роботу серверу.	Прийняття	
RS_5	Відкритий	Неправильний тайм-менеджмент робіт	Висока	Високий	9	Переоцінка календарного плану	Пом'якшення	Щоденний моніторинг поточного стану виконання робіт.

Продовження таблиці Б.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_6	Відкритий	Неправильна оцінка можливостей розробника	Середня	Високий	6	Паралельне чергування етапів розробки продукту розробником задля отримання загального розуміння своїх можливостей.	Пом'якшення	Спрощення функціоналу додатку. Отримання знань в галузях, що спонукають появи запитань.
RS_7	Відкритий	Перевиконання поставленої задачі	Низька	Низький	1	Відсіювання непотрібного функціоналу, який не нестиме великої користі.	Пом'якшення	Переогляд поставленого завдання із замовником у ході розробки продукту.
RS_8	Відкритий	Неправильна оцінка розміру проекту	Низька	Середній	2	Обговорення із керівником всіх аспектів проекту під час його ініціації та протягом розробки.	Пом'якшення	Внесення коректив в поставлену задачу в напрямку спрощення.
RS_9	Відкритий	Неправильне розуміння поставлених замовником задач	Низька	Високий	3	Підтримка постійного контакту із замовником у ході виконання кожного етапу проекту.	Пом'якшення	
RS_10	Відкритий	Помилки в роботі розробленого продукту	Висока	Високий	9	Постійне тестування у ході розробки продукту.	Пом'якшення	Показ замовнику проміжних результатів розробки продукту.

ДОДАТОК В

ЛІСТИНГ КОДУ

Файл `trainer.html`

```

<!DOCTYPE html>
<html>
<head>
  <title>jTrainer :: {{TRAINER_NAME}}</title>
  <meta charset="UTF-8"/>
  <meta name="description" content=""/>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="Cache-Control" content="private">
  <meta http-equiv="Cache-Control" content="no-cache">

  <script>
    var startTime = (new Date()).getTime();
  </script>
  <script src="/cdn/javascripts/jtrainer/current/js/jquery.min.js"></script>
  <script>
    $(window).load(function () {
      var endTime = (new Date()).getTime();
      var secondsLoading = ((endTime - startTime) / 1000).toFixed(4);
      $('<span>.page-loading</span>').append(secondsLoading + ' seconds. ');
    });
  </script>
  <script src="/cdn/javascripts/jtrainer/current/js/mustache.js"></script>

  <link rel="stylesheet" href="/cdn/javascripts/jtrainer/current/css/animate.css" type="text/css"/>

  <!-- Bootstrap 3 -->
  <link rel="stylesheet" href="/cdn/javascripts/jtrainer/current/css/bootstrap.min.css" type="text/css"/>
  <script src="/cdn/javascripts/jtrainer/current/js/bootstrap.min.js" ></script>

  <!-- Bootstrap 4 button style adds (CUSTOM)-->
  <link rel="stylesheet" href="css/bootstrap-adds.css" type="text/css"/>

  <!-- jTrainer's styles -->
  <link rel="stylesheet" href="/cdn/javascripts/jtrainer/current/css/trainer.style.css" type="text/css"/>
  <link rel="stylesheet" href="css/trainer.custom.css" type="text/css"/>

  <!-- jTrainer's engine script -->
  <script src="/cdn/javascripts/jtrainer/current/js/trainer.engine.js"></script>

  <!-- Custom script -->
  <script src="/trainer/scripts/customScript.js"></script>

</head>
<body>
<nav class="navbar navbar-default">
  <div class="navbar-header">
    <div><a class="navbar-brand" href="#" onclick="window.location.reload()"><span class="trainer-brand"></span>{{TRAINER_NAME}}</a></div>
    </div>
    <div id="navbar" class="navbar-collapse collapse">

```

```

<div class="navbar-right info">
  <div class="dropdown">
    <button type="button" class="btn btn-link dropdown-toggle" id="menu" data-toggle="dropdown"
      aria-haspopup="true" aria-expanded="true" aria-label="Option Vertical"><span
        class="glyphicon glyphicon-option-vertical" aria-hidden="true"></span></button>
    <ul class="dropdown-menu" aria-labelledby="menu">
      <li><a href="#" data-toggle="modal" data-target="#aboutModal">About</a></li>
      <li><a href="#" class="page-loading text-muted"><span class="is-faded">Page generated in
</span></a>
      </li>
    </ul>
  </div>
  <div class="modal fade" id="aboutModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
            aria-hidden="true">&times;</span></button>
          <h4 class="modal-title" id="aboutModalLabel">About</h4>
        </div>
        <div id="aboutBody" class="modal-body">
          ...
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="modal modal-static fade" id="cogwheel-modal" role="dialog" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-body">
        <div class="text-center">
          
          <h4><span id="cogwheel-desc">Processing</span>... </h4>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="text-center animated bounceInUp">
  <div class="btn-group btn-group-md rotators">
    <a id="prevController" href="javascript:void(0)" class="btn btn-default col-xs-6 col-md-3"
>{{PREV_STEP}}</a>
    <a class="teacher-help btn btn-outline-warning col-xs-6 col-md-3" data-toggle="modal"
      data-target="#helpModal">{{TEACHER_HELP}}</a>
    <a id="endTrainer" class="btn btn-outline-danger col-xs-6 col-md-3" data-toggle="modal" data-
      target="#endModal">{{END_TRAINER}}</a>
    <a id="begin" class="btn btn-outline-danger col-xs-6 col-md-3"
      onclick="window.location.reload(true)">{{BEGIN_TRAINER}}</a>
    <a id="nextController" href="javascript:void(0)" class="btn btn-default col-xs-6 col-md-
      3">{{NEXT_STEP}}</a>
  </div>
</div>

```



```

<!-- This is modal popup which shows up after click on "Teacher Help" button -->
<div class="modal fade" id="helpModal" tabindex="-1" role="dialog" aria-labelledby="helpModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close" id="closeButton"><span
          aria-hidden="true">&times;</span></button>
        <h4 class="modal-title" id="helpModalLabel">{{TEACHER_HELP}}</h4>
      </div>
      <div class="modal-body">
        <label class="control-label">{{ENTER_HELP_TEXT}}</label>
        <input type="text" class="form-control" id="helpInput"/>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">{{HELP_REQUEST_CLOSE}}</button>
        <button type="button" class="btn btn-primary"
onclick="Service.getHelpModal()">{{HELP_REQUEST_SEND}}
        </button>
      </div>
    </div>
  </div>
</div>
<!-- This is modal popup which shows up after click on "End trainer" button -->
<div class="modal fade" id="endModal" tabindex="-1" role="dialog" aria-labelledby="endModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close" id="closeButton"><span
          aria-hidden="true">&times;</span></button>
        <h4 class="modal-title" id="endModalLabel">{{END_TRAINER}}</h4>
      </div>
      <div class="modal-body">
        <label class="control-label">{{END_ARE_YOU_SURE}}</label>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">{{END_TRAINER_CLOSE}}</button>
        <button type="button" class="btn btn-primary" onclick="Service.pushResultsEarly()">
          {{END_TRAINER_SEND}}
        </button>
      </div>
    </div>
  </div>
</div>
<div class="progress trainer-progress-bar">
  <div class="progress-bar"></div>
</div>
<div class="alert alert-dismissible alert-success validation-alert validation-alert-success">
<!-- <button type="button" class="close" data-dismiss="alert">×</button>-->
  <strong>{{ALERT_WELL_DONE}}</strong>{{ALERT_STEP_SUCCESS}}
</div>
<div class="alert alert-dismissible alert-danger validation-alert validation-alert-danger">
  <button type="button" class="close" data-dismiss="alert" onclick="window.location.reload(true)">×</button>
  {{ALERT_STEP_FAIL}}
</div>
<div class="under-alert-danger"></div>

```

```

<div class="under-alert-danger"></div>

<div class="alert alert-dismissible alert-danger validation-alert validation-alert-middle">
  {{ALERT_FAIL}}
</div>

</body>
</html>

<script type="text/javascript" src="/cdn/javascripts/jtrainer/current/js/additions/html2canvas.js" async></script>

<script type="text/javascript">

</script>

