

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Онлайн-тренажер з вивчення властивостей CSS»

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

Виконавець роботи: студент групи ІТДН-61с Пономарьова Анастасія Євгенівна

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

_____ «__» _____ 2020 р.

Науковий керівник

(підпис)

к.т.н., Антипенко В.П.
(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)

Суми-2020

Сумський державний університет
Центр заочної, дистанційної та вечірньої форм навчання
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Пономарьова Анастасія Євгенівна

1 Тема роботи Онлайн-тренажер з вивчення властивостей CSS

керівник роботи Антипенко Вікторія Петрівна, к.т.н.

затверджені наказом по університету від «__» травня 2020 р. № _____

2 Строк подання студентом роботи «» червня 2020 р.

3 Вхідні дані до роботи технічне завдання на розробку онлайн-тренажеру, методичні матеріали

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування онлайн-тренажеру з вивчення властивостей CSS, розробка онлайн-тренажеру з вивчення властивостей CSS

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність проблеми, аналіз аналогів-тренажерів, мета дипломної роботи, задачі дипломного проекту, аналіз технологій, проектування онлайн-тренажеру, етапи розробки онлайн-тренажеру, висновки.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Ознайомлення з предметною областю	28.10.19 – 05.11.19	
2	Визначення в потребі веб-додатку	06.11.19 – 08.11.19	
3	Аналіз сайтів-конкурентів	11.11.19 – 19.11.19	
4	Визначення вимог додатку	20.11.19 – 28.11.19	
5	Визначення інструментарію	29.11.19 – 04.12.19	
6	Планування WBS	05.12.19 – 06.12.19	
7	Планування OBS	09.12.18 – 10.12.19	
8	Складання календарного плану	11.12.19 – 12.12.19	
9	Визначення ризиків	13.12.19 – 17.12.19	
10	Створення каркасу онлайн-тренажеру	18.12.19 – 26.12.19	
11	Оформлення дизайну сторінки	27.12.19 – 06.01.20	
12	Проектування бази даних	07.01.20 – 10.01.20	
13	Розробка функціоналу тренажеру	13.01.20 – 21.02.20	
14	Тестування розробником	24.02.20 – 24.02.20	
15	Тестування незалежною особою	25.02.20 – 25.02.20	
16	Оформлення супровідної документації	26.02.20 – 05.03.20	
17	Архівація проекту	06.03.20 – 09.03.20	
18	Введення в експлуатацію	10.03.20 – 10.03.20	

Студент _____
(підпис)

Пономарьова А.Є.

Керівник роботи _____
(підпис)

к.т.н., Антипенко В.П.

РЕФЕРАТ

Тема дипломної роботи «Онлайн-тренажер з вивчення властивостей CSS».

Дипломна робота складається зі вступу, трьох розділів, висновка, списку використаної літератури та додатків.

Пояснювальна записка містить с., рис., табл., додатків, джерел.

Перший розділ присвячений дослідженню актуальності проблеми: аналіз існуючих аналогів, формування мети дипломного проекту та задач, аналіз технологій для реалізації онлайн-тренажеру.

У другому розділі виконується проектування онлайн-тренажеру з використанням діаграм нотації IDEF0 та діаграми варіантів використання.

Останній розділ присвячений процесу реалізації онлайн-тренажеру: визначення схеми тренажеру, проектування бази даних, формування каркасу онлайн-тренажеру, розробка функціональної частини.

Результатом проведеної роботи є розроблений онлайн-тренажер, який дозволяє користувачам системи детально вивчити властивості CSS в інтерактивній формі.

Ключові слова: ОНЛАЙН-ТРЕНАЖЕР, AJAX, JAVASCRIPT, HTML, CSS.

ЗМІСТ

ВСТУП.....	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Актуальність проблеми.....	8
1.2 Аналіз існуючих аналогів	10
1.3 Мета та задачі тренажеру «CSStarting».....	14
1.4 Аналіз технологій розробки та вибір засобів реалізації проекту	16
2. ПРОЕКТУВАННЯ ОНЛАЙН-ТРЕНАЖЕРУ «CSSTARTING».....	20
2.1 Діаграми нотації IDEF0.....	20
2.2 Use Cases Діаграма	25
3. РЕАЛІЗАЦІЯ ОНЛАЙН-ТРЕНАЖЕРУ «CSSTARTING».....	27
3.1 Визначення структури онлайн-тренажеру	27
3.2 Проектування та реалізація бази даних.....	30
3.3 Розробка інтерфейсу.....	32
3.4 Розробка програмного коду.....	39
ВИСНОВКИ.....	42
СПИСОК ЛІТЕРАТУРИ.....	44
ДОДАТОК А.....	46
ДОДАТОК Б	49
ДОДАТОК В.....	58
ДОДАТОК Г	77
ДОДАТОК Д.....	82

ВСТУП

На сьогоднішній день у час бурхливого розвитку інформаційних технологій користувач стає залежним від використання онлайн-ресурсів, онлайн-тренажерів та інших подібних технологій для розвитку власної професійної діяльності.

Звичайні підручники стають не актуальними через відсутність наочної інтерактивності, яка на даний час набирає бурхливого зросту популярності серед користувачів. У професії програміст невід'ємною складовою є використання онлайн-курсів для поліпшення власної кваліфікації, отримання та закріплення нових знань. Більшість із цих онлайн-ресурсів містять у своєму складі наступну інтерактивну частину:

- динамічні відео-ролики з ігровим моментом;
- динамічні тести;
- динамічне формування коду та онлайн компіляція.

Згідно цьому на даний час широко популярною стає професія веб-дизайнер. Вона містить наступні навички:

- володіння останніми веб-технологіями;
- володіння художніми якостями;
- вміння слідкувати за останніми змінами в розробці й оформленні веб-сайтів.

На даний час не є доцільним використання звичайної, «паперової» літератури. У зв'язку з цим більшість веб-дизайнерів використовують такі ресурси як онлайн-посібники або онлайн-тренажери.

Онлайн-тренажер представляє систему в мережі Інтернет, що надає можливість навчатися користувачу самостійно або під наглядом вчителя.

Даний вид навчання як звичайних користувачів, так і тих, що прагне поліпшити власну кваліфікацію, має попит у розширенні теоретичної бази та методів навчання, оскільки сучасний світ – це час постійного розвитку технологій

та їх можливостей. Тому користувачі повинні мати актуальну навчальну базу для повноцінного та систематичного вивчення змін в ІТ-сфері.

На основі аналізу ринку аналогів було сформульовану мету дипломної роботи: розробка онлайн-тренажера для вивчення властивостей CSS з можливістю мобільності.

Для досягнення мети необхідно вирішити наступний перелік задач:

- проаналізувати предметну область;
- розробити структуру онлайн-тренажеру;
- розробити онлайн-тренажер;
- додати матеріальну базу до тренажеру;
- провести тестування розробленого продукту.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність проблеми

Світ веб-розробки постійно змінюється. Досить важливим є уважно стежити за всім, що відбувається: актуальність існуючих фреймворків, вибір оптимального серед них, тенденції розробки та оформлення сайтів та інші.

Розробнику важливо бути в курсі поточних технологічних розробок, наскільки це можливо. Він повинен дивитися на можливості, але також враховувати можливі ризики, пов'язані з новими технологіями. Регулярні експерименти з різними структурами допомагають відчувати, як потенційні вигоди, так і обмеження.

Робота розробника полягає головним чином у вирішенні технічних завдань, і тому досить важливо працювати з перевіреною технологією, з якої продуктивно працювати, і за нею стоїть велика спільнота [1].

У двадцять першому столітті велику роль грає інформація та майже жодна галузь не працює без використання інформаційних технологій. У зв'язку з цим виникає постійний попит на розробку та використання новітніх програм в процесі діяльності підприємств, як старих, так і нових [2].

Нерідко ІТ-компанії вибирають працівників, ґрунтуючись на конкретному комплекті необхідних знань, умінь і здібностей. Однак, навіть у разі, якщо здобувач придбав пізнання в курсах, дане не означає, що він може їх використовувати. Неможливо навчитися вести автомобіль у разі, якщо не сідати за кермо [3].

Фундаментальні знання ІТ-фахівець може отримати, навчаючись у вищому навчальному закладі (ВНЗ) на спеціальності «Комп'ютерні науки» або іншій спорідненій. Але також навчання у ВНЗ дає найголовніше вміння – самостійно знаходити та дослідити новий матеріал. Більшість студентів оцінюють курси як місце для отримання не знань, а поштовху для пошуку престижної роботи. Проте

насправді більшість із них приведуть студента-слухача до конкретного проміжного фінішу, що завершиться сертифікатом. Найголовніша цінність будь-якого курсу – це не працевлаштування, а розвиток у студента критичного мислення та вміння думати не шаблонами, а опираючись на власні навички та досвід [4].

У зв'язку з високим попитом на онлайн-курси на просторах мережі Інтернет створюється велика кількість онлайн-тренажерів та посібників. Перевага даних ресурсів полягає в тому, що сам користувач вирішує, коли йому зручно витратити час на виконання того чи іншого завдання. Онлайн-курси слідує такому принципу як дотримання часових рамок задачі завдання. У свою чергу, онлайн-тренажер/посібник не містять обмежень у часі.

На сьогоднішній день існує велика кількість тренажерів різного формату, серед яких наступні:

- тренажери для отримання навиків сліпого введення;
- тренажери для вивчення правил дорожнього руху;
- тренажери для вивчення іноземних мов;
- тренажери для вивчення мов програмування.

Зараз на просторах Всесвітньої мережі існує велика кількість тренажерів для користувачів, які прагнуть вивчити основи програмування та підвищити власну кваліфікацію. Проте більшість із них мають певний наступний перелік недоліків:

- не зручний дизайн;
- відсутність логіки формування завдань;
- відсутність мобільної версії тренажеру;
- інсталювання допоміжних плагінів для роботи тренажерів.

Тому майбутній онлайн-тренажер необхідно розробити у форматі адаптивного веб-сайту, де користувач зможе обрати як тренажер для освоєння теоретичного матеріалу, так і тренажер із динамічними підказками або перейти до виконання практичного завдання для закріплення знань теоретичного матеріалу.

Даний проект буде нести соціально-економічну цінність, оскільки його можна використовувати в навчально-виховних закладах у курсах, пов'язаних з

веб-розробкою. Онлайн-тренажер «CSStraining» можна використовувати як допоміжний ресурс для закріплення знань студентів як очного, так і заочного відділення.

1.2 Аналіз існуючих аналогів

Перед початком розробки онлайн-тренажеру необхідно перш за все провести аналіз існуючих аналогів.

На сьогоднішній день, не дивлячись на великий асортимент тренажерів, користувачу досить складно відшукати необхідний, який буде задовольняти його потреби. Перш за все, проблема в тому, що при пошуку тренажерів, більшість запитів пошукової системи пропонують онлайн-курси. Їх недоліки – це платна основа використання, встановленні часові рамки на виконання завдання та ризик психологічного впливу на користувача курсів із боку тьюторів.

Проте існують онлайн-тренажери, які виступають у ролі інтерактивного посібника, у якому користувач самостійно контролює використання часу.

Першим аналогом онлайн-тренажеру з вивчення CSS є HTML academy. HTML academy – це платформа інтерактивних онлайн-курсів по HTML і CSS. Конвеєр з підготовки верстальників. Розробники даної платформи стверджують, що мета курсів – зробити навчання максимально цікавим, яке затягує. «Ми з самого початку вчимо працювати з живим кодом і самостійно писати код, вирішуючи завдання, наближені до реальних. Також учневі періодично доводиться проходити випробування» [5]. Проте, виконавши аналіз відгуків від користувачів платформи, було отримано наступні дані:

- відсутність отримання цілісних знань;
- відсутність оновлення матеріальної бази;
- хаотичне викладання матеріалу;
- відсутність мобільної версії.

На рисунку 1.1 представлена початкова сторінка HTML academy.

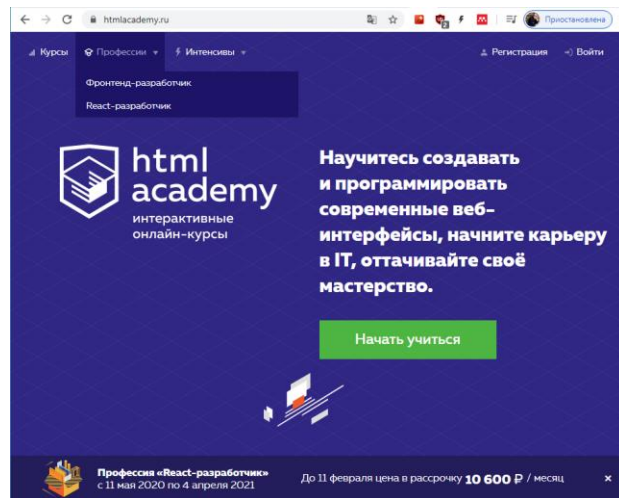


Рисунок 1.1 – Початкова сторінка HTML academy

Дана платформа передбачає реєстрацію користувача. Проте, останній може оминуть даний етап та перейти відразу до навчання, натиснувши кнопку «Почати навчання». Після цього користувачу відкривається завдання в інтерактивній формі з супровідною теорією. Панель теоретичного матеріалу до завдання представлена на рисунку 1.2.

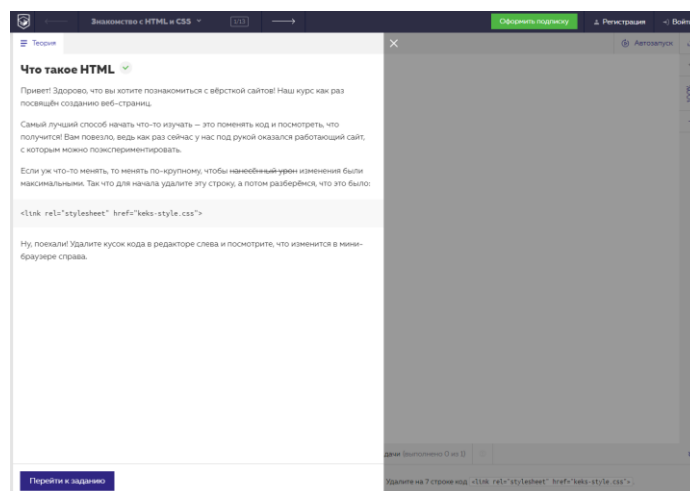


Рисунок 1.2 – Теоретичний блок до завдання

Далі користувач переходить до виконання завдання. Воно сформовано як для школярів. Не зрозуміло, коли почнеться навчання саме з CSS, де можна

переглянути повний список тем навчання тощо. Дана платформа не є зручною для самостійного вивчення матеріалу через відсутність систематичності викладення матеріалу.

Наступний аналог розглянутих онлайн-тренажерів – CSS3 Perspective Playground. CSS3 Perspective Playground – це онлайн шпаргалка для більш наочного вивчення трансформації і перспективи в CSS3 (властивість `transform: perspective`) від Mehmet Burak Erman.

«Дана властивість задає перспективу в тривимірному просторі, створюючи для користувача ілюзію глибини. Тобто елемент виглядає немов він знаходиться десь в об’ємному просторі, прямі кути здаються зовсім не прямими, то що знаходиться ближче до нас – буде здаватися більше, а то, що знаходиться далі буде виглядати менше.» – розробники тренажеру [6].

Дана розробка позиціонує себе як онлайн-ресурс, проте на офіційному сайті девелопери надаються лише можливість завантажити даний тренажер. Отриманий архів містить веб-сторінку та супровідні файли коду до ресурсу. Для нього відкривається початкова та єдина сторінка тренажеру, яка представлена рисунку 1.3.

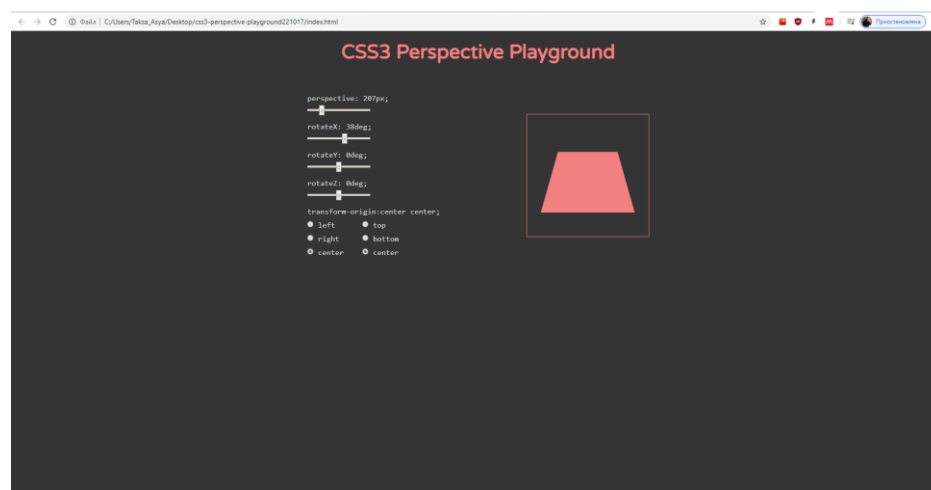


Рисунок 1.3 – Тренажер «CSS3 Perspective Playground»

Дане зображення демонструє, який функціонал має користувач для вивчення CSS властивостей. Недоліки даного тренажеру наступні:

- відсутність теоретичного матеріалу;

– відсутність додаткових практичних завдань.

