

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
Фізико-математичний факультет
Кафедра інформатики та прикладної математики

«Допущено до захисту»

В. о завідувача кафедри

_____ Семеріков С. О.

« ____ » _____ 2025 р.

Реєстраційний № _____

« ____ » _____ 20 ____ р.

**РОЗРОБКА ІНТЕРАКТИВНОГО ОНЛАЙН-ТРЕНАЖЕРА
ДЛЯ ВИВЧЕННЯ ВЛАСТИВОСТЕЙ CSS**

Кваліфікаційна робота студента
групи І-21

ступінь вищої освіти «бакалавр»
спеціальності 014 Середня освіта
(Інформатика)

Литвиненка Віталія Андрійовича

Керівник:

кандидат фізико-математичних наук,
доцент кафедри інформатики та
прикладної математики
Тарасова Олена Юріївна

Оцінка:

Національна шкала _____

Шкала ECTS ____ Кількість балів _____

Голова комісії

(підпис)

(прізвище та ініціали)

Члени комісії

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

ЗАПЕВНЕННЯ

Я, Литвиненко Віталій Андрійович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело. Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.



ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ТЕОРЕТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....	5
1.1 Класифікація інтерактивних онлайн-тренажерів.....	5
1.2 Аналіз аналогів.....	6
1.3 Технічні вимоги до розробки онлайн-тренажера.....	10
1.4 Висновки до розділу 1.....	11
РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО ОНЛАЙН-ТРЕНАЖЕРА ДЛЯ ВИВЧЕННЯ ВЛАСТИВОСТЕЙ CSS.....	12
2.1 Опис та обґрунтування вибору інструментів для розробки.....	12
2.2 Опис програмної реалізації додатка.....	14
2.3 Тестування.....	21
2.4 Інструкція щодо використання.....	23
2.5 Висновки до розділу 2.....	26
ВИСНОВКИ.....	27
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	29
ДОДАТКИ.....	31

ВСТУП

Актуальність дослідження. Традиційні методи вивчення вебпрограмування часто виявляються менш ефективними у сучасних швидкозмінних освітніх реаліях. Для підвищення продуктивності та скорочення часу, необхідного на освоєння вебтехнологій, широко використовуються різноманітні онлайн-ресурси.

Онлайн-платформи забезпечують швидкий та зручний доступ до нової інформації, корисної у різних сферах життя, зокрема й у вебпрограмуванні.

Існують спеціалізовані онлайн-тренажери, призначені для навчання основ вебпрограмування, серед яких значну популярність мають тренажери з вивчення CSS [5]. Такі додатки сприяють появі нових веброзробників, здатних швидко та ефективно створювати вебсайти різного типу.

Мета: розробка інтерактивного тренажера для вивчення CSS у вигляді вебдодатка, що дозволяє користувачам практикувати використання CSS властивостей в ігровому середовищі.

Для досягнення поставленої мети сформульовані **завдання:**

1. Аналіз наявних аналогів інтерактивних тренажерів для вивчення CSS та створення технічних вимог проєкту.
2. Вибір інструментів та засобів для розробки.
3. Проєктування структури та інтерфейсу вебдодатку.
4. Реалізація функціональності тренажера.
5. Розробка інструкції користувача.

Об'єктом дослідження є процес розробки інтерактивного онлайн-тренажера для вивчення властивостей CSS.

Предметом дослідження є використання вебтехнологій (HTML, CSS, JavaScript) для створення інтерактивного тренажера, що дозволяє користувачам практикуватися у використанні властивостей CSS через ігрові елементи.

РОЗДІЛ 1. ТЕОРЕТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Класифікація інтерактивних онлайн-тренажерів

Класифікація тренажерів для вивчення CSS досить різноманітна, та залежить від критеріїв, які обираються [1].

Тренажери для вивчення CSS можна класифікувати за **типом взаємодії**:

- **тренажери з візуальним редактором** – такі тренажери дозволяють бачити результат змін коду в режимі реального часу, що дозволяє швидко та чітко змінювати все, що потрібно. Такими тренажерами є CodePen, JSFiddle та CSSBattle;

- **тренажери з текстовим редактором** – ці тренажери орієнтовані на написання коду, де є функція автодоповнення коду, що допомагає уникнути помилок. Такими тренажерами є FreeCodeCamp та Codewars;

- **тренажери-ігри** – тренажери, які пропонують вивчати CSS в формі ігор, виконуючи завдання та проходячи рівні, що підвищує мотивацію в учнів. Такими тренажерами є CSS Diner, Flexbox Froggy, Grid Garden.