```

Файл ua.json

```

{
  "local": "Українська",
  "lang": {
    "TRAINER_NAME": "Геометричні примітиви",
    "MAX_SCORE": "Кількість балів",
    "NEXT_STEP": "Наступний крок",
    "PREV_STEP": "Попередній крок",
    "ENTER_TEXT": "Введіть текст",
    "CHOOSE_LANG": "Оберіть мову",
    "CHOOSE_SELECT": "Оберіть елемент зі списку",
    "TEACHER_HELP": "Допомога викладача",
    "ENTER_HELP_TEXT": "Опишіть з чим Вам потрібна допомога...",
    "HELP_REQUEST_SEND": "Відправити",
    "HELP_REQUEST_CLOSE": "Відмінити",
    "ALERT_WELL_DONE": "Відмінно!",
    "ALERT_FAIL": "<strong>Помилка!</strong> Спробуйте ще",
    "ALERT_STEP_SUCCESS": "Крок пройдений! Перейдіть до наступного.",
    "ALERT_STEP_FAIL": "Ви погано засвоїли теоретичний матеріал. Повторіть його і почніть виконувати завдання з початку.",
    "STEP_COUNT": "Кількість кроків",
    "END_TRAINER": "Завершити сеанс",
    "BEGIN_TRAINER": "Розпочати знову",
    "END_ARE_YOU_SURE": "Ви впевнені, що хочете завершити виконання тренажера?",
    "END_TRAINER_CLOSE": "Продовжити поточний сеанс",
    "END_TRAINER_SEND": "Завершити поточний сеанс",
    "CHECK": "Перевірити",
    "ATTEMPTS_LEFT": "Ще спроб - ",

    "THEORY_TITLE": "Теорія",
    "TO_THEORY": "До теорії",
    "FROM_THEORY": "Назад",

    "THEORY_COLUMN1": "Примітиви (GLAUX)",
    "THEORY_COLUMN2": "Примітиви (GLUT)",
    "THEORY_COLUMN3": "Координати",
    "THEORY_COLUMN4": "Трансформація",
    "THEORY_COLUMN5": "Масштабування",
    "THEORY_COLUMN6": "Колір",

    "TH1_HEAD": "Функції бібліотеки <span style='color: #259191;'>GLAUX</span>",

```

"TH1_TEXT1": "Для побудови геометричних примітивів у OpenGL використовують функції додаткових бібліотек, зокрема бібліотеки GLAUX (OpenGL Auxiliary Library) та GLUT (GL Utility Toolkit)."

"TH1_TEXT2": "Кожен примітив має два типи представлення: Solid (Твердотільне) та Wire (Каркасне), що відображається у назві функції. Нижче наведено приклади функцій побудови примітивів."

"TH1_TABLE1": "Коробка
width – ширина,
height – висота,
depth – глибина примітиву",

"TH1_TABLE2": "Торr – внутрішній радіус,R – зовнішній радіус",

"TH1_TABLE3": "Кубwidth – розмір кубу",

"TH1_TABLE4": "Циліндр",

"TH1_TABLE5": "Конус",

"TH1_TABLE6": "Ікосаедр",

"TH1_TABLE7": "Октаедр",

"TH1_TABLE8": "Тетраедр",

"TH1_TABLE9": "Додекаедр",

"TH1_TABLE10": "Чайник",

"TH1_TABLE11": "Сфера",

"TH2_HEAD": "Функції бібліотеки GLUT",

"TH2_TEXT1": "Кожен примітив має два типи представлення: Solid (Твердотільне) та Wire (Каркасне) - що відображається у назві функції. Функції цієї бібліотеки дуже схожі на функції бібліотеки GLAUX. Проте на відміну від неї, деякі із функцій містять більше параметрів, (наприклад, можна детальніше налаштувати кількість поздовжніх та поперечних ділень), або зовсім не містять параметрів (не можна задавати навіть габарити)."

"TH2_TEXT2": "Наприклад, для ікосаедра, тетраедра, октаедра не вказують параметр розміру об'єкта - функція їх побудови налаштована таким чином, що радіус описаної сфери навколо об'єкта дорівнює 1. Аналогічно, при побудові додекаедра радіус описаної сфери навколо об'єкта дорівнює квадратному кореню з трьох."

"TH2_TEXT3": "Нижче наведено приклади функцій побудови примітивів твердотілого представлення (каркасне представлення налаштовується так само)",

"TH2_TABLE1": "Конусbase - радіус основи,height - висота конуса,slices - кількість поздовжніх ділень,stacks - кількість поперечних ділень",

"TH2_TABLE2": "ТорinnerRadius - внутрішній радіус,outerRadius - зовнішній радіус тора,sides - кількість ділень вздовж осі тора,rings - кількість ділень поперек осі тора",

"TH2_TABLE3": "Сфераsize - радіус сфери,slices - кількість поздовжніх ділень,stacks - кількість поперечних ділень",

"TH2_TABLE4": "Кубsize - розмір примітиву",

"TH2_TABLE5": "Чайникsize - розмір примітиву",

"TH2_TABLE6": "Додекаедр",

"TH2_TABLE7": "Ікосаедр",

"TH2_TABLE8": "Октаедр",

"TH2_TABLE9": "Тетраедр",

"TH3_HEAD": "Координатна система",

"TH3_TEXT1": "В OpenGL система координат має наступну орієнтацію осей",

"TH3_TEXT2": "За замовчуванням примітиви завжди будуються в початку координат, тобто у точці (0,0,0)."

"TH3_TEXT3": "Перед тим, як вносити зміни в систему координат, нам потрібно запам'ятати поточні координати, - це дозволить в подальшому використовувати функції зміни координат.",

"TH3_TEXT4": "Функція glPushMatrix() - зберігає поточну систему координат, glPopMatrix() - повертає параметри до останньої збереженої системи координат.",

"TH3_TEXT5": "Тобто ви можете змінювати координати, будувати й трансформувати примітиви, та потім відновити координатну систему у попередньо збереженому вигляді. Також можна робити вкладення із цих функцій до 10-го рівня. Наприклад",

"TH4_HEAD": "Трансформація об'єктів",

"TH4_TEXT1": "Для можливості відображення об'єктів не на початку системи координат, використовують функції трансформації (зміни)."

"TH4_TEXT2": "Існують такі функції: glTranslated - функція переміщення об'єкта у точку з заданими координатами, glRotated - функція обертання об'єкта на заданий кут відносно осей координат."

"TASK3": "Задайте вірну послідовність розташування рядків коду, щоб результат виконання програми збігався із результатом на рисунку",
"MARK_STEP3_INPUT1": "glPopMatrix();",
"MARK_STEP3_INPUT2": "auxSolidCube(2);",
"MARK_STEP3_INPUT3": "auxSolidCube(4);",
"MARK_STEP3_INPUT4": "glTranslated(-4,5,0);",
"MARK_STEP3_INPUT5": "glPushMatrix();",
"TASK3_TOOLTIP": "Результат роботи коду",

"TASK4_TITLE": "Оберіть усі правильні відповіді",
"TASK4": "Що робить функція `glPushMatrix()` ?",
"MARK_STEP4_CHECKBOX1": "видаляє поточну матрицю перетворення",
"MARK_STEP4_CHECKBOX2": "зберігає поточну матрицю перетворення",
"MARK_STEP4_CHECKBOX3": "налаштовує проекцію зображення вікна",
"MARK_STEP4_CHECKBOX4": "надає можливість будувати декілька фігур в різних точках простору",

"TASK5_TITLE": "Перетягніть блоки з відповідями в поле",
"TASK5": "Які із перелічених функцій не були застосовані для побудови фігур на рисунку ?",
"TASK5_TOOLTIP": "Набір фігур",

"TASK6_TITLE": "Оберіть правильну відповідь",
"TASK6": "Яка із перелічених бібліотек не містить функцій побудови примітивів ?",
"STEP6_RADIO_TEXT_1": "glu",
"STEP6_RADIO_TEXT_2": "aux",
"STEP6_RADIO_TEXT_3": "std",
"STEP6_RADIO_TEXT_4": "glut",