Останнім розглянутим аналогом є онлайн-тренажер – Flexbox froggy. Даний тренажер розроблений у форматі онлайн-гри, у якій протягом двадцяти чотирьох рівнів необхідно розташовувати жаб на відповідні їм за кольором листя. У даній грі екран браузера розділений на дві частини. У лівій частині необхідно вписувати CSS властивості, а в правій – в інтерактивній формі жаби будуть рухатися за вказаним кодом [7]. Дана форма навчання є легкою та привабливою, проте є ризик того, що користувач отримає досить малу кількість практичного матеріалу через відсутність теоретичної бази. На рисунку 1.4 представлена початкова сторінка розглянутого тренажеру.

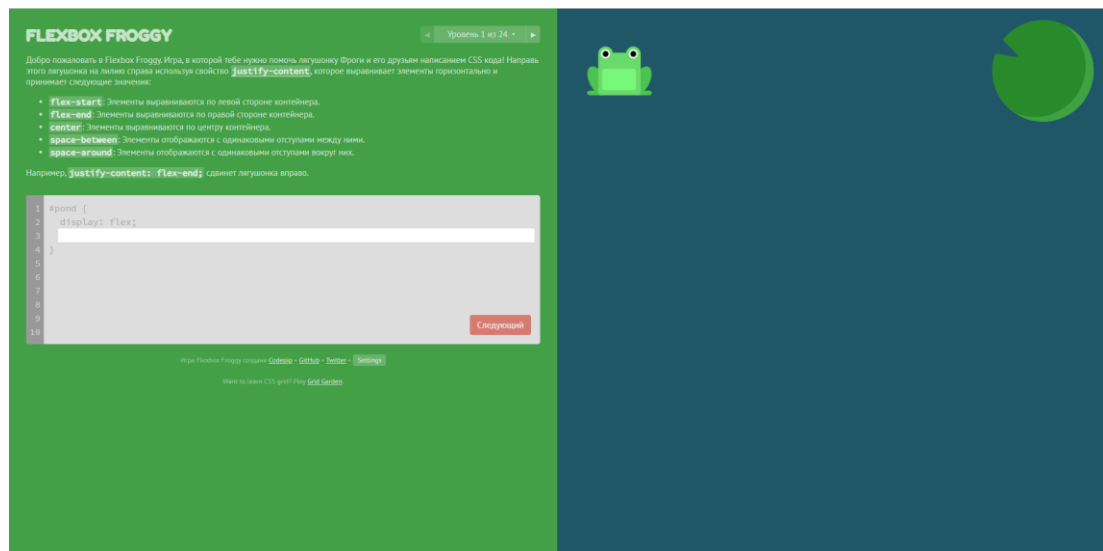


Рисунок 1.4 – Тренажер «Flexbox froggy»

На основі проведеного аналізу аналогів, була сформована порівняльна таблиця розглянутих прикладів та майбутнього проекту. Результати аналізу представлено у таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика аналогів

Назва критерію	Назва ресурсу
----------------	---------------

	HTML academy	CSS3 Perspective Playground	Flexbox froggy	CSStraining
1. Використання ресурсу без реєстрації	+	+	+	+
2. Наявність теоретичного матеріалу	+	-	-	+
3. Систематичність матеріалу	-	-	-	+
4. Адаптивність сайту під мобільну версію	-	-	+	+

1.3 Мета та задачі тренажеру «CSStarting»

Метою дипломного проекту є розробка онлайн-тренажеру для вивчення теоретичного матеріалу стосовно CSS. Також розроблений продукт дозволить закріпити отримані знання користувачі в інтерактивній формі, за рахунок практичних завдань та тестів. Даний продукт буде мати попит у діяльності онлайн-шкіл/курсів, які спеціалізуються на випуску ІТ-фахівців в галузі front-developer.

При подальшій підтримки та модернізації онлайн-тренажеру існує можливість додати функціонал та інші матеріали для вивчення: основи роботи з HTML5, JS, React та інші технології, які мають попит та є актуальними в області front-development.

Розроблений тренажер повинен підтримувати наступні функції використання:

- можливість перегляду теоретичного матеріалу;
- можливість проходження практичних завдань;
- можливість проходження інтерактивних тестів.

Функціональні вимоги до веб-додатку є такі:

- адаптивність сайту;
- наявність форми реєстрації/входу;
- наявність інтерактивних/анімованих елементів.

Нефункціональні вимоги є наступними:

- актуальний дизайн;
- зрозумілі навігація та процес проходження навчання;

Розроблений тренажер «CSStarting» повинен бути розроблений за останніми тенденціями дизайну сайтів, оскільки тренажер повинен мати попит серед сучасних користувачів-початківців, та в адаптивній формі, оскільки девелопер повинен пам'ятати про той фактор, що більшість користувачів велику частину власного часу проводять за мобільним телефоном, ніж за комп'ютером\ноутбуком.

Для реалізації поставленої мети в дипломному проекті необхідно реалізувати наступні задачі:

- провести аналіз предметної області;
- провести аналіз технологій розробки тренажеру;
- розробити інтерфейс тренажеру;
- провести проектування бази даних;
- виконати підключення бази даних до тренажеру;
- провести тестування коректної роботи тренажеру.

Для покоркового та коректного процесу реалізації тренажеру з урахуванням всіх вимог до функціоналу було розроблено технічне завдання, що наведено в Додатку А.

1.4 Аналіз технологій розробки та вибір засобів реалізації проекту

Процес розробки інтерактивних сайтів включає наступні етапи:

- розробка макету сайту;
- верстка шаблону;
- програмування інтерактивних елементів;
- наповнення сайту контентом;

Інтерактивність сайту полягає в наступних процесах:

- підтримка зв'язку між адміністратором та користувачем;
- наявність інтерактивних елементів для наочності контенту [8].

У зв'язку з вище зазначеними етапами розробки та необхідність наявності вище згаданих елементів, було прийнята рішення про використання наступних технологій:

- HTML та CSS
- JavaScript
- PHP
- MySQL
- AJAX

HTML – це комп'ютерна мова, розроблена для створення веб-сайтів. Сайти, спроектовані мовою HTML, можуть бути переглянуті на будь-якому електронному носії, що мають підключення до Інтернет-мережі. Також технологія HTML є досить легкою у вивченні для будь-якого початкового користувача. Розвиток HTML не стоїть на місці, тому існує велика кількість ресурсів для вивчення тих чи інших аспектів використання можливостей HTML [9].

HTML використовують для наступних операцій.дій:

- створення структурованого документа з наявністю абзаців, списків, таблиць;
- створення модальних форм;

- інтегрування звуку, зображень, відео до веб-сторінки.

CSS розшифровується як каскадні таблиці стилів. Хоча HTML використовується для структури веб-документа (визначаючи такі речі, як заголовки та абзаци, і дозволяючи вставляти зображення, відео та інші медіа), CSS визначає стиль спроектованого документа – макети сторінки, кольори та шрифти визначаються за допомогою CSS [10].

CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерів та ін.). Переваги використання CSS є наступними:

- розмежування коду і оформлення;
- різне оформлення для різних пристроїв;
- розширені в порівнянні з html способи оформлення елементів;
- прискорення завантаження сайту;
- єдине стильове оформлення безлічі документів;
- централізоване зберігання [11].

JavaScript – це сценарій або мова програмування, що дозволяє реалізовувати наступні складні функції на веб-сторінках:

- відображення своєчасного оновлення вмісту;
- підключення інтерактивних карт;
- підключення анімації 2D/3D-графіки;
- прокрутка відео роботів [12].

JavaScript не забезпечує низькорівневий доступ до пам'яті чи процесора, оскільки будь-який браузер підтримує синтаксис даної мови програмування.

Можливості JavaScript значною мірою залежать від середовища, в якому він працює. JavaScript у браузері виконує наступне:

- реагує на дії користувача: виконання натискання миші, руху вказівника, натискання клавіш;
- надсилає запити по мережі на віддалені сервери, завантажує та вивантажує файли;

- отримає та встановлює файли cookie;
- запам'ятовує дані на стороні клієнта («локальне зберігання») [13].

PHP – сценарійна мова з відкритим кодом, яка використовується для веб-розробки та може бути вбудована у веб-сторінку [14].

Причини використання PHP є такими:

- PHP є відкритим і безкоштовним кодом;
- коротка крива навчання порівняно з іншими мовами, такими як JSP, ASP тощо;
- постійне оновлення бібліотек мови програмування [15].

MySQL – підтримувана Oracle система керування базами даних з відкритим вихідним кодом (RDBMS), заснована на мові структурованих запитів (SQL).

MySQL дозволяє зберігати дані й отримувати до них доступ через кілька механізмів зберігання, включаючи InnoDB, CSV і NDB. Користувачі MySQL не зобов'язані вивчати нові команди; вони можуть отримати доступ до своїх даних за допомогою стандартних команд SQL [16].

MySQL має наступні переваги у використанні:

- підвищення продуктивності програм;
- висока швидкість процедури зберігання даних;
- портативність MySQL;
- безпека процедури зберігання даних [17].

AJAX – це асинхронний JavaScript, мова опису набору методів розробки, які використовуються для створення інформаційних інтерактивних ресурсів. За словами веб-розробника і інструктора WordPress по Skillcrush Енн Каскарано, кращий спосіб зрозуміти AJAX – почати з визначення його конкретної мети в процесі веб-розробки. Основна функція AJAX полягає в асинхронному оновленні веб-вмісту («А» в AJAX), тобто веб-браузеру користувача не потрібно перезавантажувати всю веб-сторінку, коли потрібно змінити тільки невелику частину вмісту на сторінці [18].

Аjax використовує XHTML для вмісту, CSS для презентації, а також Модель об'єкта документа та JavaScript для динамічного відображення вмісту. Звичайні

веб-додатки передають інформацію до та від серйозних, використовуючи синхронні запити. Це означає, що користувач заповнює форму, натискає надіслати та направляє на нову сторінку нову інформацію з сервера. Із AJAX, коли користувач натисне надіслати, JavaScript подасть запит на сервер, інтерпретує результати та оновить поточний екран. У чистому сенсі користувач ніколи не дізнається, що щось навіть передається на сервер [19].

2. ПРОЕКТУВАННЯ ОНЛАЙН-ТРЕНАЖЕРУ «CSSTARTING»

Провівши детальний аналіз предметної області:

- вирішення питання актуальності проблеми;
- експертний аналіз тренажерів-аналогів;
- визначення мети дипломного проекту та формування переліку задач;
- аналіз існуючих технологій для реалізації проекту

розробник проекту перейшов до наступного етапу роботи – проектування онлайн-тренажеру. Під час проектування необхідно було розробити діаграми нотації IDEF0 для представлення процесу розробки проекту та розробити діаграму використання для демонстрації сценаріїв дій акторів відносно тренажеру. У Додатку Б детально описаний процес планування робіт для реалізації проекту.

2.1 Діаграми нотації IDEF0

IDEF0 – методологія функціонального моделювання та графічна нотація для представлення бізнес-процесів. Особливість IDEF0 полягає в тому, що в даній нотації розглядається логічне відношення та послідовність робіт без часової залежності.

Модель IDEF0 – це набір блоків, кожний із яких представляє собою «чорний ящик», що має входи, виходи, механізми та управління. Кожний блок деталізується до рівня, який необхідний розробнику [20].

Перший блок нотації IDEF0 – це діаграма А-0, що дає загальне представлення процесу, який буде у майбутньому детально описаний під рівнями. Для проекту «Онлайн-тренажер з вивчення властивостей CSS» контекстна діаграма А-0 складається даних, які описані нижче:

- вхідні дані: технічне завдання, методичні матеріали;
- вихідні дані: онлайн-тренажер;
- управління: методологія створення онлайн-тренажерів, загальноприйняті принципи макетування;
- механізми: види схем розробки тренажерів, концептуальна модель бази даних, MySQL, HTML, CSS, PHP, AJAX, JavaScript.

На рисунку 2.1 представлена контекстна діаграма А-0.

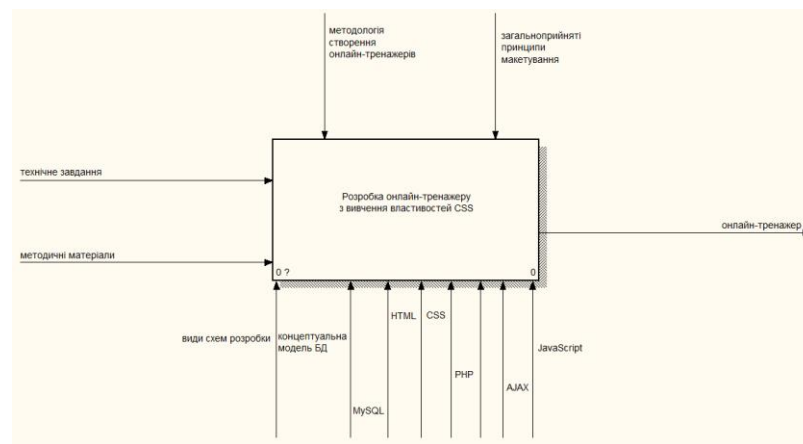


Рисунок 2.1 – Діаграма першого рівня

Оскільки контекстна діаграма присвячена процесу розробки онлайн-тренажеру, далі необхідно було визначити ключові етапи даного процесу. Оперуючи методологією розробки онлайн-тренажерів, розробником проекту було вирішено, що складові етапи процесу створення тренажеру є такими:

- проектування онлайн-тренажеру;
- реалізація онлайн-тренажеру.

Для етапу «Проектування онлайн-тренажеру» були визначені дані, що описані нижче:

- вхідні дані: технічне завдання;
- вихідні дані: база даних онлайн-тренажеру;
- управління: загальноприйняті принципи макетування;

– механізми: види схем розробки тренажерів, концептуальна модель бази даних, MySQL.

Аналогічним чином були визначені дані для блоку «Реалізація онлайн-тренажеру»:

- вхідні дані: технічне завдання, методичні дані;
- вихідні дані: онлайн-тренажер;
- управління: методологія створення онлайн-тренажерів, база даних онлайн-тренажеру;
- механізми: HTML, CSS, PHP, AJAX, JavaScript.

На рисунку 2.2 представлена діаграма другого рівня деталізації.

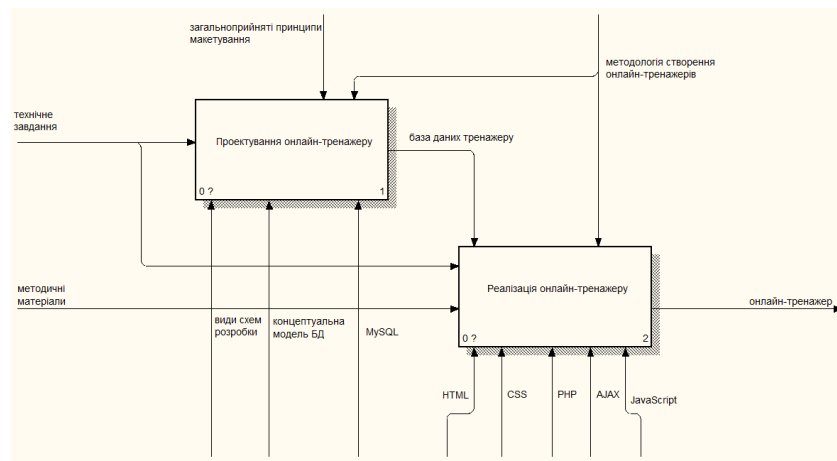


Рисунок 2.2 – Діаграма другого рівня

Після того, як було здійснено деталізацію другого рівня, розробником проекту було виконано розбиття кожного процесу на підрівні для повноцінного представлення процесу розробки онлайн-тренажеру.

Під час процесу проектування онлайн-тренажеру необхідно було провести аналіз схем розробки онлайн-ресурсів та обрати оптимальну схему, яка б задовольняла всі вимоги. Також на даному етапі необхідно було провести проектування бази даних та виконати її реалізацію для подальшого використання під час розробки тренажеру.

Із огляду на етапи роботи під час проектування тренажеру були визначені такі процеси:

- визначення структури тренажеру;
- проектування бази даних.

Для процесу «Визначення структури тренажеру» розробником проекту були зазначені такі дані:

- вхідні дані: технічне завдання;
- вихідні дані: визначена схема тренажеру;
- управління: загальноприйняті принципи макетування, методологія створення онлайн-тренажерів;
- механізми: види схем розробки тренажерів.

Для процесу «Проектування бази даних» аналогічно були визначені дані, які зазначені нижче:

- вхідні дані: технічне завдання;
- вихідні дані: база даних онлайн-тренажеру;
- управління: визначена схема тренажеру, загальноприйняті принципи макетування, методологія створення онлайн-тренажерів;
- механізми: концептуальна модель бази даних, MySQL.

На рисунку 2.3 представлена діаграма деталізації процесу «Проектування онлайн-тренажеру».

Аналогічним чином було проведено деталізацію процесу «Реалізація онлайн-тренажеру». Даний процес складається з двох ключових етапів:

- розробка каркасу онлайн-тренажеру;
- підключення функціональної частини.