Також, можна виділити тренажери за **рівнем складності в користуванні**:

- **для початківців** – такі тренажери знайомлять з базовими поняттями CSS, вони пропонують прості вправи на стилізацію тексту, кольору та шрифту. Такими тренажерами є сайти W3School та Mozilla Developer Network;

- **середнього рівня** – тренажери такого рівня складності дозволяють поглиблено вивчати CSS, по типу вивчення псевдокласів, елементів, позиціонування, анімацій та медіазапитів. Такими тренажерами є Codewars та Frontend Mentor;

- **просунутий рівень** – ці тренажери орієнтовані на вивчення складних тем, таких як, Grid, Flexbox, псевдопроцесори CSS (Sass, Less).

Такими тренажерами є CSSbattle та CodinGame.

За спеціалізацією тренажери для вивчення CSS діляться на:

- **тренажери для вивчення *Flexbox*** – зосереджені на вивченні гнучкої сітки Flexbox. Такими тренажерами є Flexbox Froggy, Flexbox Zombies;
- **тренажери для вивчення *Grid*** – допомагають освоїти сітку Grid для створення складних макетів. Такими тренажерами є Grid Garden, CSS Grid Generator;
- **тренажери для вивчення *анімації*** – пропонують вправи на створення анімаційних ефектів за допомогою CSS. Такими тренажерами є Animista, CSS Animations.

За форматом навчання тренажери можна поділити на:

- **інтерактивні тренажери** – дозволяють взаємодіяти з кодом в режимі реального часу, отримувати зворотний зв'язок та підказки;
- **тренажери з тестами** – пропонують перевірити свої знання CSS за допомогою тестів та завдань. Такими тренажерами є W3Schools Quizzes, Quizlet;
- **тренажери з відеоуроками** – поєднують теорію та практику, пропонуючи відеоуроки та вправи для закріплення матеріалу. Курси на платформах Coursera, Udemy, freeCodeCamp.

1.2 Аналіз аналогів

Одними з найпопулярніших додатків для вивчення CSS є CodePen, freeCodeCamp, CSS Diner, Flexbox Froggy, Grid Garden, CSSBattle [8].

CodePen [10] – це онлайн-платформа, яка дозволяє вивчати та писати код CSS в інтерактивному середовищі. Цей вебдодаток призначений для всіх рівнів (від початківців до просунутих). Спеціалізація на загальний CSS та фронтенд-розробку. CodePen має зручний інтерфейс де миттєво відображується написаний користувачем код. Цей додаток має спільноту в

якій можна поділитись своїми проєктами та оцінити роботу інших користувачів. Крім таких переваг в CodePen є і ряд недоліків, наприклад, у цьому додатку обмежена кількість навчального матеріалу, також цей додаток не підходить для безпосереднього вивчення CSS, він більше призначений для створення основ сайтів з особливими дизайнами (див. Табл. 1.1).

freeCodeCamp [11] – це інтерактивний тренажер, який поєднує текстовий редактор з уроками, які містять різноманітні завдання. Тренажер для початківців та середнього рівня знань. Спеціалізація тренажера – загальний CSS та фронтенд-розробка. Додаток має простий інтерфейс та багато навчального матеріалу, що допомагає користувачам більше поглибитись в навчання CSS. Попри це додаток freeCodeCamp має мало можливостей для візуального експериментування з власно створеними сайтами та не підходить для вивчення окремих аспектів CSS (див. Табл. 1.1).

CSS Diner [12] – це тренажер-гра. Тренажер призначений для початківців. Спеціалізація тренажера – селектори CSS. Додаток має ігрову форму навчання де завдання представляються у вигляді рівнів, чим вище рівень, тим складніше буде завдання. CSS Diner наочно демонструє роботу селекторів і це є також мінусом додатка через те, що він тільки спрямований на вивчення селекторів, що значить цей додаток не призначений для вивчення інших аспектів CSS (див. Табл. 1.1).