"TASK7_TITLE": "Оберіть коректний елемент зі списку",
"TASK7": "Виберіть елемент коду, який дасть результат як на рисунку",
"STEP7_OPTION_ONE": "glTranslated(45,0,0);",
"STEP7_OPTION_TWO": "glRotated(45,1,0,0);",
"STEP7_OPTION_THREE": "glScaled(45,0,0);",
"STEP7_OPTION_FOUR": "glRotated(45,0,0,1)",
"TASK7_TOOLTIP": "Результат роботи коду",

"TASK8_TITLE": "Впишіть числа у текстові поля (від 1 до 3)",
"TASK8": "Задайте порядок застосування зазначених нижче функцій перед використанням функції відображення куба `auxWireBox`, щоб отримати результат, наведений на рисунку",
"MARK_STEP8_INPUT1": "glScaled(2,2,2);",
"MARK_STEP8_INPUT2": "glTranslated(-4,0,0);",
"MARK_STEP8_INPUT3": "glRotated(45,1,1,0);",
"TASK8_TOOLTIP": "Послідовність викликання функцій",

"TASK9_TITLE": "Впишіть числа у текстові поля",
"TASK9": "У кодї програми задано відображення об'єкта «куб». Задайте такі значення параметрів функції `glRotated(45,a,b,c)`, щоб результат вийшов як на рисунку",
"TASK9_TOOLTIP": "Результат обертання",

"TASK10_TITLE": "Оберіть коректний елемент зі списку",
"TASK10": "Виберіть із списку фрагмент коду, що дозволить отримати результат, показаний на рисунку",
"STEP10_OPTION_ONE": "glTranslated(0,2,0);",
"STEP10_OPTION_TWO": "glTranslated(0,-2,0);",
"STEP10_OPTION_THREE": "glScaled(2,2,0);",
"STEP10_OPTION_FOUR": "glScaled(1,3,1);",
"TASK10_TOOLTIP": "Результат роботи коду",

"TASK11_TITLE": "Впишіть слова у текстові поля (регістр літер враховується)",
"TASK11": "Впишіть назви функцій трансформації (перетворення) для параметрів із типом даних `double`",
"MARK_STEP11_TEXTINPUT1": "Масштабування",
"MARK_STEP11_TEXTINPUT2": "Переміщення",

```

"MARK_STEP11_TEXTINPUT3": "Обертання",

"TASK12_TITLE": "Впишіть слова у текстові поля",
"TASK12": "Впишіть назви кольорів (англійською мовою), які відповідають параметрам функції
<em>glColor3d(a,b,c)</em> та визначають загальний колір об'єкту",

"RESULTS_TITLE": "Ваші результати",
"RESULTS_START_TIME": "Час початку тестування",
"RESULTS_END_TIME": "Час завершення тестування",
"RESULTS_TIME_DIFF": "Час виконання",
"RESULTS_TIME_SEC": "хв.",
"RESULTS_YOUR_SCORE": "Ви отримали",

"STEP1_NAME": "Вступ",
"STEP2_NAME": "Теорія",
"STEP3_NAME": "Крок 1",
"STEP4_NAME": "Крок 2",
"STEP5_NAME": "Крок 3",
"STEP6_NAME": "Крок 4",
"STEP7_NAME": "Крок 5",
"STEP8_NAME": "Крок 6",
"STEP9_NAME": "Крок 7",
"STEP10_NAME": "Крок 8",
"STEP11_NAME": "Крок 9",
"STEP12_NAME": "Крок 10",
"STEP13_NAME": "Крок 11",
"STEP14_NAME": "Крок 12",
"STEP15_NAME": "Результати"
}
}

```

Файл `trainer.steps.json`

```

[
  {
    "filename" : "introduction",
    "score" : 0,
    "hasScript" : true
  },
  {
    "filename" : "theory",
    "score" : 0,
    "hasScript" : true
  },
  {
    "filename" : "step1",
    "score" : 8,
    "hasScript" : true
  },
  {
    "filename" : "step2",
    "score" : 8,
    "hasScript" : true
  },
  {
    "filename" : "step3",
    "score" : 10,
    "hasScript" : true
  },
  {

```

```

    "filename" : "step4",
    "score"    : 8,
    "hasScript" : true
  },
  {
    "filename" : "step5",
    "score"    : 10,
    "hasScript" : true
  },
  {
    "filename" : "step6",
    "score"    : 8,
    "hasScript" : true
  },
  {
    "filename" : "step7",
    "score"    : 8,
    "hasScript" : true
  },
  {
    "filename" : "step8",
    "score"    : 8,
    "hasScript" : true
  },
  {
    "filename" : "step9",
    "score"    : 8,
    "hasScript" : true
  },
  {
    "filename" : "step10",
    "score"    : 8,
    "hasScript" : true
  },
  {
    "filename" : "step11",
    "score"    : 8,
    "hasScript" : true
  },
  {
    "filename" : "step12",
    "score"    : 8,
    "hasScript" : true
  },
  {
    "filename" : "results",
    "score"    : 0,
    "hasScript" : true
  }
]

```

Файл `trainer.config.json`

```

{
  "LANGUAGES": {
    "ua": "Українська",
    "ru": "Русский",

```

```

    "en": "English"
  },
  "DEFAULT_LANG": "ua",
  "TRAINER_SCORE": 100,
  "TRAINER_AUTHOR": "Semen Duvanov",
  "COURSE_AUTHOR": "Irina Baranova",
  "PRODUCTION": false
}

```

Файл `trainer.custom.css`

```

/*START OVERRIDING*/

body {
  background-image: url("/img/trainer-background.png") !important;
}

.navbar-brand .trainer-brand {
  background: url("/img/trainer-brand.png") no-repeat !important;
}

/*Start page*/
.start
{
  padding-top: 15px !important;
}

/*Overriding jumbotron class*/
.jumbotronT {
  padding-top: 0 !important;
  padding-bottom: 0px;
  margin-bottom: 0px
}

.btn
{
  box-shadow: 0px 0px 0px 0px;
}

.btn:hover
{
  box-shadow: 1px 1px 4px rgba(0, 0, 0, 0.4);
}

.bg-danger
{
  padding-left: 17px;
  padding-top: 8px;
  padding-bottom: 8px;
}

.btn-outline-warning
{
  color: #f2bc1c !important;
}

```



```
.btn-outline-warning:hover
{
  background-color: #f2bc1c !important;
  color: #ffffff !important;
}

.btn-outline-success
{
  padding: 9px 22px;
}

.navbar-brand
{
  color: #5e9fb1 !important;
  transition: all 0.3s ease;
}

.navbar-brand:hover
{
  color: #b8c4f5 !important;
  text-shadow: 1px 1px 1px #677375;
}

.nav-stacked a
{
  padding-left: 0px !important;
}

.tooltip > .tooltip-inner
{
  background-color: black;
}

.tooltip.left, .tooltip.right
{
  white-space: nowrap;
}

.tooltip.top .tooltip-arrow
{
  border-top-color: #000;
}

.tooltip.right .tooltip-arrow
{
  border-right-color: #000;
}

.tooltip.left .tooltip-arrow
{
  border-left-color: #000;
}

.tooltip.bottom .tooltip-arrow
{
  border-bottom-color: #000;
}

.alert-dismissible
{
  font-size: 18px !important;
}
```

```

div.alert-dismissible.alert-danger
{
  z-index: 2;
}

/*alert-danger background*/
.under-alert-danger
{
  display: none;
  position: absolute;
  z-index: 1;
  left: 0px;
  top: 0px;
  right: 0px;
  bottom: 0px;
  background-color: rgba(10, 10, 10, 0.61);
}

/*alert-danger between attempts*/
.validation-alert-middle
{
  display: none;
}

/*END OVERRIDING*/

/*Deactivate breadcrumb link*/
.breadcrumb li a
{
  pointer-events: none;
  cursor: default;
  color: #ffffff;
  background-color: #8e9197;
  margin-left: -3px;
  padding-left: 7px;
  border-radius: 10px;
  padding-right: 8px;
  padding-top: 3px;
  padding-bottom: 3px;
}

/*Override rotators*/
.rotators a
{
  padding: 14px 58px !important;
  text-indent: -16px !important;
}

.rotators
{
  margin-bottom: 20px !important;
}

/*Override draggable*/
div.draggable.value
{
  padding: 12px 11px !important;
  color: #ffffff !important;
  background-color: #7b7073 !important;
  border: 0px !important;
}