Для блоку «Розробка каркасу онлайн-тренажеру» були визначені дані, які перелічені нижче:

- вхідні дані: технічне завдання;
- вихідні дані: каркас онлайн-тренажеру;
- управління: методологія створення онлайн-тренажерів;
- механізми: HTML, CSS, PHP, AJAX.

За аналогією були визначені дані для блоку «Підключення функціональної частини»:

- вхідні дані: технічне завдання, методичні матеріали;
- вихідні дані: онлайн-тренажер;
- управління: методологія створення онлайн-тренажерів, каркас онлайн-тренажеру, база даних онлайн-тренажеру;
- механізми: AJAX, JavaScript.

На рисунку 2.4 представлена діаграма деталізації процесу «Реалізація онлайн-тренажеру».

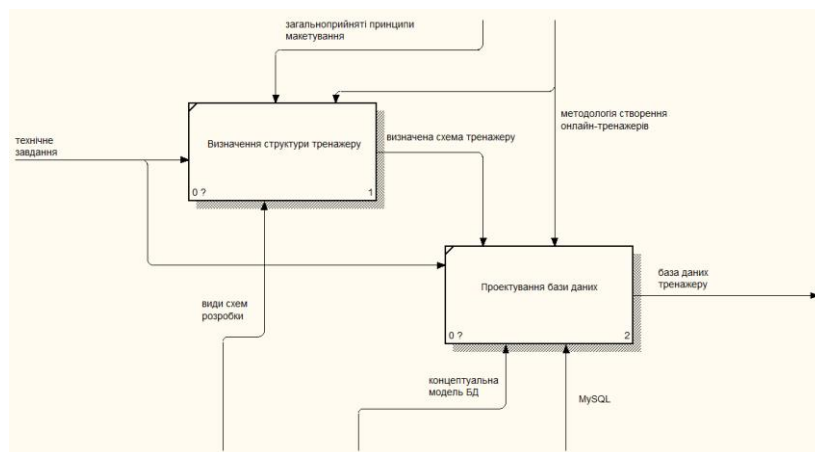


Рисунок 2.3 – Деталізація процесу «Проектування онлайн-тренажеру»

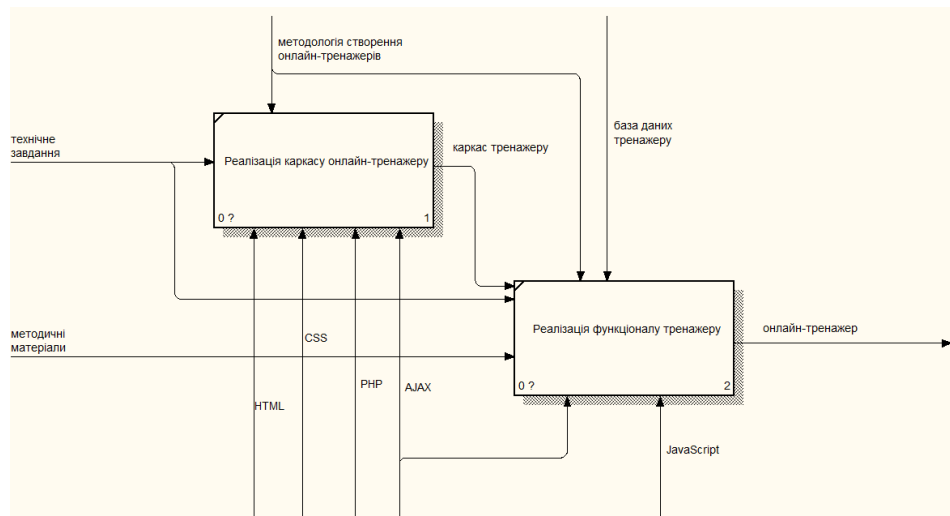


Рисунок 2.4 – Деталізація процесу «Реалізація онлайн-тренажеру»

2.2 Use Cases Діаграма

Діаграма варіантів використання (use case diagram) описує взаємовідносини та залежності між варіантами використання та акторами, які беруть участь у процесі.

Актором виступає дієве обличчя, яке є зовнішнім явищем та взаємодіє із системою, використовуючи певний сценарій. Акторами можуть виступати як реальні фізичні особи (клієнт, адміністратор), так і не фізичні суб'єкти (допоміжні програмні засоби) [21].

Для проекту «Онлайн-тренажер з вивчення властивостей CSS» були визначені наступні актори:

- Користувач – суб'єкт, який виконує реєстрацію/авторизацію для використання тренажеру в рамках навчання;
- Адміністратор – суб'єкт, який виконує реєстрацію/авторизацію для редагування вмісту тренажеру;
- База даних – суб'єкт, який виконує запис про реєстрацію або авторизацію кожного користувача тренажеру.

На рисунку 2.5 представлена діаграма використання проекту «Онлайн-тренажер з вивчення властивостей CSS»

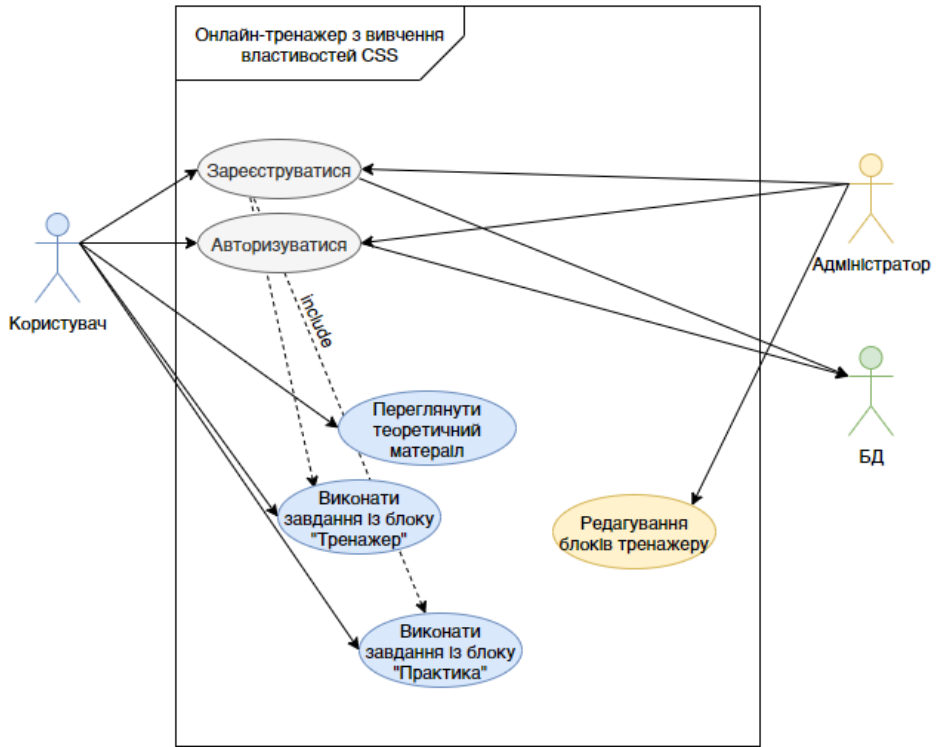


Рисунок 2.5 – Use case diagram

3. РЕАЛІЗАЦІЯ ОНЛАЙН-ТРЕНАЖЕРУ «CSSTARTING»

3.1 Визначення структури онлайн-тренажеру

Будь-який онлайн-ресурс має власну структуру розміщення інформації та елементів для зручної навігації. Онлайн-тренажер «CSStarting» розроблений за останніми тенденціями в сфері UI/UX дизайну. Тренажер має зручну та інтуїтивно зрозумілу навігацію, усі сторінки ресурсу мають загальний корпоративний стиль.

У ході реалізації тренажеру розробником було розглянуто структуру сторінки на двох рівнях: зовнішній (логічний) та внутрішній (фізичний).

Логічна структура побудована з урахуванням різного рівня компетентності користувачів, оскільки при невдалій побудові логічної структури додатку, користувачі стикаються з такими проблемами:

- відсутність комфортного пошуку інформації;
- втрата бажання використовувати ресурс;
- відсутність комфортної роботи.

Для онлайн-тренажеру «CSStarting» був розроблений інтуїтивно зрозумілий та привабливий дизайн. Зовнішня структура тренажеру зображена на рисунку 3.1.

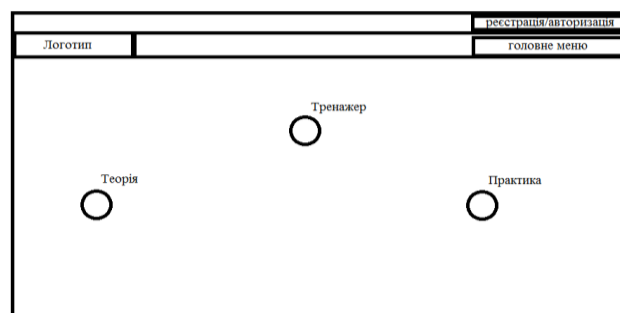


Рисунок 3.1 – Зовнішня структура онлайн-тренажеру

Структура онлайн-тренажеру розроблена таким чином, щоб користувач з перших хвилин його експлуатації мав повноцінне уявлення, яку саме інформацію він може знайти та де шукати. Логічна структура тренажеру розроблена за

принципом «онлайн-книга». Користувач знає, що його буде чекати на наступній сторінці та зможе повернутися до попередніх пройдених сторінок.

При розробці внутрішньої структури онлайн-тренажеру було розглянуто такі схеми:

- лінійна;
- ієрархічна;
- мережева;
- комбінована.

Лінійна структура – структура, при якій сторінки посилаються один на одного й одночасно на головну. Цей тип структури відмінно підходить для портфоліо, презентаційних ресурсів ті інших специфічних Інтернет-майданчиків, метою яких є послідовне ознайомлення користувачів із усім наявним контентом. Увага читача тут переходить від головної сторінки до останньої, проходячи через всі інші. Мінус такої схеми в тому, що успішному просуванню підлягає лише головна сторінка. А значить, залучати нових відвідувачів буде важко. Лінійна структура більше підходить тим, для кого не принципово отримувати трафік з пошуку [22]. На рисунку 3.2 схематично зображена лінійна структура веб-ресурсу.

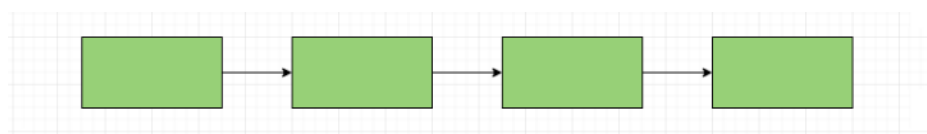


Рисунок 3.2 – Схема лінійної структури

Ієрархічна структура – схожий тип з лінійною структурою. Вона дозволяє застосовувати одночасно кілька продуктів. Найпоширеніший приклад такої схеми – онлайн-бібліотека одного учасника з його творами. Читач, знову ж таки, буде рухатися від головної сторінки. Для SEO-просування дана структура сайтів мережі Інтернет не підходить [23]. На рисунку 3.3 представлена схема ієрархічної структури веб-ресурсу.

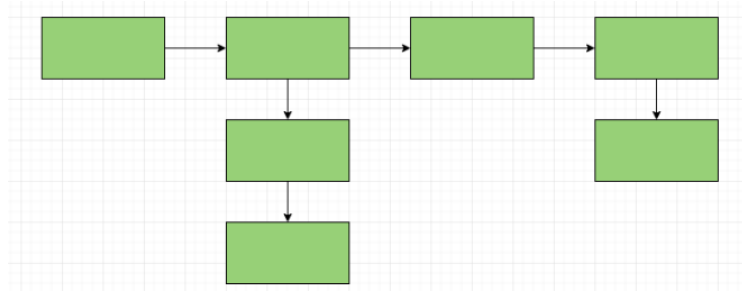


Рисунок 3.3 – Схема ієрархічної структури

Мережева структура – структура, при якій кожна сторінка посилається на кілька інших, рівнозначних між собою. Дану конструкцію можна використовувати для деякого певного продукту, розміщуючи окремі записи з описом його переваг/властивостей або сукупностей характеристик. Усі записи пов’язані між собою та посилаються на головну, що дозволяє ефективно її просувати. Однак даний підхід вкрай специфічний і підходить далеко не для всіх Інтернет-майданчиків [22]. Приклад мережевої схеми веб-ресурсу представлений на рисунку 3.4.

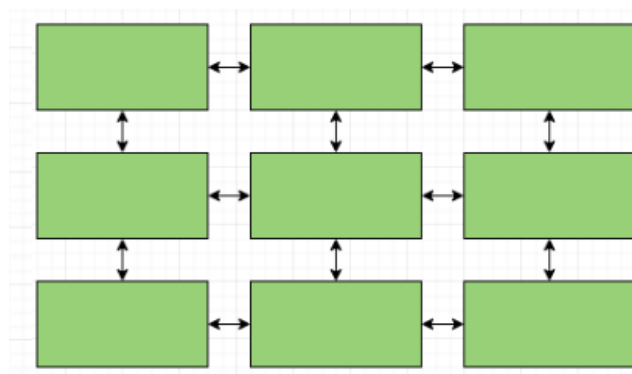


Рисунок 3.4 – Схема мережевої структури

Комбінована структура є поєднанням кількох попередніх структур. Наприклад, на сайті може бути передбачена ієрархічна структура, яка в певному місці може містити виконання покрокових дій (лінійну структуру).

Для онлайн-тренажеру «CSSStarting» була обрана мережева структура, схема якої представлена на рисунку 3.5.

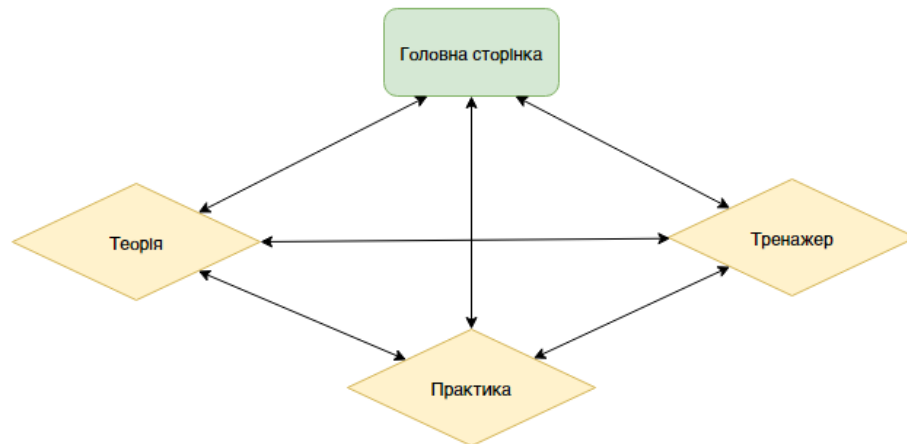


Рисунок 3.5 – Внутрішня структура тренажеру

3.2 Проектування та реалізація бази даних

Після того, як було визначено зовнішню та внутрішню структуру онлайн-тренажеру, необхідно було вирішити наступне питання – спосіб виведення інформації на тренажер. Для повноцінної та коректної його роботи необхідно спроектувати бази даних та виконати її підключення до онлайн-тренажеру. База даних тренажеру «CSStarting» складається із таких таблиць: `blockpossiblevalues`, `blockproperties`, `sectionsofclasses`, `textpossiblevalues`, `textproperties`, `theoryclasses`, `transformpossiblevalues`, `transformproperties`, `Users`. На рисунку 3.6 зображена структура бази даних тренажеру.

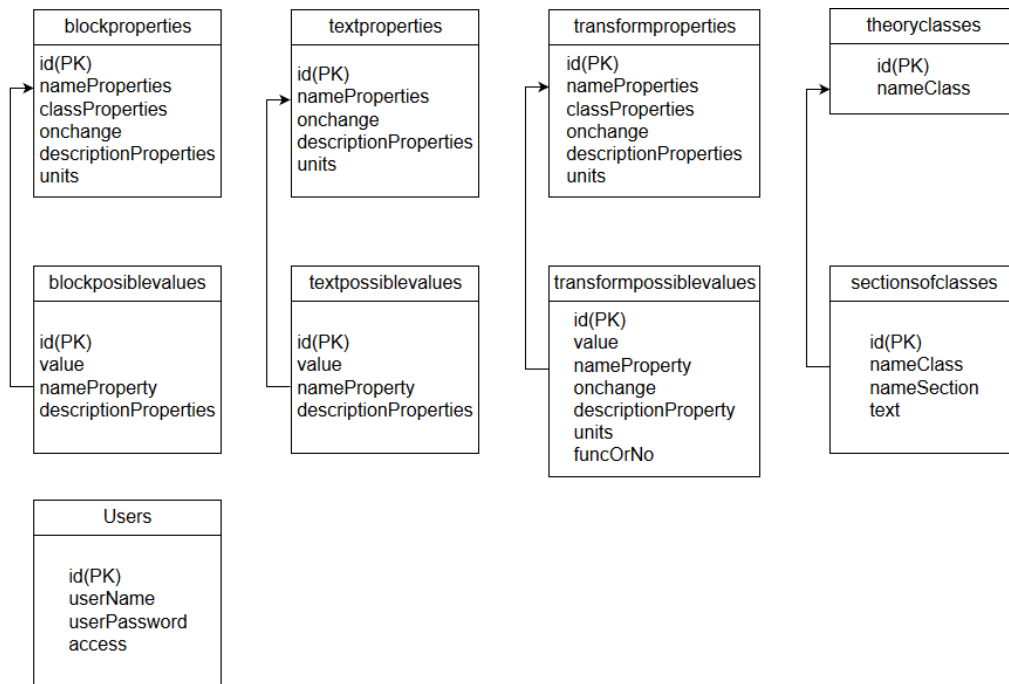


Рисунок 3.6 – Схема бази даних онлайн-тренажеру «CSStarting»

Після проектування бази даних була виконана її реалізація засобами MySQL. На рисунку 3.7 представлено таблиці бази даних «csstarting.sql» для онлайн-тренажеру «CSStarting».