Flexbox Froggy [13] – також можна віднести до типу тренажер-гра. Для початківців та середнього рівня навичок та знань. Тренажер спеціалізований на вивченні Flexbox. Додаток має вигляд гри з виконавцем команд-жабою, з якою взаємодіє користувач при виконанні завдань. Завдання спрямовані лише на окремі фрагменти коду, а не на повне написання сайтів. Flexbox Froggy підходить для початкового вивчення Flexbox аспекту та не має інших функціоналів та матеріалів (див. Табл. 1.1).

Grid Garden [14] – цей тренажер також ігрового типу, схожий на Flexbox Froggy, призначений для початківців та середнього рівня навичок та знань. Спеціалізація тренажера – вивчення Grid. Додаток має функціонал,

схожий на Flexbox Froggy, що означає, що його можливості для вільного програмування певним чином є обмеженими. Відмінність від Flexbox Froggy тільки в тому, що Grid Garden спрямований на вивчення Grid (див. Табл. 1.1).

CSSBattle [15] – це візуальний редактор-тренажер у формі змагальних випробувань, розроблений для користувачів із середнім і просунутим рівнем знань та навичок CSS. Спеціалізація тренажера – вивчення загального CSS та оптимізація коду.

Табл. 1.1. Переваги та недоліки аналогів онлайн-тренажерів

Назва додатку	Переваги	Недоліки
CodePen	- зручний інтерфейс з миттєвим відображенням результату; - присутня спільнота, в якій знаходяться роботи інших користувачів; - можливість створювати особливі проекти, якими можна поділитись; - інтеграція з інструментами препроцесорів CSS	- обмежена кількість навчального матеріалу; - не підходить для вивчення CSS; - призначений лише для створення основ для сайтів;
freeCodeCamp	- простий інтерфейс користувача; - безкоштовний доступ до матеріалів сайту; - структурований навчальний план з інтерактивними вправами та проектами;	- має мало можливостей для експериментів з візуалізацією сайтів; - не підходить для вивчення окремих аспектів CSS;
CSS Diner	- форма навчання програмуванню CSS у вигляді гри; - завдання поступово стають складнішими; - наочне демонстрування роботи селекторів;	- функціонал обмежений тільки вивченням селекторів; - не підходить для вивчення інших аспектів CSS;
Flexbox Froggy	- ігрова форма навчання; - візуальне представлення результатів завдань;	- спеціалізація додатку тільки на вивчення Flexbox;

Назва додатку	Переваги	Недоліки
	- поступове збільшення складності завдань;	
Grid Garden	- ігрова форма навчання; - візуальне представлення результатів завдань; - поступове збільшення складності завдань;	- спеціалізація додатку тільки вивчення Grid;
CSSBatle	- вивчення CSS у змагальній формі; - розвивання навичок оптимізації коду; - можливість переглядати результати інших користувачів;	- не підходить для початківців; - вимагає знання різних технік CSS;

Для створення власного сайту потрібно обрати кілька механік з цих аналогів, які підійдуть для швидкого вивчення CSS і зможе надати тренажеру аудиторію для багатьох учнів середніх та старших класів. Сайт має містити поля для вводу коду та відображення виконання цього коду, щоб користувач не використовував сторонні програми та не засмічував пам'ять свого пристрою.

Тренажер має містити рівневі завдання, щоб розвивати у користувача логічне мислення, навички написання структурованого коду, розуміння принципів роботи CSS та поступово підвищувати рівень його знань і впевненості у використанні мови стилів.

Тренажер не має зациклюватись на вивченні лише одного аспекту CSS, він має бути різносторонній. Також у тренажері можна створити кілька видів вивчення програмування та об'єднати ігрову форму зі стандартним вивченням програмування виконанням простих завдань, де користувач сам створює код та бачить свої власні розробки. В ігровій же формі користувачу буде надано готову картинку сайту, де йому треба лише доповнити код для виконання завдання.

1.3 Технічні вимоги до розробки онлайн-тренажера

Для забезпечення популярності онлайн-тренажера необхідно визначити його цільову аудиторію. У цьому випадку нею є учні середніх і старших класів (а саме з 8 по 11 класи), які вже мають уявлення про програмування та можуть бути зацікавлені у створенні власних вебсайтів або ж у вивченні CSS зокрема.

Вебсайт повинен відображатися коректно на всіх типах пристроїв, включаючи комп'ютери, планшети, смартфони тощо. Під час розробки онлайн-тренажера необхідно враховувати сучасні вебтехнології, такі як HTML5, CSS3, JavaScript, а також можливість використання фреймворків [2].