```

```
div.droppable
{
  padding: 21px 4px !important;
}

div.droppable.input
{
  min-width: 341px !important;
  min-height: 75px !important;
}

/*Task div*/
.task
{
  min-height: 500px;
  border-radius: 10px 0 0 10px;
  max-width: 100%;
}

.task h1,h2,h3,h4,h5
{
  line-height: 1.7;
  color: #324e52;
}

/*Left div of Task div*/
.block1
{
  padding-top: 16px;
  background-color: #dde6e6;
  min-height: 500px;
  border-radius: 10px 0 0 10px;
}

/*Right div of Task div*/
.block2
{
  display: flex;
  align-items: center;
  justify-content: center;

  min-height: 500px;
  border-radius: 0px 10px 10px 0px;
}

.block2,.block1
{
  padding-left: 44px;
  padding-right: 29px;
}

.block3
{
  padding-top: 16px;
  padding-left: 30px;
  background-color: #dde6e6;
  min-height: 500px;
  border-radius: 10px 10px 10px 10px;
}
```

```

.block4
{
  background-color: #ffffff;
  min-height: 500px;
  border-radius: 10px 10px 10px 10px;
}

.shadow
{
  box-shadow: 0px 0px 6px #a3a3a3;
}

/*results div*/
.results
{
  display: flex;
  flex-direction: column;
  align-items: center;
  background-color: #ffffff;
  border-radius: 10px 10px 10px 10px;
}

.task-dec
{
  background-color: #495662;
  margin: -27.5px -14px -13px -29px;
  padding: 30px 20px 30px 31px;
  color: white;
};

/*task's question decoration*/
.rounded-l
{
  border-radius: 10px 0px 0px 10px;
}

/*1 type of div rounding*/
.rounded-r
{
  border-radius: 0px 10px 10px 0px;
}

/*2 type of div rounding*/
.rounded-3
{
  border-radius: 10px 0 10px 10px;
}

/*3 type of div rounding*/
.rounded-4
{
  border-radius: 10px 10px 10px 10px;
}

.color-1
{
  background-color: #4b777f;
}

.color-2
{
  background-color: rgba(38, 66, 46, 0.84);
}

```

```

.color-3
{
  background-color: #ebebeb;
}

.color-4
{
  background-color: #576075;
}

/*Image style*/
.task img
{
  max-width: 102%;
  max-height: 90%;
  /* border-radius: 5px;*/
}

/*Table style*/
.task table
{
  border: 8px solid #dddddd;

  max-width: 80%;
  border-radius: 5px;
  box-shadow: 1px 1px 1px #655555;
}

/*Overriding check button style*/
.task button
{
  margin-bottom: 13px;
}

/*Task decoration line*/
.task th
{
  padding: 10px 5px 0px 0px;
}

.task-dec em
{
  color: #da8c8c;
}

hr
{
  border-top: 3px solid #9a8080;
}

h5 > hr {
  display: block;
  content: "";
  height: 30px;
  margin-top: 4px;
  margin-left: -25px;
  margin-right: -14px;
  border-style: solid;
  border-color: #a881b2;
  border-width: 0 0 2px 0;
  border-radius: 11px 11px;
}

```

```

.block3 hr,
{
  display: block;
  content: "";
  height: 30px;
  margin-top: 4px;
  margin-left: -27px;
  margin-right: -11px;
  border-style: solid;
  border-color: #a881b2;
  border-width: 0 0 2px 0;
  border-radius: 11px 11px;
}

em
{
  color: #bd4747;
}

/*Radio label style*/
label
{
  margin-left: 4px;
  margin-bottom: 17px
}

.row-flex
{
  display: -webkit-box;
  display: -moz-box;
  display: -ms-flexbox;
  display: -webkit-flex;
  display: flex;
  flex:1 1 auto;
}

.code
{
  display: block;
  min-width: 20px;
  min-height: 30px;

  padding: 14px 14px;
  font-size: 90%;
  color: #c7254e;
  background-color: #f9f2f4;
  border-radius: 3px;
}

/*'To theory' link style*/
.book-style
{
  font-size: 45px;
  color: #4b6262;

  display: block;
  position: absolute;
  top: -14px;
  right: 28px;
}

```

```

/*'Return from theory' link style*/
.return-style
{
  font-size: 45px;
  color: #ffffff;
  display: block;
  position: absolute;
  bottom: -14px;
  left: 13px;
}

.book-style:hover,.return-style:hover
{
  color: #5da2a2;
}

/*Current step in breadcrumb style*/
.isDisabled {
  color: currentColor;
  cursor: not-allowed;
  opacity: 0.5;
  text-decoration: none;
}

/*Google Chart result block style*/
#gc {
  display: block;
  box-shadow: 3px 2px 3px -2px;
}

#gc > svg
{
  width: 800px;
}

#begin
{
  display: none;
}

/*3 types of columns in result div*/
.results .column1
{
  background-color: #495662;
  padding: 20px 20px;
  width: 106%;
  border-radius: 10px;
  text-align: center;
}

.column2
{
  background-color: #f4f6f6;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 42px;
  width: 390px;
  margin-top: -6px;
  margin-bottom: -6px;

  //box-shadow: 0px 1px 1px;
}

```

```

.column3
{
  background-color: #f4f6f6;
  display: flex;
  justify-content: center;
  align-items: center;
  color: #072a41;
  height: 58px;
  min-width: 454px;
  margin-top: -15px;
  margin-bottom: 1px;
}
/**/

div.column2 > b
{
  color: #b74848;
}

/*Overriding badge style*/
div.column2 > .badge
{
  padding: 3px 9px 3px;
  font-size: 21px;
  background-color: #b74848;
}

div.column3 > b
{
  color: #b74848;
}

.results .column1 h2
{
  margin-top: 0px;
  margin-bottom: 0px;
  color: white;
}

/*Results decoration line*/
.results hr
{
  display: block;
  content: "";
  margin-left: -20px;
  margin-right: -18px;
  margin-top: 0px;
  margin-bottom: 0px;
  border-style: solid;
  border-color: #ccd7d7;
  border-width: 0 0 2px 0;
  width: 100%;
}

.theory
{
  font-size: 20px;
}

div.theory > div > ul > li > a
{

```



```

    color: #ffffff;
    background-color: #495662;
}

/*First column of theory div*/
.Tcolumn1
{
    background-color: #495662;
    border-radius: 10px 0px 0px 10px;
    padding-top: 17px;
    padding-bottom: 17px;
    height: 550px;
}

/*Second column of theory div*/
.Tcolumn2
{
    padding-top: 0;
    padding-bottom: 17px;
    height: 550px;
    overflow-y: scroll;
    color: #000000;

    font-size: 16px;
}

.Tcolumn2 table
{
    max-width: 100%;
    box-shadow: 0px 0px;
}

.Tcolumn2 th
{
    text-align: center;
    vertical-align: middle;
    padding: 10px 10px 10px 10px;
    font-weight: 400;
}

.Tcolumn2 h3
{
    font-weight: 100;
    text-indent: 5%;
    background-color: #f8f8f8;
    display: block;
    border-radius: 0px 0px 10px 10px;
}

.Tcolumn2 th span
{
    color: #890606;
    /* font-style: italic;*/
}

.Tcolumn2 h3 span
{
    font-weight: 100;
    font-style: normal;
}

```

```

.Tcolumn2 span:not(.glyphicon)
{
  color: #890606;
  font-style: italic;
  font-weight: 400;
}

.Tcolumn2 .first img
{
  max-width: 49%;
  max-height: 90%;
}

.Tcolumn2 .second img
{
  max-width: 49%;
  max-height: 90%;
}

.Tcolumn2 hr {
  display: block;
  content: "";
  height: 30px;
  margin-top: -23px;
  margin-left: -13px;
  margin-right: -14px;
  border-style: solid;
  border-color: #c0c7cb;
  border-width: 0 0 2px 0;
  border-radius: 11px 11px;
}

.Tcolumn2 .second b,.Tcolumn2 .first b,.Tcolumn2 .third b
{
  background-color: #656566;
  border-radius: 10px;
  padding: 0px 6px 0px 6px;
  color: white;
}

.Tcolumn2 .second img
{
  min-width: 100%;
}

/*Link to the top of the theory div*/
#aTop
{
  position: absolute;
  left: 0;
  right: 0;
  background-color: rgba(231, 231, 231, 0.7);
  text-align: center;
  margin-top: 15px;
  padding-top: 5px;
  padding-bottom: 5px;
  color: #606060 !important;

  transition-property: background-color, color;
}

#aTop span
{