	Таблиця	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/>	blockpossiblevalues	☆ [иконки]	57	InnoDB	utf8_general_ci	1.6 Киб	-
<input type="checkbox"/>	blockproperties	☆ [иконки]	25	InnoDB	utf8_general_ci	48 Киб	-
<input type="checkbox"/>	sectionsofclasses	☆ [иконки]	3	InnoDB	utf8_general_ci	32 Киб	-
<input type="checkbox"/>	textpossiblevalues	☆ [иконки]	42	InnoDB	utf8_general_ci	1.6 Киб	-
<input type="checkbox"/>	textproperties	☆ [иконки]	10	InnoDB	utf8_general_ci	1.6 Киб	-
<input type="checkbox"/>	theoryclasses	☆ [иконки]	1	InnoDB	utf8_general_ci	1.6 Киб	-
<input type="checkbox"/>	transformpossiblevalues	☆ [иконки]	10	InnoDB	utf8_general_ci	1.6 Киб	-
<input type="checkbox"/>	transformproperties	☆ [иконки]	6	InnoDB	utf8_general_ci	1.6 Киб	-
<input type="checkbox"/>	Users	☆ [иконки]	3	InnoDB	utf8_general_ci	1.6 Киб	-
	9 таблиц	Всего	157	InnoDB	utf8_general_ci	192 Киб	0 Байт

Рисунок 3.7 – База даних онлайн-тренажеру

На рисунку 3.8 представлена ER модель бази даних «csstarting.sql», створеної за допомогою Дизайнера phpMyAdmin.

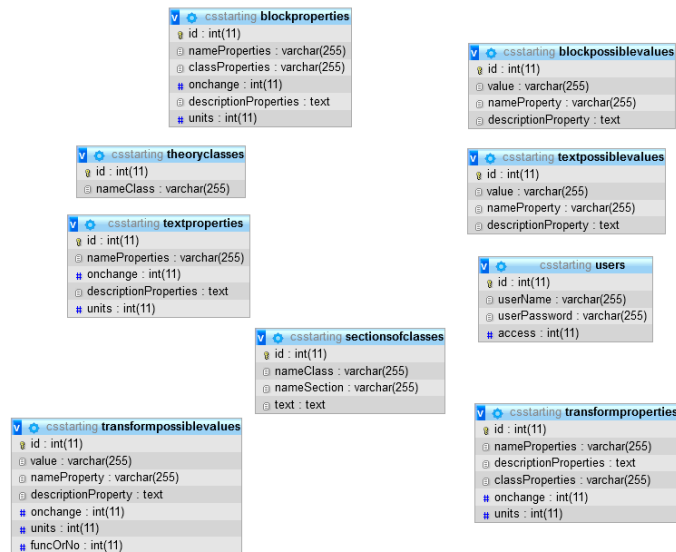


Рисунок 3.8 – ER модель бази даних

3.3 Розробка інтерфейсу

Як і будь-який веб-додаток на теренах мережі Інтернет-простору, онлайн-тренажер «CSStarting» повинен мати унікальний інтерфейс. Сучасні веб-сервіси та системи радикально змінюють роботу людей. Існують, наприклад, системи управління театральними механізмами, які спрощують роботу з електронним обладнанням, позбавляючи необхідності постійно перебувати у апаратури. Але робота не спроститься, якщо у таких систем буде занадто складний інтерфейс.

Виходить, що програмний інтерфейс не тільки вирішує проблему взаємодії з додатком, а й робить дану взаємодію максимально комфортною. Існує важлива наявність інтерфейсу, що дозволяє при меншій кількості зусиль ознайомитися з можливостями програми та зрозуміти принципи роботи [23].

Головна сторінка «CSStarting» має сучасний вигляд і оригінальне представлення навігації по сайту (рис. 3.9).

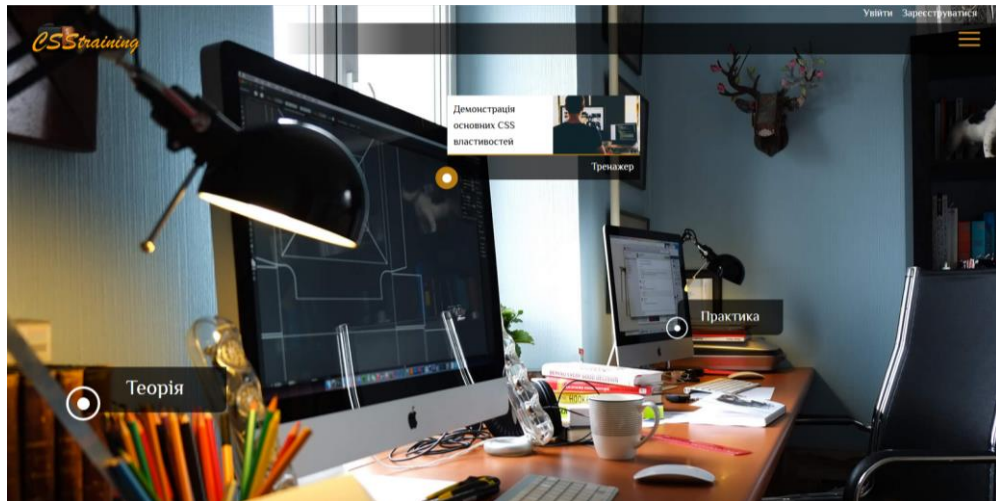


Рисунок 3.9 – Головна сторінка онлайн-тренажеру

За допомогою JavaScript та CSS реалізовано привабливий ефект руху фонового зображення в залежності від пересування миші. Використовуючи каскадні таблиці стилів були створені посилання на сторінки другого рівня. При наведенні на одне з таких посилань стає доступним відповідний блок із назвою розділу, коротким описом та зображенням, а при натисненні – користувач переходить на цей розділ.

У верхній частині головної сторінки розташована «шапка» сайту, яка дублюється на всіх інших сторінках. Вона створюється за допомогою шаблонів CSS та функцій PHP. «Шапка» тренажеру має стриманий дизайн та спокійні, актуальні на сьогоднішній день кольори.

Для швидкого переходу на головну сторінку достатньо натиснути на логотип сайту, а для переходу до основних розділів було створено меню у вигляді списку гіперпосилань на інші сторінки сайту. Кожне таке посилання називається пунктом меню. Це важливий елемент сайту, так як користувач постійно звертається до нього і тому вимоги до нього дуже високі.

Меню онлайн-тренажеру складається з наступних сторінок: «Теорія», «Тренажер», «Практика». Кожен пункт, при наведенні мишею, змінює колір та зміщується ліворуч (рис. 3.10).

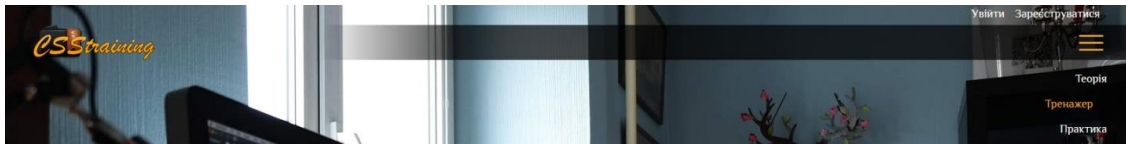


Рисунок 3.10 – Меню онлайн-тренажеру

Для досягнення даного ефекту були використані властивості каскадних таблиць стилів, що представлені в таблиці 3.1.

Таблиця 3.1 – Властивості CSS для меню

Властивість	Пояснення
hover	псевдоклас, який визначає стиль елемента при наведенні на нього курсора миші, але при цьому елемент ще не активований, іншими словами кнопка миші не було натиснуто.
transition	встановлює ефект переходу між двома станами елемента, вони можуть бути визначені з допомогою псевдокласу :hover або :active, також динамічно через JavaScript.
color	визначає колір тексту елемента.
padding	встановлює значення полів навколо вмісту елемента.

Для реєстрації та авторизації користувача, шапка сайту містить посилання на відповідні форми, які були реалізовані засобами HTML, CSS, JavaScript на стороні клієнта та PHP на стороні сервера (рис. 3.11-3.12).

На головній сторінці існують мітки основних блоків онлайн-тренажеру. При наведенні курсору миші, користувач може одразу побачити діалогове вікно з підказками, за що відповідає обраний блок. На рисунку 3.13 представлений приклад діалогового вікна.

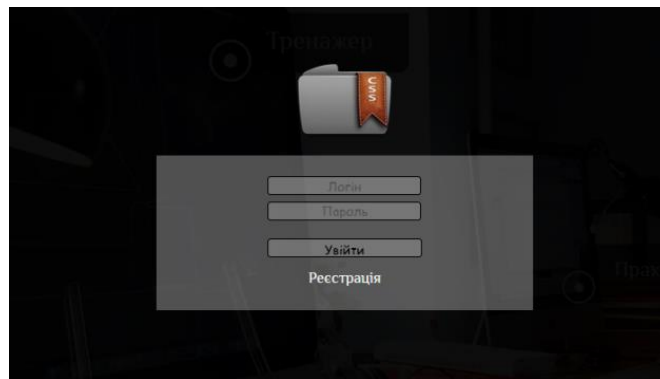


Рисунок 3.11 – Форма авторизації



Рисунок 3.12 – Форма реєстрації

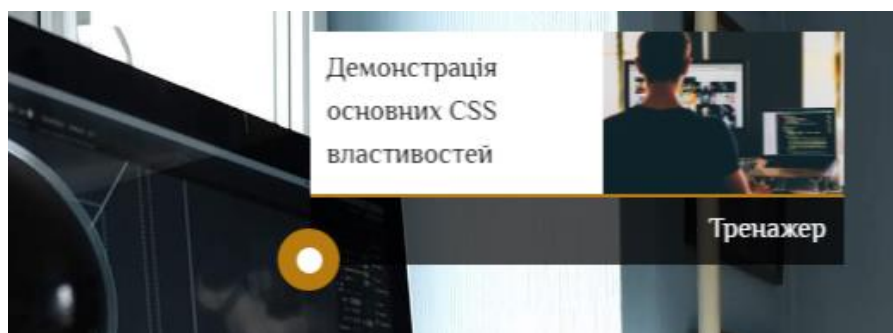


Рисунок 3.13 – Діалогове вікно підказки

Сторінки другого рівня онлайн-тренажеру мають єдиний шаблон оформлення у вигляді слайдера, який написаний на мові програмування JavaScript, а стилізований та анімований за допомогою CSS.

На сторінці, яка присвячена теорії, кожен слайд – це певний теоретичний розділ, а кожен пункт меню в сайдбарі – це окрема тема цього розділу. Завдяки

такому підходу відображення контенту користувач вільно зможе ознайомлюватися з теорією і не плутатися. На рисунку 3.14 представлений блок теоретичного розділу.



Рисунок 3.14 – Теоретичний розділ

У розділі «Тренажер» кожен слайд присвячений окремій, тематично об'єднаній групі CSS правил. У сайдбарі відображені ці правила та відповідні поля для зміни їх значення. На рисунку 3.15 представлений приклад блоку тренажеру.

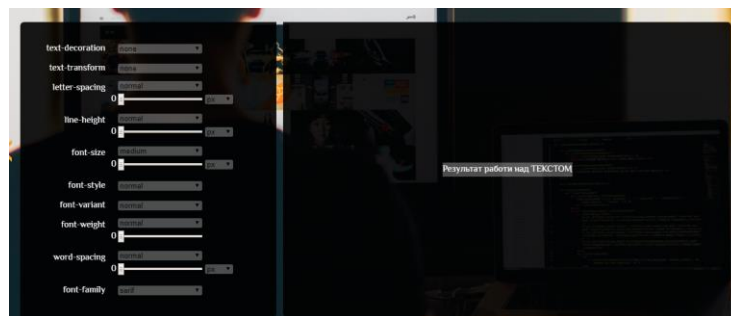


Рисунок 3.15 – Приклад блоку тренажеру

У ході заповнення даних полів користувач одразу може спостерігати результат застосування прописаних правил у блоці з прикладом (рис. 3.16).

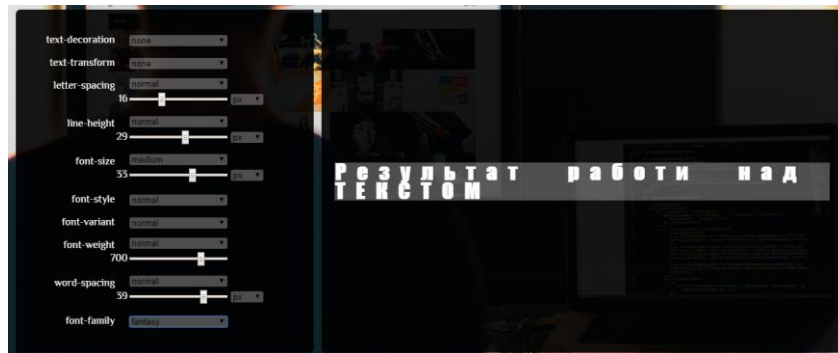


Рисунок 3.16 – Зміна тексту від дій користувача

Також на кожній сторінці розділу «Тренажер» реалізована система, яка дозволяє, не переходячи в іншу вкладку, прочитати коротку довідку про одне з CSS-правил. Для цього користувачу достатньо натиснути на назву потрібної властивості, після чого стає доступним вікно з коротким описом (рис. 3.17). Такий ефект створений за допомогою мови програмування JavaScript та технології асинхронних запитів на сервер – AJAX.

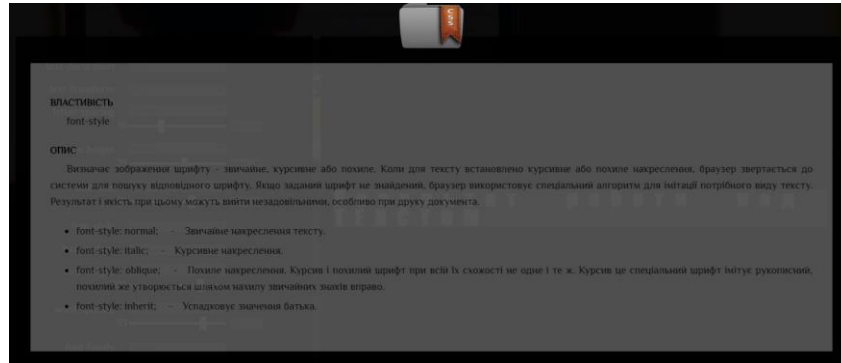


Рисунок 3.17 – Вікно з інформацією про обрану властивість

У розділі «Тренажер» існують завдання, які стосуються або лише батьківських блоків, або лише дочірніх. І для того, щоб користувач розумів які саме властивості застосовуються до батька елемента, а які, безпосередньо, до елемента їх було розмежовано таким чином, що в сайдбарі показана одна з груп правил. Для відображення тих чи інших властивостей користувач повинен обрати відповідний елемент в верхній частині сайдбару (рис. 3.18-3.19). Така організація була створена засобами JavaScript.