Окрім того, сайт повинен підтримувати кілька мов, щоб зробити вивчення CSS доступним не лише для українськомовних користувачів, але й для користувачів з інших країн, які не володіють українською. У розробленому сайті планується реалізувати підтримку двох мов – української та англійської.

Структурно сайт складатиметься з головної сторінки, яка міститиме привітання користувача, опцію вибору режиму завдань та навігаційні елементи для переходу між розділами сайту.

Система завдань поділена на рівні складності та пропонує два режими: ігровий і звичайний. В ігровому режимі користувач може, наприклад, інтерактивно переміщувати об'єкти для їхнього застосування у певній ситуації, як-от складання піци шляхом перетягування інгредієнтів до миски або розставляння книг на полиці. Звичайний режим передбачає наявність текстового завдання, відповідно до якого користувач у спеціальному полі вводить програмний код, а система здійснює його перевірку на відповідність умові завдання.

Сайт матиме унікальну назву та логотип [2]. У «шапці» сайту розміщені навігаційні кнопки, такі як «Інформація», «Меню» та «Інструкція користувача».

Меню сайту стає доступним за кількох умов: при натисканні на спеціальну кнопку в «шапці» сайту, а також при натисканні на кнопку, розташовану після привітального тексту на головній сторінці.

При виборі завдання сайт переходить в інтерактивний режим, де відображається текст завдання, поле для введення коду та поле для візуалізації результату виконання коду. Крім того, присутні кнопки для перевірки правильності виконання завдання, отримання підказки та навігації між завданнями.

1.4 Висновки до розділу 1

Класифікація тренажерів для вивчення CSS досить різноманітна та залежить від критеріїв, які обираються [1]. Найпопулярнішими додатками для вивчення CSS є CodePen, freeCodeCamp, CSS Diner, Flexbox Froggy, Grid Garden, CSSBattle [6]. Для створення власного сайту потрібно обрати кілька механік з цих аналогів, які підійдуть для вивчення CSS у швидкий спосіб та зможуть відкрити тренажер для цільової аудиторії (учнів гімназій та ліцеїв, які вивчають вебтехнології). Тренажер має містити рівневі завдання, щоб розвивати логічне мислення, навички написання структурованого коду, розуміння принципів роботи CSS та поступово підвищувати рівень знань і впевненості користувача у використанні мови стилів. Тренажер не має зациклюватись на вивченні лише одного аспекту CSS, він має бути універсальним. Сайт повинен коректно відображатись як на комп'ютерах, так і на інших пристроях таких як планшети, смартфони та інших. Також сайт має підтримувати кілька мов щоб вивчення CSS було доступне не тільки українським користувачем, а і користувачам інших країн, які не знають української. Сайт повинен мати свою унікальну назву та логотип.

РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО ОНЛАЙН-ТРЕНАЖЕРА ДЛЯ ВИВЧЕННЯ ВЛАСТИВОСТЕЙ CSS

2.1 Опис та обґрунтування вибору інструментів для розробки

Для розробки сайту обрали середовище програмування Visual Studio Code (VS Code). Це середовище забезпечує підтримку широкого спектра мов програмування, зокрема HTML та JavaScript, необхідних для створення сайту-тренажера.

Окрім підтримки багатьох мов, VS Code вирізняється зручною роботою з файлами, що значно полегшує створення структури сайту, яка може складатися з багатьох файлів. Кожен файл, залежно від його типу (мови програмування), має відповідну іконку, що спрощує навігацію між великою кількістю файлів (рис. 2.1).

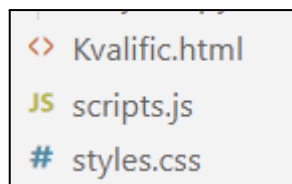


Рис 2.1. Відображення різних форматів файлів у середовищі VS Code

Крім того, це середовище розробки здатне виявляти помилки, яких припускається користувач під час написання коду. Це значно полегшує пошук і виправлення синтаксичних, логічних, функціональних та інших помилок у програмному коді. До того ж VS Code не лише виявляє помилки, але й може пропонувати шляхи їх усунення. Це може бути посилання на довідкову інформацію щодо конкретної помилки або пропозиція щодо вставки іншої команди або фрагмента коду замість помилкового (рис. 2.2).