```

```

font-style: normal;
font-weight: 100;
font-size: 20px;
padding-top: 3px;
}

#aTop:hover
{
background-color: rgba(160, 160, 160, 0.9);
color: #ffffff !important;
}

/*Animation*/

.start div:first-child img
{
border-radius: 100%;
transition: all 0.5s ease;
}
.start div:first-child img:hover
{
transform: scale(1.075);
}

```

Файл introduction.html

```

<div class="row jumbotron start">
  <div class="col-xs-5 animated zoomIn">
    
  </div>
  <div class="col-xs-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" data-toggle="tooltip" data-placement="bottom" title="ua" id="ua"></div>
      <div class="flag-icon flag-icon-ru" data-toggle="tooltip" data-placement="bottom" title="ru" id="ru"></div>
      <div class="flag-icon flag-icon-en" data-toggle="tooltip" data-placement="bottom" title="en" id="en"></div>
    </p>
  </div>
</div>

<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();
});
</script>

```

Файл introduction.js

```

var introduction = function () {
  this.preDispatch = function () {
    Scorer.start();
  };

  this.postDispatch = function () {
    $('div.flag-icon').each(function() {

```



```

<div class="tab-pane active" id="primitivesA">
  <h3>{{{TH1_HEAD}}}</h3>
  <hr>

  <p style="text-indent: 5%;">{{{TH1_TEXT1}}}</p>

  <p style="text-indent: 5%;">{{{TH1_TEXT2}}}</p>

  <table class="table-bordered table-hover first">
    <tr>
      <th nowrap><span>auxSolidBox(width, height, depth) </span> <br> <span>
auxWireBox(width, height, depth)</span>
      {{{TH1_TABLE1}}}
    </th>
    <th>
      
      
    </th>
  </tr>
  <tr>
    <th><span>auxSolidTorus(r,R)</span><br><span>auxWireTorus(r,R)</span>
      {{{TH1_TABLE2}}}
    </th>
    <th>
      
      
    </th>
  </tr>
  <tr>
    <th><span>auxSolidCube(width)</span><br><span>auxWireCube(width)</span>
      {{{TH1_TABLE3}}}
    </th>
    <th>
      
      
    </th>
  </tr>
  <tr>
    <th><span>auxSolidCylinder(r,height)</span><br><span>auxWireCylinder(r,height)
</span>
      {{{TH1_TABLE4}}}
    </th>
    <th>
      
      
    </th>
  </tr>
  <tr>
    <th><span>auxSolidCone(r,height)</span><br><span>auxWireCone(r,height)</span>
      {{{TH1_TABLE5}}}
    </th>
    <th>
      

```

```

        
    </th>
</tr>
<tr>
    <th><span>auxSolidSphere(r)</span><br><span>auxWireSphere(r)</span>
    {{{TH1_TABLE11}}}</th>
    <th>
        
        
    </th>
</tr>
<tr>
    <th><span>auxSolidIcosahedron(width)</span><br><span>auxWireIcosahedron(width)
</span>
    {{{TH1_TABLE6}}}</th>
    <th>
        
        
    </th>
</tr>
<tr>
    <th><span>auxSolidOctahedron(width)</span><br><span>auxWireOctahedron(width)</
span>
    {{{TH1_TABLE7}}}</th>
    <th>
        
        
    </th>
</tr>
<tr>
    <th><span>auxSolidTetrahedron(width)</span><br><span>auxWireTetrahedron(width)
</span>
    {{{TH1_TABLE8}}}</th>
    <th>
        
        
    </th>
</tr>
<tr>
    <th><span>auxSolidDodecahedron(width)</span><br><span>auxWireDodecahedron(wi
dth)</span>
    {{{TH1_TABLE9}}}</th>
    <th>
        

```

```

        
    </th>
</tr>
<tr>
<th><span>auxSolidTeapot(width)</span><br><span>auxWireTeapot(width)</span>
{{{TH1_TABLE10}}}</th>
</th>
<th>


</th>
</tr>
</table>

</div>

<div class="tab-pane" id="primitivesG">
<h3>{{{TH2_HEAD}}}</h3>
<hr>

<p style="text-indent: 5%;">{{{TH2_TEXT1}}}</p>

<p style="text-indent: 5%;">{{{TH2_TEXT2}}}</p>

<p style="text-indent: 5%;">{{{TH2_TEXT3}}}</p>

<table class="table-bordered table-hover second">
<tr>
<th nowrap><span>glutSolidCone(base,height, slices,stacks);</span>
{{{TH2_TABLE1}}}</th>
</th>
<th>

</th>
</tr>

<tr>
<th nowrap><span>glutSolidTorus(innerRadius,outerRadius,sides,rings)</span>
{{{TH2_TABLE2}}}</th>
</th>
<th>

</th>
</tr>

<tr>
<th nowrap><span>glutSolidSphere(radius,slices,stacks)</span>
{{{TH2_TABLE3}}}</th>
</th>
<th>

</th>
</tr>

<tr>
<th nowrap><span>glutSolidCube(size)</span>
{{{TH2_TABLE4}}}</th>

```

```

        </th>
        <th>
            
        </th>
    </tr>

    <tr>
        <th nowrap><span>glutSolidTeapot(size)</span>{{{TH2_TABLE5}}}</th>
        <th>
            
        </th>
    </tr>

    <tr>
        <th nowrap><span>glutSolidDodecahedron()</span>{{{TH2_TABLE6}}}</th>
        <th>
            
        </th>
    </tr>

    <tr>
        <th nowrap><span>glutSolidIcosahedron()</span>
{{{TH2_TABLE7}}}</th>
        <th>
            
        </th>
    </tr>

    <tr>
        <th nowrap><span>glutSolidOctahedron()</span>
{{{TH2_TABLE8}}}</th>
        <th>
            
        </th>
    </tr>

    <tr>
        <th nowrap><span>glutSolidTetrahedron()</span>
{{{TH2_TABLE9}}}</th>
        <th>
            
        </th>
    </tr>

</table>

</div>

<div class="tab-pane" id="coordinates">
    <h3>{{{TH3_HEAD}}}</h3>
    <hr>

```



```

</div>

<script>
  $(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();
  });

  $(function() {
    $('#aTop').click(function(){
      $('#div.Tcolumn2').animate({ scrollTop:0}, 'slow');
    });
  });
</script>

```

Файл theory.js

```

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

  };

  this.postDispatch = function () {

    Rotator.enableNextButton();
    Rotator.disablePrevButton();
    disableEndButton();

    hideReturnFromTheoryButton();

  };

  this.mustache = function () {
    return {

    }
  }
};

```

Файл step1.html

```

<div class="jumbotron jumbotronT page1">
  <hr>
  <h3 style="text-align: center" >{{TASK1_TITLE}}</h3>
  <hr>

<div class="container task">
  <div class="row">
    <div class="col-lg-5 col-lg-offset-1 col-xs-6 block1">
      <div class="row">
        <div class="col dec">
          <h5>
            <div class="task-dec rounded-3">
              {{{TASK1}}}
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        {{{STEP1_CHECKBOX1}}}
        {{{STEP1_CHECKBOX2}}}
        {{{STEP1_CHECKBOX3}}}
        {{{STEP1_CHECKBOX4}}}
    </h5>
</div>
</div>

<div class="row">
    <div class="col-lg-5 col-xs-6 pull-left">
        <button class="btn btn-outline-success check">{{CHECK}}</button>
    </div>
    <div class="col-xs-6 pull-right">
        <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-
toggle="tooltip" data-placement="right" title="{{TO_THEORY}}">
            <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
        </a>
    </div>
</div>

</div>

<div class="col-lg-5 block2 color-1">
    
</div>
</div>

</div>

<hr>

</div>

<script>
$(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();
});
</script>