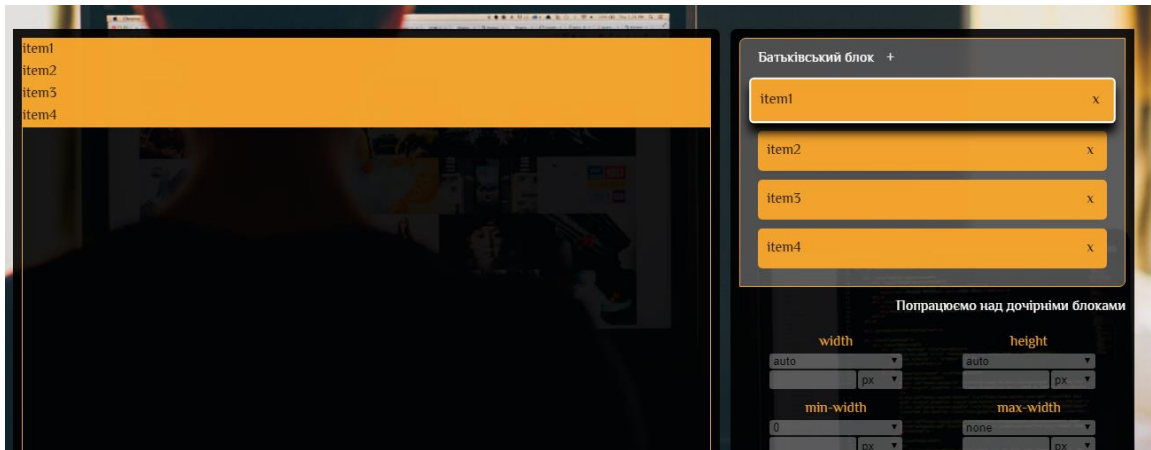


Рисунок 3.18 – Приклад роботи з батьківським елементом

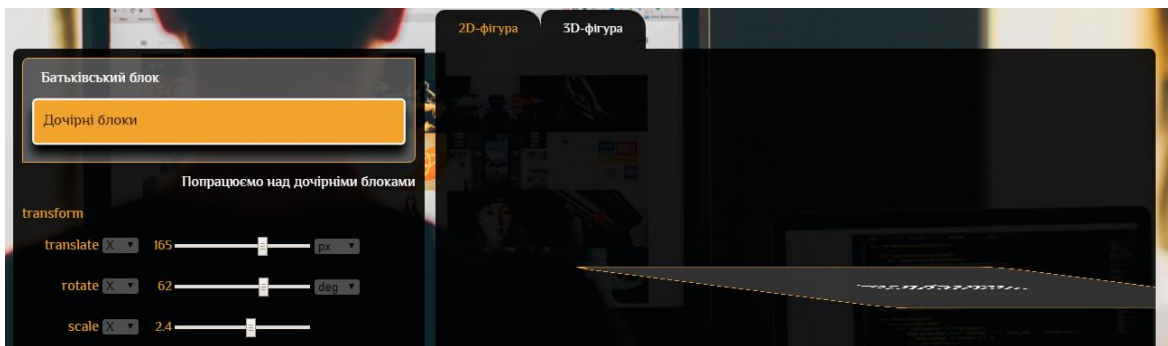


Рисунок 3.19 – Приклад роботи з дочірнім елементом

У розділі «Практика» кожен слайд – це тематично об'єднані приклади, а кожен пункт меню в сайдбарі – це окремий приклад. На рисунку 3.20 представлений блок розділу «Практика».

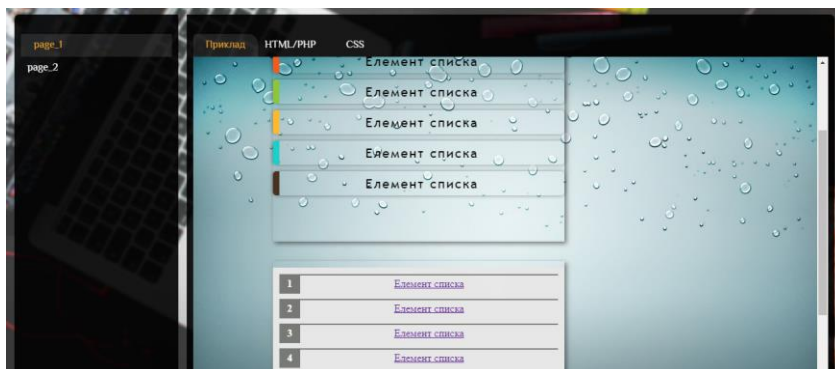


Рисунок 3.20 – Приклад блоку «Практика»

Засобами PHP та JavaScript реалізована можливість не лише перегляду прикладів, а й ознайомлення з їх вихідними HTML, PHP, CSS та JavaScript кодами. Для цього користувачу потрібно обрати відповідний пункт над прикладом. Завдяки тому, що користувач може одразу переглянути супровідний код та побачити результат в тому ж самому вікні, розділ «Практика» є максимально ефективним для засвоювання знань користувача.

Також в структурі онлайн-тренажу розроблена адміністративна панель, яка надасть адміністратору можливість додавати, видаляти, змінювати контент онлайн-тренажеру. На рисунку 3.21 представлена панель адміністратора.

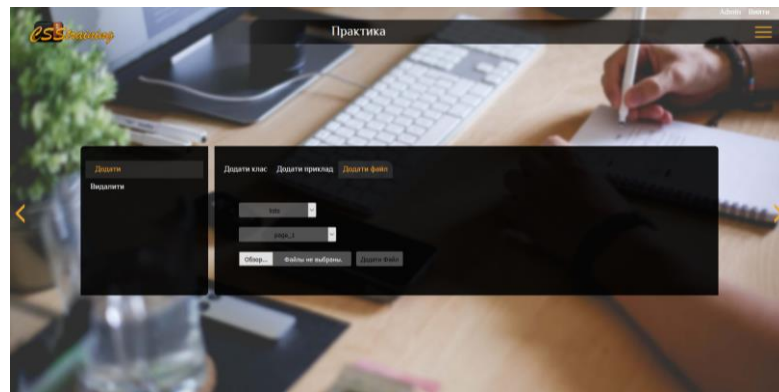


Рисунок 3.20 – Панель адміністратора

У Додатку В представлений супровідний код для реалізації всіх тих процес, які були описані раніше.

3.4 Розробка програмного коду

Інтерактивність формам надає JavaScript, а обробку вхідних даних виконує PHP, після чого повертає результат клієнту. Також на всіх сторінках додано preload – це певна анімація, яка відображається до того моменту, поки сторінка не буде повністю готова для роботи з клієнтом. Preload встановлюється для того, щоб користувач не спостерігав як елементи поступово завантажуються, чи як вони

мерехтять. Для його реалізації було використано засоби JavaScript, а саме: подія `window.onload()`, та каскадні таблиці стилів, а саме: `@keyframes` та `animation`. Фрагмент коду, відповідальний за `preload`, наведено нижче:

```
// js код
function preloader(){
  setTimeout(function(){
    var preloader = document.querySelector('.preloader');
    if( !preloader.classList.contains('done') ){
      preloader.classList.add('done');
    }
  }, 300);
}

window.onload(){
  preloader();
}

//css код
.load-2{
position: absolute;
width: 50px;
height: 50px;
border-radius: 50%;
border: 5px solid black;
border-top: 5px solid #F2A42E;
border-bottom: 5px solid rgba(199, 55, 55, 0.99);
top: 50%;
left: 50%;
transform: translate(-50%, -50%, 0);
animation: .9s spin infinite linear reverse;
}

@keyframes spin {
  from{
    transform: translate(-50%, -50%) rotate(0deg);
  }
  to{
    transform: translate(-50%, -50%) rotate(360deg);
  }
}
```

У теоретичному та практичному розділах для відображення потрібної лекції чи прикладу, відповідно, була використана технологія AJAX. Приведений нижче фрагмент коду відповідає за відправку запиту на сервер та отримання відповіді від нього:

```
function formForUploadFiles(param){
  var request = new XMLHttpRequest();

  request.onreadystatechange = function(){
    if(request.readyState == 4 && request.status == 200){
```



```

    var rez = request.responseText; // відповідь від серверу
  }
}
request.open('POST', '../php/formForUploadFiles.php');
request.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded');
request.send(param);
}

```

Для реалізації тренажерів було використано засоби JavaScript, а саме його методи роботи з стилями елементів DOM. Перелік цих методів наведено в таблиці 3.2.

Таблиця 3.2 – Методи для роботи з елементами DOM.

Метод	Опис методу
element.style	повертає об'єкт, який дає доступ до стилю елемента на читання і запис.
element.cssText	повертає об'єкт, який дає доступ до стилю елемента на читання і запис цілком у вигляді рядка.
element.classList	повертає список класів елемента.
element.classList.add('sda')	додає елементу клас 'sda' .
element.classList.remove('sda')	видаляє елементу клас 'sda' .
element.classList.contains('sda')	перевіряє чи є у елемента клас 'sda' .

Після того, як онлайн-тренажер був розроблений та протестований, розробником проекту було сформовано інструкцію користувача (Додаток Г) та інструкцію адміністратора (Додаток Д).

ВИСНОВКИ

Під час виконання дипломного проекту «Онлайн-тренажер з вивчення властивостей CSS» був розроблений інтерактивний тренажер у вигляді веб-ресурсу. Онлайн-тренажер був розроблений до вимог, що були зазначені у технічному завданні, а саме:

- наявність адаптивної верстки онлайн-тренажеру;
- можливість перегляду теоретичного матеріалу;
- можливість проходження практичних та інтерактивних завдань;
- наявність форми реєстрації/авторизації користувачів.

Даний проект буде нести соціально-економічну цінність, оскільки його можна використовувати в навчально-виховних закладах у курсах, пов'язаних з веб-розробкою. Онлайн-тренажер «CSStraining» можна використовувати як допоміжний ресурс для закріплення знань студентів як очного, так і заочного відділення.

У ході виконання дипломного проекту розробником було проведено детальний аналіз предметної області: визначено питання актуальності проблеми, проведено аналіз онлайн-тренажерів аналогів, сформовано мету та задачі для реалізації продукту, проведено аналіз існуючих технологій розробки. Технічне завдання до онлайн-тренажеру представлено у Додатку А.

Також було проведено проектування онлайн-тренажеру, що дозволило повноцінно та детально вивчити процес розробки онлайн-тренажерів з використанням діаграм нотації IDEF0. На даному етапі роботи було розроблено діаграму варіантів використання, що відображає головних акторів системи та їх сценарії дій з онлайн-тренажером. Розробником проекту було проведено планування робіт, що дозволило визначити ключові етапи робіт, їх часові рамки виконання, ризи та сценарії їх мінімізації.

Після того, як був реалізований онлайн-тренажер, було проведено його тестування. Результати даного етапу описані нижче:

- адаптивність верстки працює коректно;
- форма реєстрації/авторизації містить модулі верифікації даних та має властивість запам'ятовувати дані користувача;
- анімація блоків працює коректно.

СПИСОК ЛІТЕРАТУРИ

1. Тенденции развития веб-технологий в 2019 году | PowerCode Academy [Электронный ресурс] – режим доступа: <https://cutt.ly/LtzwzZM>
2. IT-специалист - где учиться, зарплата, преимущества профессии – “Навигатор Образования” [Электронный ресурс] – режим доступа: <https://cutt.ly/qtzwvdL>
3. Что нужно знать, чтобы стать IT-специалистом | СЕГОДНЯ [Электронный ресурс] – режим доступа: <https://cutt.ly/Ltzwbnf>
4. Как прийти в IT (с чего начинать) | Хроники детерминированности [Электронный ресурс] – режим доступа: <https://cutt.ly/rtzwQe7>
5. HTML Academy - о компании | Rusbase [Электронный ресурс] – режим доступа: <https://cutt.ly/otzwWDQ>
6. Онлайн генератор (тренажер) для лучшего понимания перспективы в CSS3 (transform: perspective) | NISNOM [Электронный ресурс] – режим доступа: <https://cutt.ly/TtzwETV>
7. Flexbox froggy 24; Информатика в школе [Электронный ресурс] – режим доступа: <https://cutt.ly/ZtzwTra>
8. Что такое интерактивный сайт? Разработка интерактивных сайтов [Электронный ресурс] – режим доступа: <https://cutt.ly/2tzwYZr>
9. What is HTML? | HyperText Markup Language explained [Электронный ресурс] – режим доступа: <https://cutt.ly/otzwIa3>
10. What is CSS, How Does It Work and What is It Used For? – Skillcrush [Электронный ресурс] – режим доступа: <https://cutt.ly/7tzwOMK>
11. Преимущества стилей | htmlbook.ru [Электронный ресурс] – режим доступа: <https://cutt.ly/3tzwAmn>
12. What is JavaScript? - Learn web development | MDN [Электронный ресурс] – режим доступа: <https://cutt.ly/MtzwSOV>

13. javascript.info [Электронный ресурс] – режим доступа: <https://cutt.ly/ytzwFx8>
14. PHP: What is PHP? – Manual [Электронный ресурс] – режим доступа: <https://cutt.ly/ptzwG5A>
15. What is PHP? Write your first PHP Program [Электронный ресурс] – режим доступа: <https://cutt.ly/VtzwJYy>
16. What is MySQL? - Definition from WhatIs.com [Электронный ресурс] – режим доступа: <https://cutt.ly/4tzwK5b>
17. What are the advantages and disadvantages of using MySQL stored procedures? [Электронный ресурс] – режим доступа: <https://cutt.ly/YtzwZg1>
18. What is AJAX, How Does It Work, and What Is It Used For? – Skillcrush [Электронный ресурс] – режим доступа: <https://cutt.ly/WtzwMeL>
19. What is AJAX? – Tutorialspoint [Электронный ресурс] – режим доступа: <https://cutt.ly/Otzw1sk>
20. Знакомство с нотацией IDEF0 и пример использования / Блог компании Trinion / Хабр [Электронный ресурс] – режим доступа: <https://cutt.ly/zyEVOCr>
21. UML — диаграмма вариантов использования (use case diagram) / Хабр [Электронный ресурс] – режим доступа: <https://cutt.ly/IyEVAo7>
22. Структура сайта: схемы, советы по разработке, примеры [Электронный ресурс] – режим доступа: <https://cutt.ly/qyEVSSa>
23. Проектирование пользовательских интерфейсов: что это такое и зачем его заказывать — Блог Live Typing [Электронный ресурс] – режим доступа: <https://cutt.ly/gyEVFW8>

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ на розробку онлайн-тренажеру «CSStarting»

Назва онлайн-тренажеру «CSSStarting».

Область застосування онлайн-тренажеру – онлайн-тренажер розроблений для поліпшення вивчення властивостей та атрибутів CSS слухачами курсів «Веб-дизайн та веб-розробка»

Об'єкт, у якому використовують програму – секція Інформаційних технологій проектування.

1 Основи для розробки

Розробка виконується на основі завдання, виданого викладачами секції Інформаційних технологій проектування.

2 Призначення розробки

Розробка повинна надати користувачу перевіряти знання, виконуючи інтерактивні тести та покрокові завдання.

Назва організації: Пономарьова Анастасія

Тема проекту: «Онлайн-тренажер з вивчення властивостей CSS»

3 Вимоги до веб-ресурсу

Веб-ресурс повинен бути реалізований у вигляді адаптивного сайту з наявністю форми входу/реєстрації та повинен забезпечувати виконання функціональних можливостей, визначених у пункті 3.2.

3.1 Вимоги до веб-ресурсу

Онлайн-тренажер розробляється ітеративно із врахуванням принципів та технологій уніфікованого процесу розроблення сайту. Каркас має бути реалізований мовою HTML з підключенням до CSS. Функції, перелічені в пункті 3.3, мають бути написані мовою JavaScript з використанням технології AJAX.

3.2 Вимоги до функціональних характеристик

Онлайн-тренажер повинен забезпечувати виконання наступних функцій:

- перегляд теоретичного матеріалу;
- проходження практичних завдань;
- проходження інтерактивних тестів;

4 Перелік програмної документації

- опис проекту продукту із використанням UML-діаграм;
- логічна схема роботи онлайн-тренажеру;
- програмний код розробки;
- технічне завдання;

5 Порядок виконання робіт і етапи розробки

Стадії та етапи розробки повинні складатися з наступних пунктів:

- оформлення завдання для дипломної роботи;
- планування роботи: розроблення ТЗ ,побудова мережевого графіку та діаграми Ганта;

- розроблення логічної схеми онлайн-тренажеру;
- реалізація веб-сторінки тренажеру;
- розроблення логіки роботи онлайн-тренажеру;
- провести тестування розробки;
- оформлення пояснювальної записки про виконання дипломної роботи;
- здача пояснювальної записки до дипломної роботи та розробленого онлайн-тренажеру;
- презентація роботи та її захист.

6 Порядок контролю та приймання

Контроль коректності функціонування та придатності веб-ресурсу здійснюється замовником (секцією Інформаційні технології проектування) на основі наданої пояснювальної записки до дипломної роботи та програмних файлів. Контроль ходу виконання проекту здійснюється на основі календарного плану виконання дипломної роботи:

1. Перевірка завдання дипломної роботи.
2. Перевірка тз, мережевого графіка та діаграми ганта.
3. Перевірка логічної схеми для онлайн-тренажеру.
4. Перевірка веб-сторінки.
5. Перевірка коректності логіки роботи онлайн-тренажеру.
6. Перевірка коректності тестування.
7. Здача пз.
8. Презентація.

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

Ідентифікація мети IT-проекту методом SMART.

Результатом дипломного проекту є онлайн-тренажер «CSStarting». Для визначення/ідентифікації мети був використаний відомий метод SMART. SMART це сучасний підхід до постановки працюють цілей. Система постановки smart-цілей дозволяє на етапі визначення мети узагальнити всю наявну інформацію, встановити прийнятні терміни роботи, визначити достатність ресурсів, надати всім учасникам процесу ясні, точні, конкретні завдання.

SMART є аббревіатурою, розшифровка якої: Specific, Measurable, Achievable, Relevant, Time bound. Кожна буква аббревіатури SMART означає критерій ефективності поставлених цілей. Розглянемо кожен критерій smart мети більш детально.

Результати ідентифікації мети проекту відображено в таблиці Б.1.

Таблиця Б.1 – Ідентифікація проекту

Specific	Створення онлайн-тренажеру «CSStarting» та розміщення його в інтернет-просторі.
Measurable	Збільшення кількості кваліфікованих фахівців в області front-development.
Achievable	Реалізація проекту виконується з використанням технології HTML, CSS, JS, PHP, MySQL, AJAX.
Relevant	Для реалізації проекту розробник має достатню кваліфікацію та навички роботи з технолгіями.
Time bound	Онлайн-тренажер має бути розроблено згідно календарного плану.