Рис. 2.2. Помилка та її опис

VS Code вирізняється інтуїтивно зрозумілим інтерфейсом. До того ж це середовище має ефективну механіку автозавершення коду, що значно пришвидшує процес написання та полегшує пошук необхідних команд. Також, завдяки кольоровому виділенню фрагментів коду (так званий «підсвітці» синтаксису), спрощується навігація у великих обсягах програмного коду (рис. 2.3).

```
.header {
  background-color:
  height: 10%;
```

Рис. 2.3. Розділення фрагментів коду за кольорами

Це середовище також пропонує функціональність, яка полегшує роботу над дизайном сайту. Зокрема, воно відображає кольори, що використовуються у коді, що є значною перевагою при виборі колірної схеми оформлення. Крім того, окрім відображення кольору, при натисканні на відповідну кнопку користувач може відкрити палітру кольорів і підібрати потрібний відтінок (рис. 2.4).

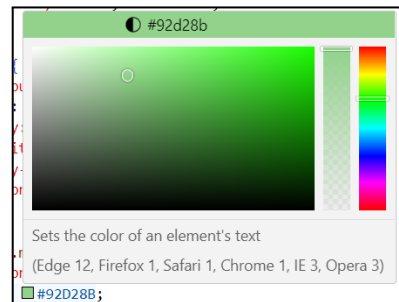


Рис. 2.4. Кольорова палітра додатка Visual Studio Code

Також було використано сайт Colors.co [18] для того, щоб забезпечити унікальний вигляд сайту, за допомогою генерації унікальних кольорів, які допомагають підібрати стиль.

Якщо обрати вдалу, професійно підбрану колірну палітру (рис. 2.5), можна освіжити сайт і водночас зберегти гармонію кольорів. Окрім цього, буде зберігатись унікальність палітри, тому що другу таку саму палітру

отримати фактично неможливо через функцію сайту заблокування окремих кольорів.

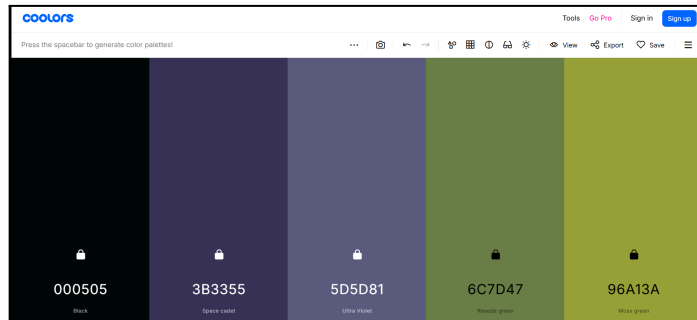


Рис. 2.5. Сайт Colors.co

2.2 Опис програмної реалізації додатка

Для ефектного та простого освоєння основ CSS у розробці сайту спробували об'єднати два типи тренажерів ігровий та редактор з вільним кодуванням.

Преш за все пропонуємо лаконічний макет сайту, який має просту структуру та складається з трьох основних блоків (рис. 2.6). Перший блок головний, у ньому знаходиться кнопка меню навігації, назва сайту та логотип.

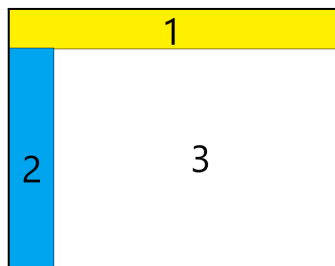


Рис. 2.6. Макет сайту

Далі потрібно обрати назву сайту, назва розроблюваного нами сайту «CssLearner». Ця назва описує простоту CSS та заохочує його вивченню. Окрім цього потрібен логотип сайту. Щоб виділити на фоні інших тренажерів даний проєкт, було створено логотип (рис. 2.7). Він вирізняється лаконічністю та складається з трьох літер, що відображають основну тематику сайту – вивчення CSS. Кожна літера має індивідуальний колір і розмір, що надає логотипу оригінальності та стильності.



Рис. 2.7. Логотип сайту

Для полегшення навігації по коду кожна сторінка сайту описана трьома файлами. Файл HTML, який відповідає за структуру та розмітку сторінки, об'єднуючи всі її елементи. Файл CSS містить стилі, що визначають зовнішній вигляд та розташування компонентів на сторінці. Файл JavaScript, включає функції та сценарії, необхідні для забезпечення інтерактивності та динамічної поведінки сайту.