```

Файл step1.js

```

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

    this.preDispatch = function () {

    };

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

        VStep1
            .addValidator($('input[name="step1-checkbox1"]'), true)
            .addValidator($('input[name="step1-checkbox2"]'), false)
            .addValidator($('input[name="step1-checkbox3"]'), false)
            .addValidator($('input[name="step1-checkbox4"]'), true)

            .setStrictMode(true)
            .setIgnoreCase(false)
    };
};

```

```

.enableStepFinishAlert(true)
.setPenalty(Rotator.getStepScore(2)*0.1);

Rotator.enablePrevButton();
enableEndButton();
enableTheoryLinks();

setTimeout(showReturnFromTheoryButton, 1000);
enableTheoryLinks();

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

  if(VStep1.validate() === true) // validate the validators
  {
    Rotator.disablePrevButton();
    disableEndButton();
    disableTheoryLinks();
  }
  else
  {
    if(VStep1.getAttempts() != 0)
    {
      $('div.validation-alert-middle').fadeIn();
      setTimeout(() => {'div.validation-alert-middle').fadeOut()}, 1200);
    }
    else
    {
      $('div.under-alert-danger').fadeIn();
    }
  }
});

});

this.mustache = function () {
  return {
    STEP1_CHECKBOX1: new CheckBox('step1-checkbox1')
      .setValue("ch1")
      .setLabel('{{CHECKBOX1}}')
      .render(),
    STEP1_CHECKBOX2: new CheckBox('step1-checkbox2')
      .setValue("ch2")
      .setLabel('{{CHECKBOX2}}')
      .render(),
    STEP1_CHECKBOX3: new CheckBox('step1-checkbox3')
      .setValue("ch3")
      .setLabel('{{CHECKBOX3}}')
      .render(),
    STEP1_CHECKBOX4: new CheckBox('step1-checkbox4')
      .setValue("ch4")
      .setLabel('{{CHECKBOX4}}')
      .render()
  }
}
};

```

Файл step2.html

```

<div class="jumbotron jumbotronT page2">
  <hr>
  <h3 style="text-align: center" >{{TASK2_TITLE}}</h3>
  <hr>

  <div class="container task">
    <div class="row">
      <div class="col-lg-5 col-lg-offset-1 col-xs-6 block1">
        <div class="row">
          <div class="col dec">
            <h5>
              <div class="task-dec rounded-3">
                {{{TASK2}}}
              </div>
              <hr><br>
              {{{MARK_STEP2_INPUT1}}}{ {{STEP2_INPUT}}}
            </h5>
          </div>
        </div>
        <br>
        <div class="row">
          <div class="col-xs-6 pull-left">
            <button class="btn btn-outline-success check">{{CHECK}}</button>
          </div>
          <div class="col-xs-6 pull-right">
            <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-
toggle="tooltip" data-placement="right" title="{{TO_THEORY}}">
              <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
            </a>
          </div>
        </div>
      </div>

      <div class="col-lg-5 col-xs-6 block2 color-1">
        
      </div>
    </div>
  </div>

  <hr>

</div>

<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();

  disableInputCache();
});
</script>

```

Файл step2.js

```

var VStep2;
var step2 = function () {

```

```

this.preDispatch = function () {

};

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

  VStep2
    .addValidator($('input[name="step2-input"]'), 8)

    .setStrictMode(true)
    .setIgnoreCase(false)
    .enableStepFinishAlert(true)
    .setPenalty(Rotator.getStepScore(3)*0.1);

  Rotator.enablePrevButton();
  enableEndButton();

  enableTheoryLinks();

  $('button.check').click(function () {
    VStep2.setAttemptsOnCheckButton($(this));

    if(VStep2.validate() === true)
    {
      Rotator.disablePrevButton();
      disableEndButton();
      disableTheoryLinks();
    }
    else
    {
      if(VStep2.getAttempts() != 0)
      {
        $('div.validation-alert-middle').fadeIn();
        setTimeout(() => {$('div.validation-alert-middle').fadeOut()}, 1200);
      }
      else
      {
        $('div.under-alert-danger').fadeIn();
      }
    }
  });
};

this.mustache = function () {
  return {
    STEP2_INPUT: new TextInput('step2-input')
      .render(),
  }
};

```

Файл step3.html

```

<div class="jumbotron jumbotronT page3">
  <hr>
  <h3 style="text-align: center" >{{TASK3_TITLE}}</h3>
  <hr>

```

```

<div class="container task">
  <div class="row row-flex">
    <div class="col-lg-5 col-lg-offset-1 col-xs-6 block1 rounded">
      <div class="row">
        <div class="col dec">
          <h5>
            <div class="task-dec rounded-3">
              {{{TASK3}}}
            </div>
          <hr>
          <table class=" table-bordered table-hover" cellpadding="1">
            <tr class="active">
              <th>{{{STEP3_INPUT1}}}</th>
              <th>{{{MARK_STEP3_INPUT1}}}</th>
            </tr>
            <tr class="active">
              <th>{{{STEP3_INPUT2}}}</th>
              <th>{{{MARK_STEP3_INPUT2}}}</th>
            </tr>
            <tr class="active">
              <th>{{{STEP3_INPUT3}}}</th>
              <th>{{{MARK_STEP3_INPUT3}}}</th>
            </tr>
            <tr class="active">
              <th>{{{STEP3_INPUT4}}}</th>
              <th>{{{MARK_STEP3_INPUT4}}}</th>
            </tr>
            <tr class="active">
              <th>{{{STEP3_INPUT5}}}</th>
              <th>{{{MARK_STEP3_INPUT5}}}</th>
            </tr>
          </table>
          </h5>
        </div>
      </div>
      <br>
      <div class="row">
        <div class="col-lg-6 col-xs-6 pull-left">
          <button class="btn btn-outline-success check">{{CHECK}}</button>
        </div>
        <div class="col-lg-6 col-xs-6 pull-right">
          <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-
toggle="tooltip" data-placement="right" title="{{TO_THEORY}}">
            <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
          </a>
        </div>
      </div>
      <br>
    </div>

    <div class="col-lg-5 col-xs-6 block2 color-1">
      
    </div>
  </div>

  <hr>

</div>

<script>

```



```

$(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();

    disableInputCache();
});
</script>

```

Файл step3.js

```

var VStep3;
var step3 = function () {

    this.preDispatch = function () {

    };

    this.postDispatch = function () {
        VStep3 = new Validator();

        VStep3
            .addValidator($('input[name="step3-input1"]'), 5)
            .addValidator($('input[name="step3-input2"]'), 4)
            .addValidator($('input[name="step3-input3"]'), 2)
            .addValidator($('input[name="step3-input4"]'), 3)
            .addValidator($('input[name="step3-input5"]'), 1)

            .setStrictMode(true)
            .setIgnoreCase(false)
            .enableStepFinishAlert(true)
            .setPenalty(Rotator.getStepScore(4)*0.1);

        Rotator.enablePrevButton();
        enableEndButton();

        enableTheoryLinks();

        $('button.check').click(function () {
            VStep3.setAttemptsOnCheckButton($(this));

            if(VStep3.validate() === true)
            {
                Rotator.disablePrevButton();
                disableEndButton();
                disableTheoryLinks();
            }
            else
            {
                if(VStep3.getAttempts() != 0)
                {
                    $('div.validation-alert-middle').fadeIn();
                    setTimeout(() => {$('div.validation-alert-middle').fadeOut()}, 1200);
                }
                else
                {
                    $('div.under-alert-danger').fadeIn();
                }
            }
        });
    };
};

```

```

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(),
  }
}
};

```

Файл step4.html

```

<div class="jumbotron jumbotronT page4">
  <hr>
  <h3 style="text-align: center" >{{TASK4_TITLE}}</h3>
  <hr>

  <div class="container task">
    <div class="row row-flex">
      <div class="col-lg-6 col-xs-6 col-sm-8 col-lg-offset-3 col-sm-offset-2 col-xs-offset-3 block3 color-3
rounded">
        <div class="row">
          <div class="col-lg-12">
            <h5>
              <div class="task-dec rounded-4">
                {{{TASK4}}}
              </div>

              <hr>

              {{{STEP4_CHECKBOX1}}}
              {{{STEP4_CHECKBOX2}}}
              {{{STEP4_CHECKBOX3}}}
              {{{STEP4_CHECKBOX4}}}
            </h5>
          </div>
        </div>
        <br>
        <div class="row">
          <div class="col-lg-6 col-xs-6 pull pull-left">
            <button class="btn btn-outline-success check">{{CHECK}}</button>
          </div>
          <div class="col-lg-6 col-xs-6 pull pull-right">
            <a href="#" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-
placement="right" title="{{TO_THEORY}}">
              <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
            </a>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```
</div>
```

```
<hr>
```

```
</div>
```