Опис фази розробки ІТ-проекту.

Ті, хто мають справу зі складними ІТ проектами керівники підтвердять, що поділ завдань на більш дрібні і керовані частини робить робочий процес набагато простіше. У цій статті розповім про процес, який допоможе структурувати кожен етап проекту і враховувати всі поставлені завдання. Йдеться про ієрархічну структуру робіт WBS (Work Breakdown Structure).

WBS заснована на графічній природі, яка допомагає менеджерам проектів передбачити результати, засновані на різних сценаріях. Процес часто описується як структура відгалуження, яка охоплює всі етапи проекту в організованому порядку. WBS також може бути представлена у вигляді табличного списку завдань і елементів в плані розбивки робіт діаграм Ганта. На рисунку. Б.1 представлена розроблена WBS-діаграма проекту «CSStarting».

Планування структури організації, для впровадження готового проекту.

Організаційна структура проекту (OBS) є ієрархічною структурою управління проектом і показує відносини між учасниками проекту.

Організаційна структура проекту (OBS):

- створюється на рівні підприємства;
- її елементи призначаються на рівні EPS, проекти, пакети робіт структури декомпозиції робіт (WBS);
- дозволяє контролювати доступ користувачів до інформації відповідного рівня.

Розроблена діаграма OBS проекту представлена на рисунку Б.2.

Побудова календарного графіку виконання ІТ – проекту.

Останнім етапом в плануванні проекту є розробка календарного плану у вигляді діаграма Ганта та формування PDM-мережі.

Діаграма Ганта – це візуальний спосіб відображення запланованих завдань. Горизонтальні графіки широко використовуються для планування проектів будь-яких розмірів в різних галузях і сферах. Це зручний спосіб показати, яка робота планується до виконання в певний день і час. Gantt Charts також допомагають командам і менеджерам проектів контролювати дати початку і закінчення будь-якого проекту .

На рисунку Б.3 представлена розроблена діаграма Ганта. Список робіт із зазначення дат початку та завершення представлений на рисунку Б.4. На рисунках Б.5-Б.8 представлена PDM-мережа

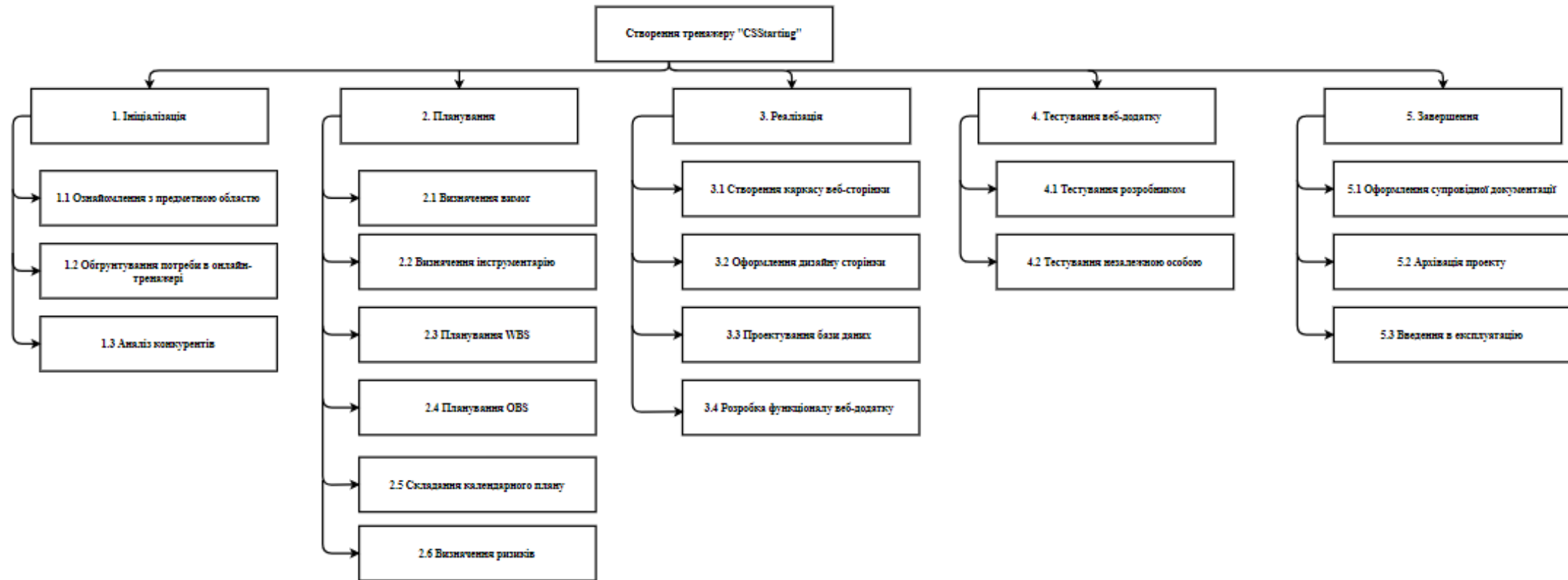


Рисунок Б.1 – WBS-діаграма дипломного проекту

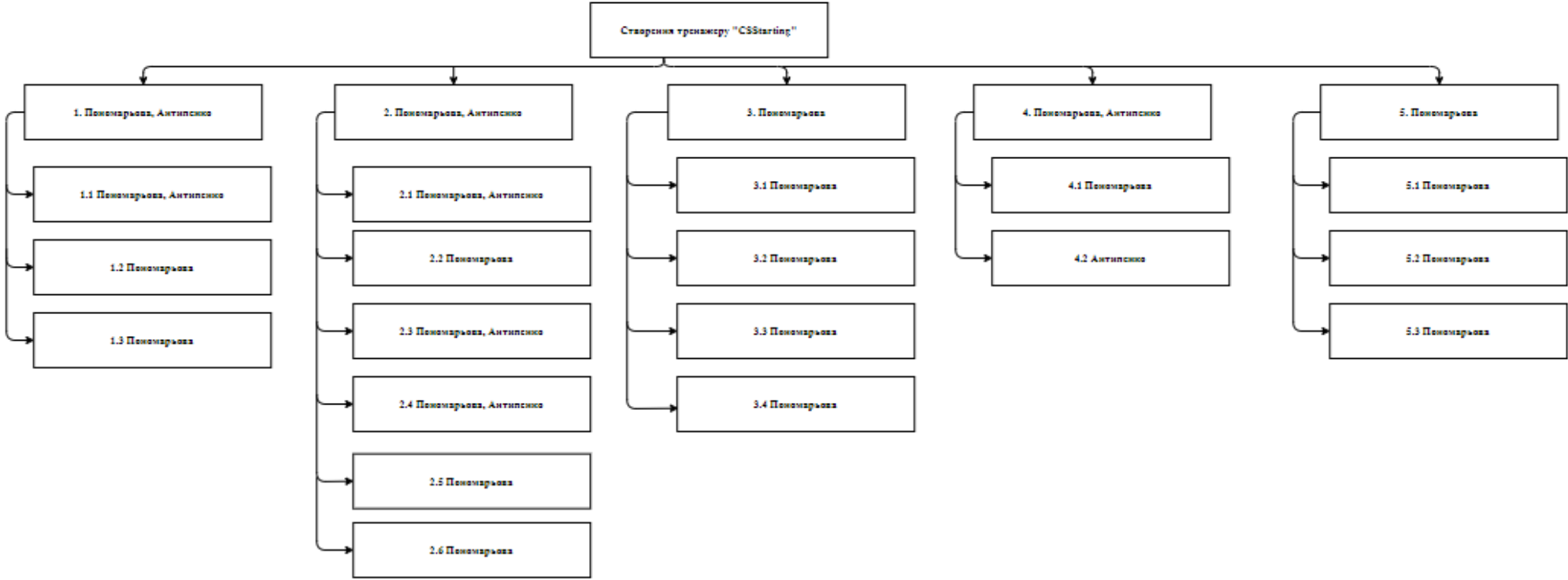


Рисунок Б.2 – WBS-діаграма дипломного проекту

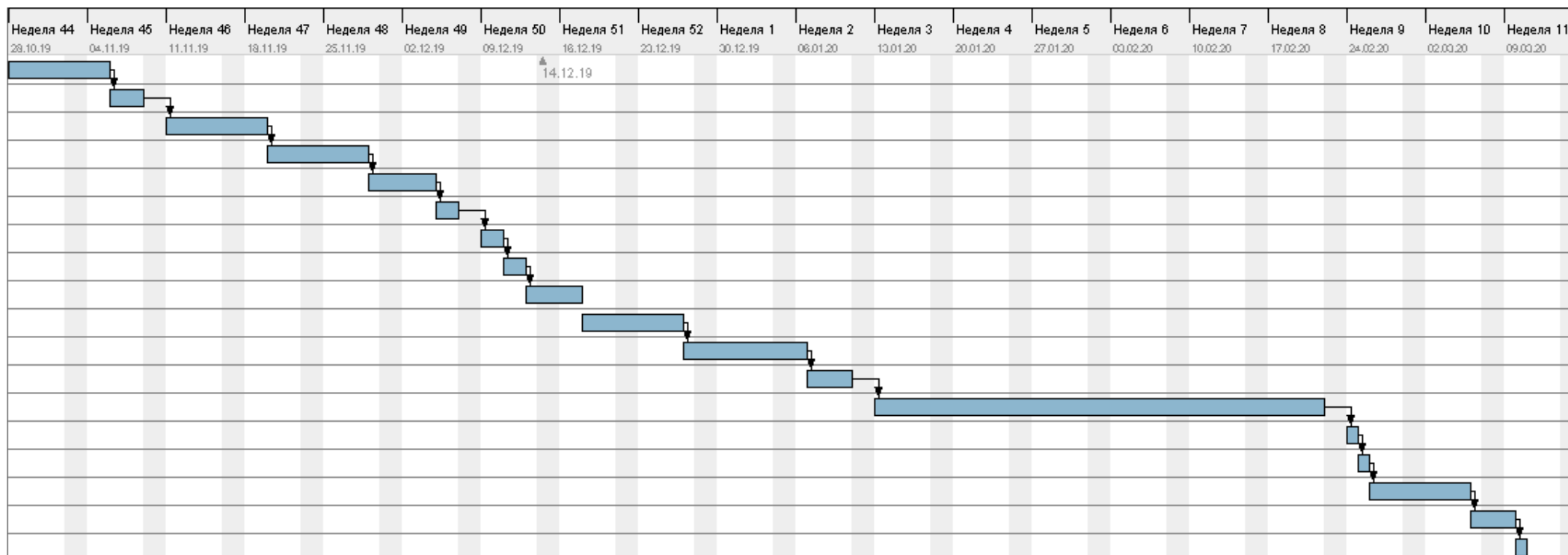


Рисунок Б.3 – Диаграмма Ганта для проекта «CSStarting»

Название	Дата начала	Дата окончания
• Ознайомлення з предметною областю	28.10.19	05.11.19
• Визначення в потребі онлайн-тренажеру	06.11.19	08.11.19
• Аналіз конкурентів	11.11.19	19.11.19
• Визначення вимог тренажеру	20.11.19	28.11.19
• Визначення інструментів	29.11.19	04.12.19
• Планування WBS	05.12.19	06.12.19
• Планування OBS	09.12.19	10.12.19
• Складання календарного плану	11.12.19	12.12.19
• Визначення ризиків	13.12.19	17.12.19
• Створення каркасу веб-сторінки	18.12.19	26.12.19
• Оформлення дизайну сторінки	27.12.19	06.01.20
• Проектування бази даних	07.01.20	10.01.20
• Розробка функціоналу тренажеру	13.01.20	21.02.20
• Тестування розробником	24.02.20	24.02.20
• Тестування незалежною особою	25.02.20	25.02.20
• Оформлення супровідної документації	26.02.20	05.03.20
• Архівація проекту	06.03.20	09.03.20
• Введення в експлуатацію	10.03.20	10.03.20

Рисунок Б.4 – Список робіт для діаграми Ганта

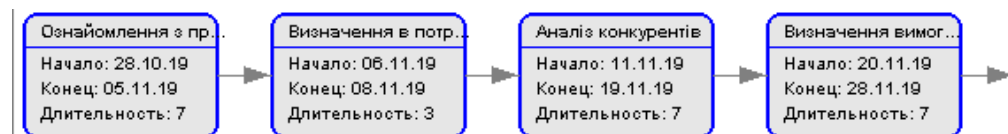


Рисунок Б.5 – PDM-мережа проекту

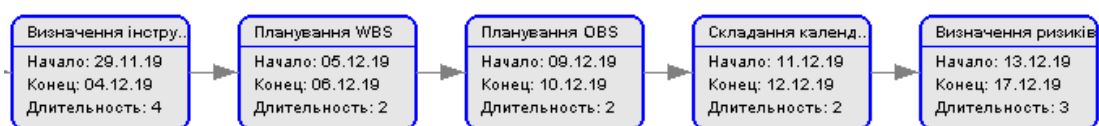


Рисунок Б.6 – Продовження PDM-мережі

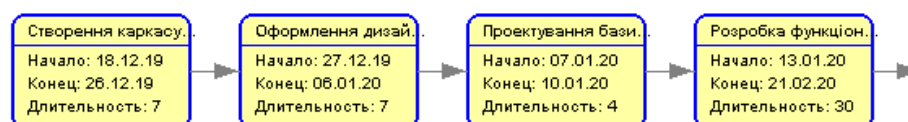


Рисунок Б.7 – Продовження PDM-мережі

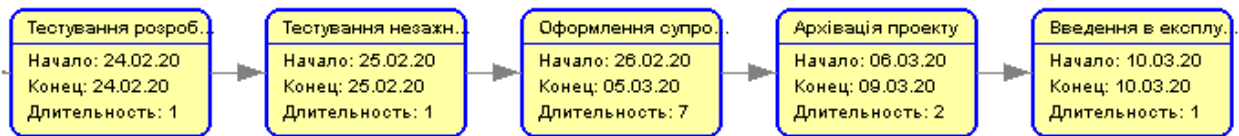


Рисунок Б.8 – Продовження PDM-мережі

Управління ризиками. Ризик – ймовірнісна подія, яка може позитивно чи негативно вплинути на проект. Ідентифікація ризиків – визначення ризиків, здатних вплинути на проект, і документування їх характеристик.

Ідентифікація ризиків визначає, які ризики здатні вплинути на проект, і документує характеристики цих ризиків. Ідентифікація ризиків не буде ефективною, якщо вона не буде проводитися регулярно протягом реалізації проекту.

Класифікація ризиків:

1. За ймовірністю виникнення:

- слабо ймовірнісні;
- мало ймовірнісні;
- ймовірні;
- досить ймовірні;
- майже ймовірні.

2. За величиною втрат:

- мінімальна;
- низька;
- середня;
- висока;
- максимальна.

На основі цих даних була проведена класифікація ризиків для даного проекту, що наведена в таблиці Б.2.

Таблиця Б.2 – Класифікація ризиків

№	Назва ризику	Ймовірність	Величина втрат
1	Некоректно складене ТЗ	2	4
2	Недотримання календарного плану	1	3
3	Некоректна робота програмного забезпечення	4	4
4	Некоректна робота апаратного забезпечення	4	4
5	Хвороба розробника	2	2
6	Некоректне тестування	2	1
7	Пошкодження файлу	4	5

Використовуючи дану класифікацію, була побудована матриця ризиків, що представлена на рисунку Б.9.

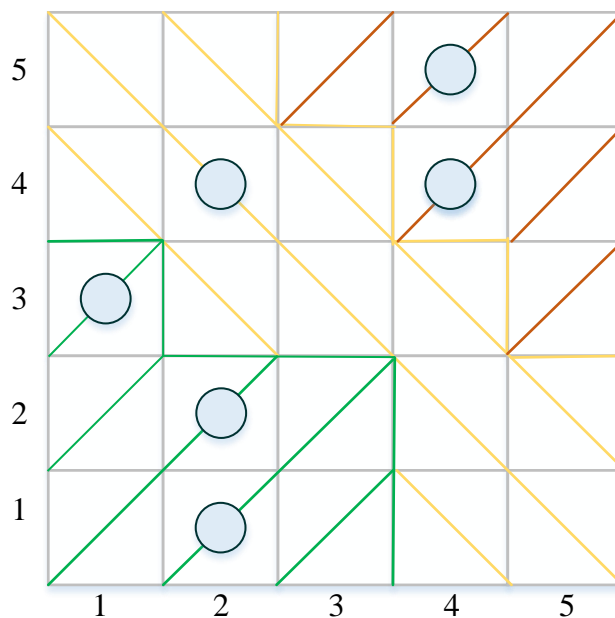


Рисунок Б.9 – Матриця ризиків

ДОДАТОК В

ФАЙЛИ ПРОГРАМНОГО КОДУ

Шаблон JavaScript

```
function preloader(){
    setTimeout(function(){
        var preloader = document.querySelector('.preloader');
        if( !preloader.classList.contains('done') ){
            preloader.classList.add('done');
        }
    }, 300);
}

function myMaxHeight(){
    var elements_saitbar =
document.querySelector('.content').getElementsByClassName('saitbar');

    var elements_result =
document.querySelector('.content').getElementsByClassName('rezult_of_changes')

    for(var i = 0; i < elements_result.length; i++){
        elements_result[i].style.maxHeight = elements_saitbar[i].offsetHeight + "px";

    }
}

function slaider(arr, leftOrRight){
    for(var i = 0; i < arr.length; i++){
        arr[i].classList.remove("show_slaid");
        arr[i].classList.remove("bounceInLeft");
        arr[i].classList.remove("bounceInRight");
    }
    if(leftOrRight == "right"){
        NumberOfSlaid++;

        if(NumberOfSlaid == arr.length){
            NumberOfSlaid = 0;
        }

        arr[NumberOfSlaid].classList.add("show_slaid");
    }
    arr[NumberOfSlaid].classList.add("animated");
}
```

```

    arr[NumberOfSlaid].classList.add("bounceInRight");
}
else{
    NumberOfSlaid--;

    if(NumberOfSlaid == -1){
        NumberOfSlaid = arr.length - 1;
    }

    arr[NumberOfSlaid].classList.add("show_slaid");
    arr[NumberOfSlaid].classList.add("animated");
    arr[NumberOfSlaid].classList.add("bounceInLeft");
}

document.querySelector('.header .header_down .wrap_for_text').innerHTML =
arr[NumberOfSlaid].getAttribute('name');
}
function ajaxAbout(param, nameProperties){
    var request = new XMLHttpRequest();
    request.onreadystatechange = function(){
        if(request.readyState == 4 && request.status == 200){
            var rez = request.responseText;
            if(rez == 1 || rez == 2 || rez == 3 || rez == 4)
                document.querySelector('.ModalAbout .about').innerHTML = "Помилка у
роботі з базою даних! Перевірте підключення і наявність файлу-обробника!";
            else{
                var arr = JSON.parse(rez);
                if(arr[0]['descriptionProperties'] == null)
                    var temp = arr[0]['descriptionProperty'];
                else
                    var temp = arr[0]['descriptionProperties'];
                var str = '<div class = "biggerText">Властивість</div>'
                    + '<p>' + nameProperties + '</p>'
                    + '<div class = "biggerText">Опис</div>'
                    + '<p>' + temp + '</p>'
                    + '<ul>';
            }
        }
    }
    request.open("GET", param, true);
    request.send();
}

```

```

    for(var i = 1; i < arr.length; i++){
        str += '<li>' + nameProperties + ': ' + arr[i]['value'] + '\t <span
class = "lineInAbout">-</span>' + arr[i]['descriptionProperty'] + '</li>';
    }

    str += '</ul>'

    document.querySelector('.ModalAbout .about').innerHTML = str;
}

document.querySelector('.ModalAbout').style.display = "flex";
}
}

request.open('POST', '../php/aboutProperties.php');

request.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded');

request.send(param);
}

```

Шаблон CSS

```

@import url("animation.css");

@font-face {
    font-family: 'ElMessiri';
    src: url('../fonts/ElMessiri-Regular.ttf') format('truetype');
    font-weight: normal;
    font-style: normal;
}

html, body{
    margin: 0;
    padding: 0;
    height: 100%;
    overflow-x: hidden;
}

```

```
}

body{
    font-family: 'ElMessiri', cursive, sans-serif;
}

*{
    margin: 0;
    box-sizing: border-box;
}

a{
    text-decoration: none;
    color: inherit;
}

::-webkit-scrollbar{
    width: 10px;
    height: 10px;
}

::-webkit-scrollbar-thumb{
    border-width: 1px 1px 1px 2px;
    border-color: #777;
    background-color: #aaa;
}

::-webkit-scrollbar-thumb:hover{
    border-width: 1px 1px 1px 2px;
    border-color: #555;
    background-color: #777;
}

::-webkit-scrollbar-track{
    border-width:0;
```

```
}  
::-webkit-scrollbar-track:hover{  
    border-left: solid 1px #aaa;  
    background-color: #eee;  
}  
  
select, input{  
    background: rgba(255, 255, 255, 0.3);  
    border: none;  
    border-radius: 3px;  
    text-align: center;  
}  
  
option{  
    background: rgba(255, 255, 255, 0.3);  
    border: none;  
}  
  
.my_header{  
    width: 100%;  
    height: 80px;  
}  
  
.header{  
    position: absolute;  
    width: 100%;  
    z-index: 10;  
    color: white;  
    top: -1px;  
}  
  
.header_top{
```

```
width: 100%;  
height: 30px;  
display: flex;  
justify-content: flex-end;  
align-items: center;  
padding: 0 2.5%;  
}  
  
.ir_btn{  
    cursor: pointer;  
    margin-left: 15px;  
    transition: color .3s; }  
  
.ir_btn:hover{  
    color: #F2A42E;  
}  
  
.header_down{  
    height: 50px;  
    display: flex;  
    justify-content: space-between;  
}  
  
.logo{  
    position: relative;  
    width: 30%;  
    height: inherit;  
    z-index: 5;  
    padding: 0 2.5%;  
}  
  
.logo img{  
    height: 100%;
```

```
}
```

```
.menu{  
    position: relative;  
    height: 100%;  
    width: 60%;  
    background: rgba(0, 0, 0, 0.7);  
    padding: 0 20px;  
    display: flex;  
    justify-content: space-between;  
}
```

```
.menu::before{  
    content: " ";  
    position: absolute;  
    box-sizing: border-box;  
    width: 20%;  
    height: 50px;  
    right: 100%;  
    background: linear-gradient(to right, rgba(0, 0, 0, 0.0), rgba(0, 0, 0,  
0.7));  
}
```

```
.main_menu{  
    position: relative;  
    width: 50px;  
    display: flex;  
    align-items: center;  
    cursor: pointer; }  
.main_menu img{  
    height: 70%;  
}
```



```
.main_menu:hover .in_main_menu{
    display: flex;
    animation: menu .5s;
}

.in_main_menu{
    padding: 10px;
    position: absolute;
    width: 200px;
    top: 100%;
    right: 0;
    display: none;
    flex-direction: column;
    align-items: flex-end;
}

.in_main_menu a{
    padding: 5px 0;
    transition: padding .3s, color .3s;
}

.in_main_menu a:hover{
    padding-right: 20px;
    color: #F2A42E;
}

@keyframes menu {
    from { right: 150%; }
    90% { right: -10px; }
    to { right: 0; }
}
```

```
.wrap_for_text{
    width: 100%;
    display: flex;
    justify-content: flex-start;
    align-items: center;
    font-size: 20pt;
    overflow: hidden;
}

.slaid_botton{
    position: fixed;
    width: 30px;
    height: 50px;
    cursor: pointer;
    z-index: 6;
    transition: transform .2s;
}

#left{
    top: 50%;
    left: 0;
    background: url('../images/svg/back.svg');
    background-size: cover;
    background-repeat: no-repeat;
}

#right{
    top: 50%;
    right: 0;
    background: url('../images/svg/back.svg');
    background-size: cover;
    background-repeat: no-repeat;
}
```

```
        transform: rotateY(180deg);
    }

#left:hover{
    transform: scale(1.5, 1.5);
}

#right:hover{
    transform: scale(1.5, 1.5) rotateY(180deg);
}

.parent_box{
    position: relative;
    display: none;
    max-width: 1400px;
    width: 100%;
    padding: 50px 45px;
}

.container{
    position: relative;
    display: flex;
    justify-content: space-between;
    z-index: 5;
}

.saitbar{
    position: relative;
    display: flex;
    flex-direction: column;
    justify-content: center;
    width: 36%;
```

```
padding: 30px 10px;
background: rgba(0, 0, 0, 0.9);
border-radius: 7px 0;
color: white;
}

.result_of_changes{
width: 63%;
background: rgba(0, 0, 0, 0.9);
padding: 10px;
overflow: auto;
position: relative;
border-radius: 0 7px;
color: white;
}

/* slaider */

.show_slaid{
display: block;
}

/*@media (max-width: 1270px){
.content{
margin: 0 10px;
}
}*/

@media (max-width: 1000px){
.wrap_for_text{
font-size: 15pt;
}
}
```

```
}
```

```
@media (max-width: 800px){  
  
  .wrap_for_text{  
    font-size: 12pt;  
  }  
  
  .rezult_of_changes{  
    order: 2;  
  }  
  
  .saitbar{  
    order: 1;  
  }  
  
  .container{  
    flex-direction: column;  
  }  
  
  .saitbar{  
    width: 100%;  
    margin-bottom: 15px;  
    flex-direction: row;  
    flex-wrap: wrap;  
    justify-content: space-around;  
  }  
  
  .option{  
    flex-basis: 250px;  
  }  
}
```

```

    .rezult_of_changes{
        width: 100%;
    }
}

```

Реалізація слайдера

// js код

```

function slaidер(arr, leftOrRight){

    for(var i = 0; i < arr.length; i++){

        arr[i].classList.remove("show_slaid");
        arr[i].classList.remove("bounceInLeft");
        arr[i].classList.remove("bounceInRight");
    }

    if(leftOrRight == "right"){

        NumberOfSlaid++;

        if(NumberOfSlaid == arr.length){

            NumberOfSlaid = 0;

        }

        arr[NumberOfSlaid].classList.add("show_slaid");
        arr[NumberOfSlaid].classList.add("animated");
        arr[NumberOfSlaid].classList.add("bounceInRight");
    }

    else{

        NumberOfSlaid--;

        if(NumberOfSlaid == -1){

```

```
        NumberOfSlaid = arr.length - 1;
    }

    arr[NumberOfSlaid].classList.add("show_slaid");
    arr[NumberOfSlaid].classList.add("animated");
    arr[NumberOfSlaid].classList.add("bounceInLeft");
}

    document.querySelector('.header .header_down .wrap_for_text').innerHTML =
arr[NumberOfSlaid].getAttribute('name');
}

window.onload = function(){
    NumberOfSlaid = 0;

    var allParentBox =
document.querySelector('.content').getElementsByClassName("parent_box");
    var left_button = document.querySelector('.content #left');
    var right_button = document.querySelector('.content #right');

    if(allParentBox.length < 2){
        document.querySelector('.content #left').style.display = "none";
        document.querySelector('.content #right').style.display = "none";
    }

    allParentBox[NumberOfSlaid].classList.add("show_slaid");
    allParentBox[NumberOfSlaid].classList.add("animated");
    allParentBox[NumberOfSlaid].classList.add("bounceInRight");

    document.querySelector('.header .header_down .wrap_for_text').innerHTML =
allParentBox[NumberOfSlaid].getAttribute('name');
```

```
left_button.onclick = function(){
    slaider(allParentBox, "left");
}

right_button.onclick = function(){
    slaider(allParentBox, "right");
}

var elements =
document.querySelector('.content').getElementsByClassName('saitbar');
}

// css код
.slaid_botton{
    position: fixed;
    width: 30px;
    height: 50px;
    cursor: pointer;
    z-index: 6;
    transition: transform .2s;
}

#left{
    top: 50%;
    left: 0;
    background: url('../images/svg/back.svg');
    background-size: cover;
    background-repeat: no-repeat;
}

#right{
```



```

    top: 50%;
    right: 0;
    background: url('../images/svg/back.svg');
    background-size: cover;
    background-repeat: no-repeat;
    transform: rotateY(180deg);
}

#left:hover{
    transform: scale(1.5, 1.5);
}

#right:hover{
    transform: scale(1.5, 1.5) rotateY(180deg);
}

.show_slaid{
    display: block;
}

```

Реалізація системи підказок

```

// js код

function myModalAbout(nameProperties, nameTableProperties,
nameTablePossibleValue){

    var param = "nameProperties=" + nameProperties
        + "&nameTableProperties=" + nameTableProperties
        + "&nameTablePossibleValue=" + nameTablePossibleValue;

    ajaxAbout(param, nameProperties);
}

function ajaxAbout(param, nameProperties){

    var request = new XMLHttpRequest();

```

```

request.onreadystatechange = function(){
    if(request.readyState == 4 && request.status == 200){
        var rez = request.responseText;
        if(rez == 1 || rez == 2 || rez == 3 || rez == 4)
            document.querySelector('.ModalAbout .about').innerHTML = "Помилка у
роботі з базою даних! Перевірте підключення і наявність файлу-обробника!";
        else{
            var arr = JSON.parse(rez);
            if(arr[0]['descriptionProperties'] == null)
                var temp = arr[0]['descriptionProperty'];
            else
                var temp = arr[0]['descriptionProperties'];
            var str = '<div class = "biggerText">Властивість</div>'
                + '<p>' + nameProperties + '</p>'
                + '<div class = "biggerText">Опис</div>'
                + '<p>' + temp + '</p>'
                + '<ul>';
            for(var i = 1; i < arr.length; i++){
                str += '<li>' + nameProperties + ': ' + arr[i]['value'] + ';\t <span
class = "lineInAbout">-</span>' + arr[i]['descriptionProperty'] + '</li>';
            }
            str += '</ul>'
            document.querySelector('.ModalAbout .about').innerHTML = str;
        }
        document.querySelector('.ModalAbout').style.display = "flex";
    }
}

request.open('POST', '../php/aboutProperties.php');
request.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded');
request.send(param);
}

```

```

//PHP код

<?php

require('db.php');

$nameProperties = $_POST['nameProperties'];

$nameTableProperties = $_POST['nameTableProperties'];

$nameTablePossibleValue = $_POST['nameTablePossibleValue'];

if($nameTableProperties == $nameTablePossibleValue){

    $descriptionProperties = mysqli_query($link, "select descriptionProperty
from $nameTableProperties where value = '\" . $nameProperties . '\"");

    $OutDescriptionProperties = mysqli_fetch_all($descriptionProperties, 1);

    if(!$descriptionProperties)

        echo 1; //ошибка выборки 1

    else if(!$OutDescriptionProperties)

        echo 10; //ошибка превращения массива 1

    else{

        echo json_encode($OutDescriptionProperties);

    }

}

else{

    $descriptionProperties = mysqli_query($link, "select descriptionProperties
from $nameTableProperties where nameProperties = '$nameProperties'");

    $OutDescriptionProperties = mysqli_fetch_all($descriptionProperties, 1);

    $descriptionPossibleProperty = mysqli_query($link, "select value,
descriptionProperty from $nameTablePossibleValue where nameProperty =
'$nameProperties'");

    $OutDescriptionPossibleProperty =
mysqli_fetch_all($descriptionPossibleProperty, 1);

    if(count($OutDescriptionPossibleProperty) == 0){

```

```
if(!$descriptionProperties)
    echo 1; //ошибка выборки 1
else if(!$OutDescriptionProperties)
    echo 2; //ошибка превращения массива 1
else{
    echo json_encode($OutDescriptionProperties);
}
}
else{
    if(!$descriptionPossibleProperty)
        echo 3; //ошибка выборки 2
    else if(!$OutDescriptionPossibleProperty)
        echo 4; //ошибка превращения массива 1
    else{
        array_unshift($OutDescriptionPossibleProperty,
$OutDescriptionProperties[0]);

        echo json_encode($OutDescriptionPossibleProperty);
    }
}
}
```

ДОДАТОК Г

ІНСТРУКЦІЯ КОРИСТУВАЧА

Для доступу на веб-сайт «CSStraining» необхідно запустити браузер і ввести в адресний рядок наступну адресу: <http://csstraining.zzz.com.ua>.

Після завершення завантаження, зазвичай це займає близько 3-4 секунд, відкриється головна сторінка. Для повнофункціонального перегляду сайту потрібно використовувати наступні браузери:

- MS Edge;
- Google Chrome 3.0 і вище;
- Opera 6.0 і вище;
- Mozilla Firefox 1.0;
- Mozilla 1.7.

У верхній частині головної сторінки розташовується «шапка», яка є незмінною на всіх сторінках сайту і складається з логотипу і випадаючого меню та посилань на форму реєстрації/авторизації.

Для реєстрації користувач повинен натиснути на посилання «Зареєструватись» та заповнити відповідну форму (рис. Г.1).

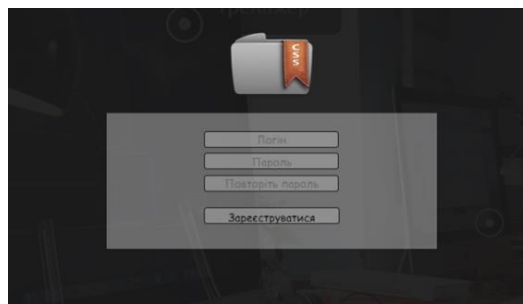


Рисунок Г.1 – Форма реєстрації

Після успішної реєстрації стає доступним повідомлення «Ви успішно зареєстровані» і сторінка оновиться. Тепер клієнт може увійти під щойно зареєстрованим ім'ям. Для цього йому потрібно натиснути на посилання «Увійти» та заповнити форму, яка з'явиться (рис. Г.2).