Файл step4.js

```
var VStep4;
var step4 = function () {

    this.preDispatch = function () {

    };

    this.postDispatch = function () {
        VStep4 = new Validator();

        VStep4
            .addValidator($('input[name="step4-checkbox1"]'), false)
            .addValidator($('input[name="step4-checkbox2"]'), true)
            .addValidator($('input[name="step4-checkbox3"]'), false)
            .addValidator($('input[name="step4-checkbox4"]'), true)

            .setStrictMode(true)
            .setIgnoreCase(false)
            .enableStepFinishAlert(true)
            .setPenalty(Rotator.getStepScore(5)*0.1);

        Rotator.enablePrevButton();
        enableEndButton();

        enableTheoryLinks();

        $('button.check').click(function () {
            VStep4.setAttemptsOnCheckButton($(this));

            if(VStep4.validate() === true)
            {
                Rotator.disablePrevButton();
                disableEndButton();
                disableTheoryLinks();
            }
            else
            {
                if(VStep4.getAttempts() != 0)
                {
                    $('div.validation-alert-middle').fadeIn();
                    setTimeout(() => {$('div.validation-alert-middle').fadeOut()}, 1200);
                }
                else
                {
                    $('div.under-alert-danger').fadeIn();
                }
            }
        });
    };
};
```

```

this.mustache = function () {
  return {
    STEP4_CHECKBOX1: new CheckBox('step4-checkbox1')
      .setValue("ch1")
      .setLabel('{{MARK_STEP4_CHECKBOX1}}')
      .render(),
    STEP4_CHECKBOX2: new CheckBox('step4-checkbox2')
      .setValue("ch2")
      .setLabel('{{MARK_STEP4_CHECKBOX2}}')
      .render(),
    STEP4_CHECKBOX3: new CheckBox('step4-checkbox3')
      .setValue("ch3")
      .setLabel('{{MARK_STEP4_CHECKBOX3}}')
      .render(),
    STEP4_CHECKBOX4: new CheckBox('step4-checkbox4')
      .setValue("ch4")
      .setLabel('{{MARK_STEP4_CHECKBOX4}}')
      .render()
  }
}
};

```

Файл step5.html

```

<div class="jumbotron jumbotronT page5">
  <hr>
  <h3 style="text-align: center;" >{{TASK5_TITLE}}</h3>
  <hr>

  <div class="container task">
    <div class="row row-flex">
      <div class="col-lg-5 col-lg-offset-1 col-xs-5 col-xs-offset-1 col-sm-6 col-sm-offset-0 block1 rounded">
        <div class="row">
          <div class="col dec">
            <h5>
              <div class="task-dec rounded-3">
                {{{TASK5}}}
              </div>
              <hr>
              {{{STEP5_DROPPABLE}}}
              <br>
              {{{STEP5_DRAGGABLE}}}
            </h5>
          </div>
        </div>
        <br>
        <div class="row">
          <div class="col-lg-6 col-xs-6 pull-left">
            <button class="btn btn-outline-success check">{{CHECK}}</button>
          </div>
          <div class="col-lg-6 col-xs-6 pull-right">
            <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-
toggle="tooltip" data-placement="right" title="{{TO_THEORY}}">
              <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
            </a>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

    <div class="col-lg-5 col-xs-5 col-sm-6 block2 color-4">
      
    </div>

  </div>
</div>

<hr>

</div>

<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();
});
</script>

```

Файл step5.js

```

var VStep5;
var step5 = function () {

  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    VStep5 = new Validator();

    VStep5
      .addValidator($('div.droppable[name="step5-droppable"]'), ['two', 'three', 'four'], false, true)

      .setStrictMode(true)
      .setIgnoreCase(false)
      .enableStepFinishAlert(true)
      .setPenalty(Rotator.getStepScore(6)*0.1);

    Rotator.enablePrevButton();
    enableEndButton();

    enableTheoryLinks();

    $('button.check').click(function () {
      VStep5.setAttemptsOnCheckButton($(this));

      if(VStep5.validate() === true)
      {
        Rotator.disablePrevButton();
        disableEndButton();
        disableTheoryLinks();
      }
      else
      {
        if(VStep5.getAttempts() != 0)
        {
          $('div.validation-alert-middle').fadeIn();
          setTimeout(() => { $('div.validation-alert-middle').fadeOut() }, 1200);
        }
      }
    });
  };
};

```

```

        else
        {
            $('div.under-alert-danger').fadeIn();
        }
    }

});

this.mustache = function () {
    return {
        STEP5_DROPPABLE: new DroppableArea('step5-droppable')
            .addClass('input')
            .render(),
        STEP5_DRAGGABLE: new DraggableGroup('step5-draggable')
            .addClass('value')
            .addOption('auxSolidSphere()', 'one')
            .addOption('auxSolidTeapot()', 'two')
            .addOption('auxSolidCylinder()', 'three')
            .addOption('auxSolidCube()', 'four')
            .addOption('auxSolidTorus()', 'five')
            .addOption('auxSolidCone()', 'six')
            .randomize() // You can randomize draggable elements
            .render()
    }
}
};

```

Файл step6.html

```

<div class="jumbotron jumbotronT page4">
    <hr>
    <h3 style="text-align: center" >{{TASK6_TITLE}}</h3>
    <hr>

<div class="container task">
    <div class="row row-flex">
        <div class="col-lg-4 col-xs-4 col-lg-offset-4 col-xs-offset-4 col-sm-6 col-sm-offset-3 block3 color-3
rounded">
            <div class="row">
                <div class="col-lg-12">
                    <h5>
                        <div class="task-dec rounded-4">
                            {{{TASK6}}}
                        </div>

                    <hr>
                    <br>
                    {{{STEP6_RADIOS}}}

                    </h5>
                </div>
            </div>
            <div class="row">
                <div class="col-lg-6 col-xs-6 pull-left">
                    <button class="btn btn-outline-success check">{{CHECK}}</button>
                </div>
                <div class="col-lg-6 col-xs-6 pull-right">

```

```

        <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-
toggle="tooltip" data-placement="right" title="{TO_THEORY}">
        <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
        </a>
    </div>
</div>

</div>

</div>
</div>

<hr>

</div>

<script>
$(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();
});
</script>

```

Файл step6.js

```

var VStep6;
var step6 = function () {

    this.preDispatch = function () {

    };

    this.postDispatch = function () {
        VStep6 = new Validator();

        VStep6
            .addValidator($('input[name="step6-radios"]'), 'three')

            .setStrictMode(true)
            .setIgnoreCase(false)
            .enableStepFinishAlert(true)
            .setPenalty(Rotator.getStepScore(7)*0.1);

        Rotator.enablePrevButton();
        enableEndButton();

        enableTheoryLinks();

        $('button.check').click(function () {
            VStep6.setAttemptsOnCheckButton($(this));

            if(VStep6.validate() === true)
            {
                Rotator.disablePrevButton();
                disableEndButton();
                disableTheoryLinks();
            }
            else
            {
                if(VStep6.getAttempts() != 0)
                {

```



```

    </div>
    <div class="col-xs-6 pull-right">
      <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-
toggle="tooltip" data-placement="right" title="{{TO_THEORY}}">
        <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
      </a>
    </div>
  </div>

  <div class="col-lg-5 col-xs-6 block2 color-1">
    
  </div>
</div>
</div>

<hr>

</div>

<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();
});
</script>