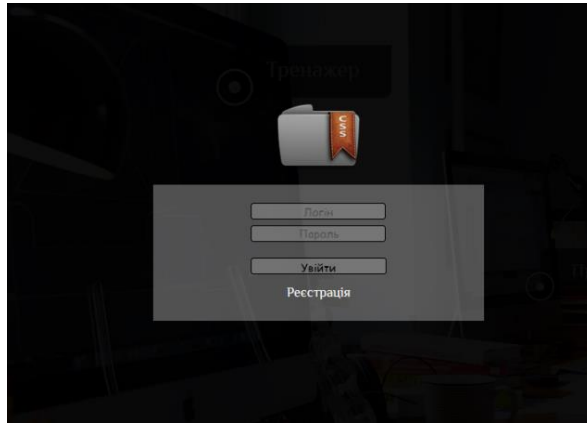


Рисунок Г.2 – Форма авторизації

Після успішної авторизації стає доступним повідомлення «Ви успішно авторизовані» і сторінка оновиться. У авторизованих користувачів замість посилань на вхід та реєстрацію відображаються його ім'я та кнопка виходу з облікового запису. У випадку, коли користувач має права доступу адміністратора, додатково відображається посилання на адміністративну панель.

Для переходу на один з розділів («Теорія», «Тренажер», «Практика») потрібно скористатися випадаючим меню або продубльованим посиланням, які тематично розташовані відносно фонового зображення. При наведенні на ці посилання з'являється блок з назвою, коротким описом та зображенням (рис. Г.3).

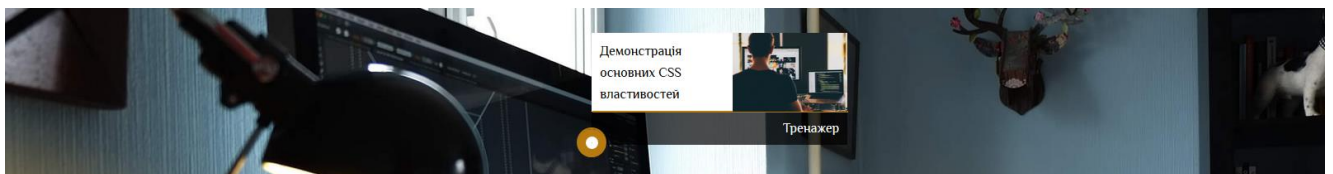


Рисунок Г.3 – Розгорнуте посилання

Теоретичний розділ надає можливість прочитати коротку інформацію про основні CSS властивості. Для переходу між розділами, користувачу потрібно скористатись кнопками переключення слайдів, при чому назва активного розділу буде відображена зверху (рис. Г.4). А потрібну лекцію розділу користувач обирає в сайдбарі (рис. Г.4).



Рисунок Г.4 – Теоретичний розділ

У розділі «Тренажер» клієнт має змогу переглянути результат роботи тих чи інших CSS правил. Для цього йому потрібно обрати потрібний тренажер, використовуючи кнопки перемикання слайдів, та у відповідні поля форми вводити значення (рис. Г.5).

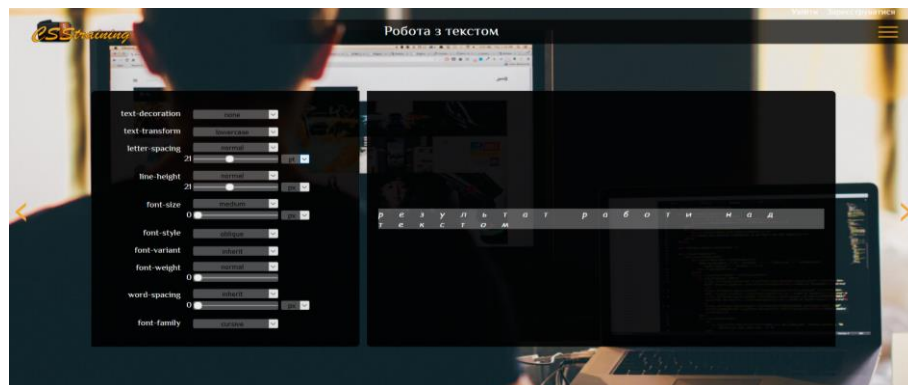


Рисунок Г.5 – Практичний розділ

Так як деякі CSS властивості повинні застосовуватись лише для батьківських елементів, а деякі безпосередньо до елементу, то їх було розділено. Для відображення потрібної групи правил потрібно натиснути на відповідний елемент у верхньому блоці сайдбару (рис. Г.6).

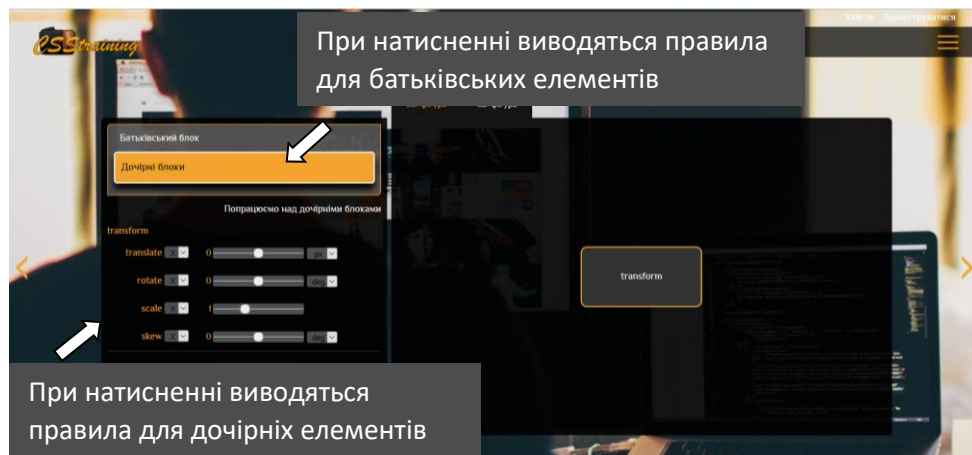


Рисунок Г.6 – Приклад розділення CSS правил

Для зручності, на сайті була розроблена система, яка дозволяє не переходячи на інші сторінки посібника дізнатися про будь-яку властивість. Для цього достатньо натиснути на назву потрібного правила, після чого стає доступним вікно з його визначенням, коротким описом та синтаксисом (рис. Г.7-Г.8).

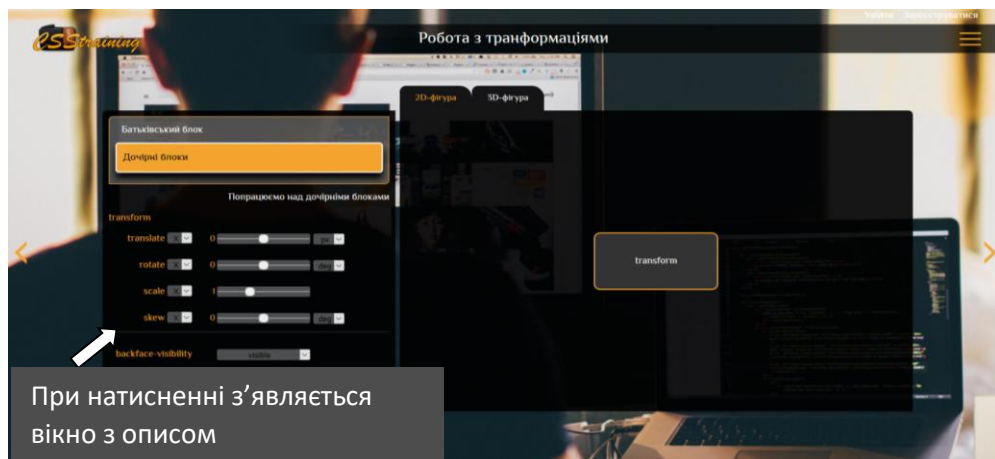


Рисунок Г.7 – При натисненні з'являється вікно з описом

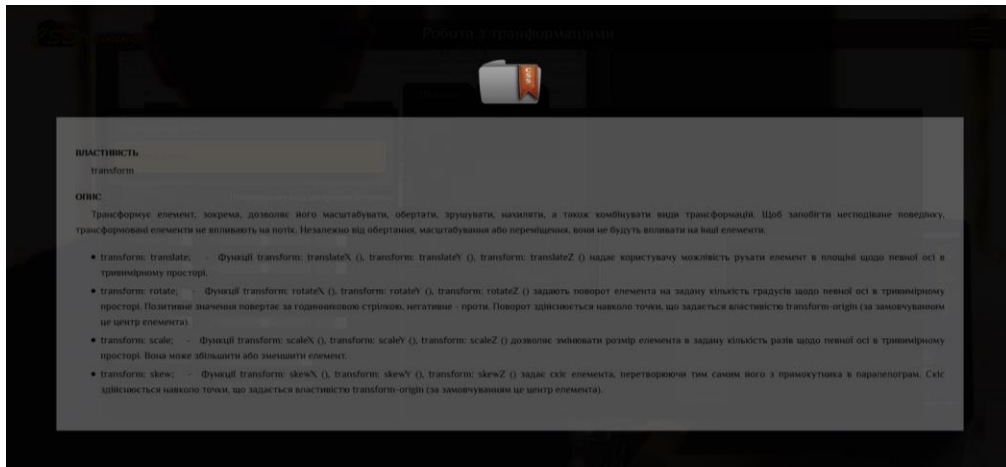


Рисунок Г.8 – Вікно опису CSS правила

У практичному розділі користувач може переглядати приклади комбінування CSS правил. Усі приклади розділені по тематичними групами. Для переходу між даними групами клієнт повинен переключати слайди, а для зміни прикладу – обрати бажаний у сайдбарі (рис. Г.9). Також користувач має змогу переглянути HTML, PHP, CSS та JavaScript (якщо такий наявний) код обраного прикладу, як показано на рисунку Г.9.

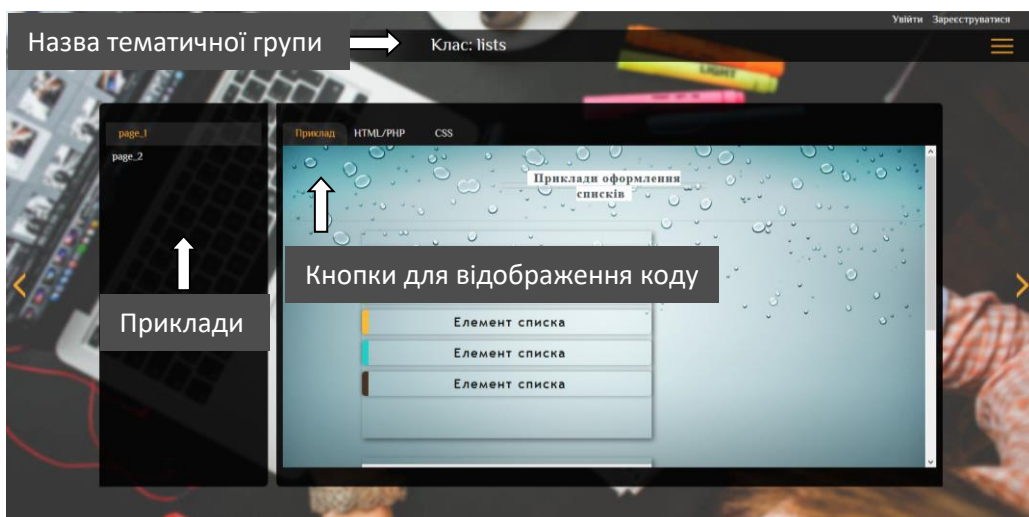


Рисунок Г.9 – Практичний розділ

ДОДАТОК Д

ІНСТРУКЦІЯ АДМІНІСТРАТОРА

Після того, як користувач авторизувався з правами адміністратора, в «шапці» сайту, поруч з ім'ям клієнта, стає доступним посилання на адміністративну панель. Також є спосіб потрапити на неї, не увійшовши в обліковий запис. Для цього в адресному рядку потрібно ввести в адресний рядок наступну адресу: <http://csstraining.zzz.com.ua/admin>. В обох випадках адміністратор потрапляє на сторінку, де йому потрібно ще раз підтвердити свій логін та пароль (рис. Д.1).



Рисунок Д.1 – Вхід до адміністративної панелі

Після успішного підтвердження стає доступним повідомлення «Ви успішно авторизовані» і сторінка оновиться. Адміністративна панель надає можливість редагування теоретичного та практичного розділу. Для того, щоб обрати розділ для редагування, адміністратор повинен скористатися кнопками переключення слайдів, при чому назва активного розділу буде виведена в «шапці» сайту (рис. Д.2). У сайдбарі відображаються доступні дії над розділом (рис. Д.2).

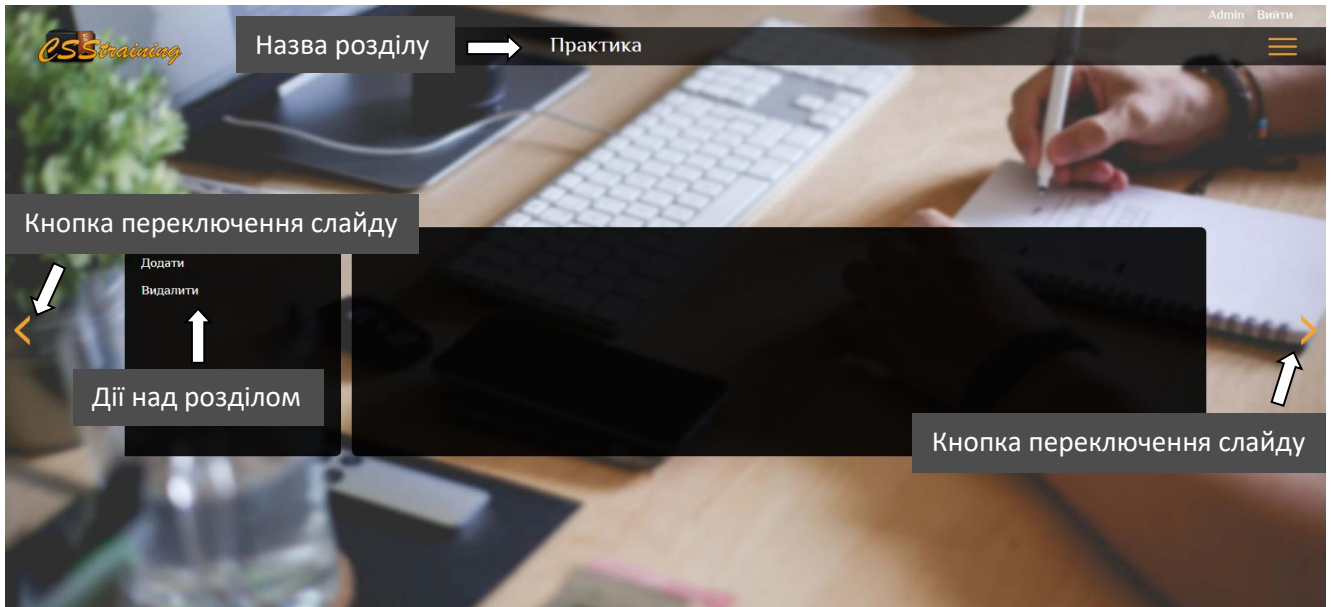


Рисунок Д.2 – Адміністративна панель

Так як теорія й практика розділена на тематичні групи (класи), в яких знаходяться окремі приклади чи лекції, то адміністратор має змогу додавати та видаляти класи, приклади та лекції. Для цього йому потрібно обрати дію в сайдбарі, а потім обрати відповідний пункт у списку, що стане доступним (рис. Д.3). Далі покроково виконати дії, які пропонує програма (рис. Д.3).

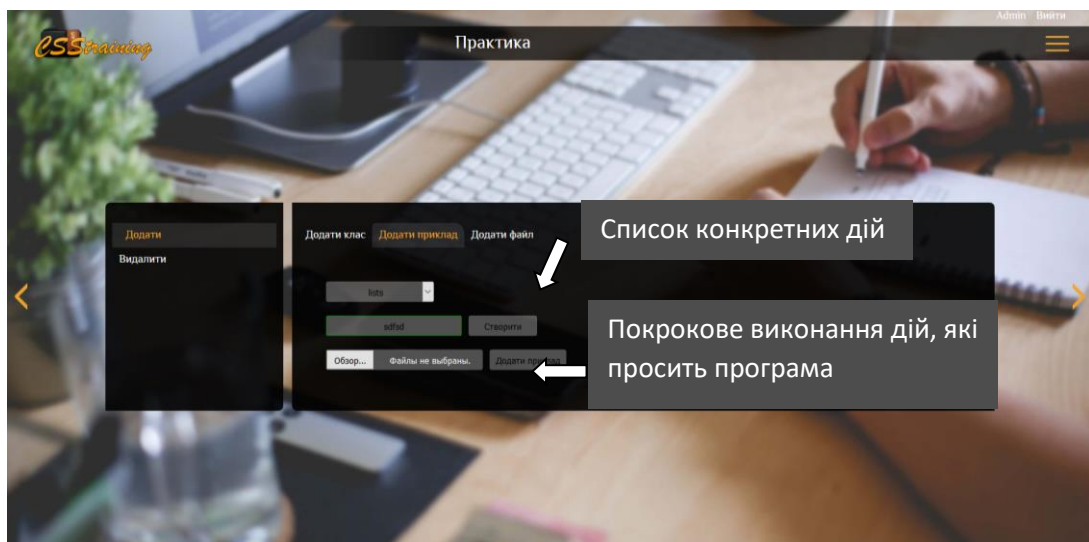


Рисунок Д.3 – Дії над розділами