```

Файл step7.js

```

var VStep7;// A variable for future validator
var step7 = function () {

  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    VStep7 = new Validator();

    VStep7
      .addValidator($('select[name="step7-select"]'), 1)

      .setStrictMode(true)
      .setIgnoreCase(false)
      .enableStepFinishAlert(true)
      .setPenalty(Rotator.getStepScore(8)*0.1);

    Rotator.enablePrevButton();
    enableEndButton();

    enableTheoryLinks();

    $('button.check').click(function () {
      VStep7.setAttemptsOnCheckButton($(this)); //dynamically changing amount of attempts left on check
      button

      if(VStep7.validate() === true)
      {
        Rotator.disablePrevButton();

```

```

        disableEndButton();
        disableTheoryLinks();
    }
    else
    {
        if(VStep7.getAttempts()!==0)
        {
            $('div.validation-alert-middle').fadeIn();
            setTimeout(() => {'div.validation-alert-middle').fadeOut()},1200);
        }
        else
        {
            $('div.under-alert-danger').fadeIn();
        }
    }
}

});

};

this.mustache = function () {
    return {
        STEP7_SELECT: new Select('step7-select')
            .addOption('{{STEP7_OPTION_ONE}}', 0)
            .addOption('{{STEP7_OPTION_TWO}}', 1)
            .addOption('{{STEP7_OPTION_THREE}}', 2)
            .addOption('{{STEP7_OPTION_FOUR}}', 2)
            .randomize()
            .render()
    }
}
};

```

Файл results.html

```

<div class="row jumbotron jumbotronT">
    <hr>

<div class="container">
    <div class="row">
        <div class="col-lg-6 col-xs-6 col-sm-8 col-lg-offset-3 col-xs-offset-3 col-xs-offset-2 results shadow">

            <div class="column1 col-lg-12 col-xs-12"><h2>{{RESULTS_TITLE}}</h2></div>
            <br>
            <hr>
            <br><div id="gc"></div>
            <br>
            <hr>
            <h5><div class="column2"><strong>{{RESULTS_START_TIME}}</strong>
<b>{{START_TIME}}</b></div></h5>
            <hr><h5><div class="column2"><strong>{{RESULTS_END_TIME}}</strong>
<b>{{END_TIME}}</b></div></h5>
            <hr><h5><div class="column2"><strong>{{RESULTS_TIME_DIFF}}</strong> <span
class="badge">{{TIME_DIFF}} <b>{{RESULTS_TIME_SEC}}</b></span></div></h5>
            <hr>
            <div><h2><div class="column3"><strong>{{RESULTS_YOUR_SCORE}}:
</strong><b>{{RESULTS_POINTS_IN_PERCENT}}%</b></div></h2></div>

        </div>
    </div>

```

```

<div class="clearfix"></div>

</div>

</div>
<hr>
</div>

```

Файл results.js

```

var results = function () {

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

    this.postDispatch = function () {
        //you can use or not this diagram
        var g = new GoogleCharts(); // Making a call to google charts api to build a column chart

        var scores = Rotator.getAllStepScores();
        var userScores = Scorer.getUserStepScores();

        var header;
        var step_name;
        var _title;
        var _subtitle;

        // Set the language to Google Chart marks
        var header_eng = ['Step', 'Step maximum', 'Your score'];
        var step_eng = ['Introduction', 'Theory', 'Step 1', 'Step 2', 'Step 3', 'Step 4', 'Step 5', 'Step 6', 'Step 7', 'Step 8', 'Step
9', 'Step 10', 'Step 11', 'Step 12', 'Results'];
        var title_eng = 'Your results';
        var subtitle_eng = 'Step statistics for current trainer run';

        var header_ru = ['Шаг', 'Максимум шага', 'Ваш счет'];
        var step_ru = ['Вступление', 'Теория', 'Шаг 1', 'Шаг 2', 'Шаг 3', 'Шаг 4', 'Шаг 5', 'Шаг 6', 'Шаг 7', 'Шаг 8', 'Шаг
9', 'Шаг 10', 'Шаг 11', 'Шаг 12', 'Результаты'];
        var title_ru = 'Ваши результаты';
        var subtitle_ru = 'Пошаговая статистика';

        var header_ua = ['Крок', 'Максимум на кроці', 'Ваш рахунок'];
        var step_ua = ['Вступ', 'Теорія', 'Крок 1', 'Крок 2', 'Крок 3', 'Крок 4', 'Крок 5', 'Крок 6', 'Крок 7', 'Крок 8', 'Крок
9', 'Крок 10', 'Крок 11', 'Крок 12', 'Результати'];
        var title_ua = 'Ваші результати';
        var subtitle_ua = 'Крокова статистика';

        if (I18N.getCurrentLang() === 'en')
        {
            header = header_eng;
            step_name = step_eng;
            _title = title_eng;
            _subtitle = subtitle_eng;
        }
        else
        if (I18N.getCurrentLang() === 'ua')
        {
            header = header_ua;

```

```

        step_name = step_ua;
        _title = title_ua;
        _subtitle = subtitle_ua;
    }
    else
    if (I18N.getCurrentLang() === 'ru')
    {
        header = header_ru;
        step_name = step_ru;
        _title = title_ru;
        _subtitle = subtitle_ru;
    }

    var data = [
        [header[0], header[1], header[2]],
        [step_name[2], scores[2], userScores[2]],
        [step_name[3], scores[3], userScores[3]],
        [step_name[4], scores[4], userScores[4]],
        [step_name[5], scores[5], userScores[5]],
        [step_name[6], scores[6], userScores[6]],
        [step_name[7], scores[7], userScores[7]],
        [step_name[8], scores[8], userScores[8]],
        [step_name[9], scores[9], userScores[9]],
        [step_name[10], scores[10], userScores[10]],
        [step_name[11], scores[11], userScores[11]],
        [step_name[12], scores[12], userScores[12]],
        [step_name[13], scores[13], userScores[13]]
    ];

    var options = {
        chart: {
            title: _title,
            subtitle: _subtitle
        }
    };

    g.setType("bar");
    g.setLibrary("google.charts.Bar");
    g.setData(data);
    g.setOptions(options);
    g.doQuery($('div#gc'));

    if (PRODUCTION)
        Service.pushResults();

    Rotator.enablePrevButton();
    disableEndButton();

    // Disable prevButton for EarlyEnd client
    if($('div[data-step="13"]').length==0)
        Rotator.disablePrevButton();

    // Hide endButton and show a beginButton
    $('#endTrainer').hide();
    $('#begin').show();
};

this.mustache = function () {
    return {
        START_TIME: FormattedStartTime(),
        END_TIME: FormattedEndTime(),
        TIME_DIFF: Math.floor(Scorer.getTimeDifference()/60),
    };
};

```

```

        SCORE: Scorer.getScore(),
        RESULTS_POINTS: getNumEnding(Scorer.getScore(), [ '{{POINTS1}}', '{{POINTS2}}',
'{{{POINTS3}}}' ]),
        RESULTS_POINTS_IN_PERCENT: Scorer.getScoreInPercent()
    }
}

var FormattedStartTime = function (){

    var s = (Scorer.getFormattedStartTime()).split(":");
    var result = ("0" + s[0]).slice(-2) + ":" + ("0" + s[1]).slice(-2) + ":" + ("0" + s[2]).slice(-2);
    return result;

}

var FormattedEndTime = function (){

    var s = (Scorer.getFormattedEndTime()).split(":");
    var result = ("0" + s[0]).slice(-2) + ":" + ("0" + s[1]).slice(-2) + ":" + ("0" + s[2]).slice(-2);
    return result;

}

};

```

Файл customScript.js

```

var enableTheoryLinks = function () {
    $('a.book-style').removeClass('isDisabled');
    $('a.book-style').attr('onclick', 'Rotator.switchStep(1)');
};

var disableTheoryLinks = function () {
    $('a.book-style').addClass('isDisabled');
    $('a.book-style').removeAttr('onclick');
};

var hideReturnFromTheoryButton= function () {
    $('a.return-style').hide();
};

var showReturnFromTheoryButton= function () {
    $('a.return-style').show();
};

var enableEndButton = function () {
    $('#endTrainer').removeAttr('disabled','disabled').attr('data-target','#endModal');
    return this;
};

var disableEndButton = function () {
    $('#endTrainer').attr('disabled','disabled').removeAttr('data-target');
    return this;
};

var disableInputCache = function () {
    $('input[type="text"]').attr('autocomplete', 'off');
}

```