У головному файлі HTML описуються усі кнопки та назви в окремих класах. Наприклад, кнопка меню яка є навігацією по сайту та завданням має клас – menu, а кожне завдання в цьому меню мають клас – menu-item.

Головне меню забезпечує навігацію по всіх сторінках сайту (їх 51). Кожен пункт меню є кнопкою, що спрямовує користувача на відповідну сторінку із завданням, які були створені на основі задач шкільного курсу для 8 класу та старших класів, довідковою інформацією тощо. Крім того, завдання розподілені за рівнями складності: легкий рівень (20 завдань, позначені зеленим кольором – рис. 2.8), середній рівень (15 завдань, позначені жовтим кольором – рис. 2.9) та складний рівень (10 завдань, позначені червоним кольором – рис. 2.10). Також в меню присутні інтерактивні завдання (5 завдань, позначені синім кольором - рис. 2.11).

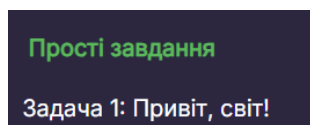


Рис. 2.8. Прості завдання

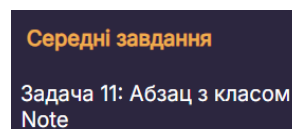


Рис. 2.9. Середні завдання

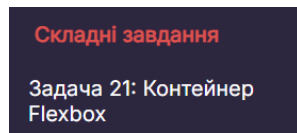


Рис. 2.10. Складні завдання

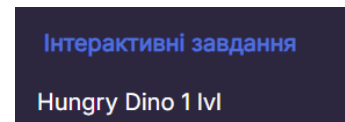


Рис. 2.11. Інтерактивні завдання

Також, для коректної роботи сайту, потрібно під'єднати необхідні файли та ресурси: CSS-файл, що містить стилі, та спеціальні шрифти, імпортовані з Googleapis [16]. Нижче наведено код для підключення:

```
<link href="https://fonts.googleapis.com/css2?family=Inter&display=swap" rel="stylesheet">
```

Далі в описуваному файлі створені класи з кнопкою навігації по сайту, логотипом та назвою сайту. Було створено клас для кнопки меню, який має назву *menu-button*, клас для логотипа та назви сайту, він має назву *logo-container*, клас для панелі меню який має назву *menu-panel*, клас для основного тексту на головній сторінці сайту, де знаходиться привітання для користувачів, додатково було створено клас для кнопки початку курсу. Класи для головної сторінки будуть використовуватись лише для навігації та назви сайту з логотипом, інші класи будуть замінені та додані. Наприклад, для сторінки завдання було видалено клас привітання та кнопки для початку курсу, натомість було додано клас для редактора коду, який має підкласи для кнопок для перемикання між HTML та CSS кодом, а також кнопки запуску коду, перевірки завдання, та підказки. Було створено клас для зони показу коду та для панелі із завданням.

Після того як усі класи було створено, для головної сторінки було написано привітальні слова. Текст привітання має ось такий зміст: «CssLearner – це інтерактивний тренажер, що допоможе тобі вивчити HTML та CSS шляхом практики. Виконуй завдання, пиши код та одразу побачиш результат своїх дій. Почни з простих структур і поступово переходь до складніших макетів». Цей текст коротко описує суть тренажера, а також пояснює, що потрібно зробити користувачу для того, щоб зрозуміти CSS.

Після опису усіх класів, створення текстів, назви, логотипу та кнопок «збирається» головна сторінка. Весь сайт має темне оформлення, кнопки в основному мають синій колір для більш видимого та приємного оформлення. Ці властивості для кольору фону, тексту, та кнопок були описані в окремому CSS файлі для кожної підсторінки. Це було зроблено для того, щоб надалі можна було змінювати стиль кожної сторінки не змінюючи інші сторінки. Функціонал головної сторінки оформлений в окремому JavaScript файлі, він невеликий за обсягом і містить опис функцій опрацювання подій для трьох кнопок – це кнопка меню, кнопка старту, та контейнеру з логотипом та назвою сайту, це було зроблено для того, щоб додати можливість переходу до головної сторінки під час натискання на логотип та назву. Кнопка меню своєю чергою відкриває поле меню де знаходиться навігація по сайту. Кнопка «Старт» також відкриває дане меню, для того, щоб користувач самостійно обирав з якого завдання він хоче розпочати.

У кінцевому варіанті вигляд головної сторінки (рис. 2.12) має досить приємний дизайн. Вона має простий дизайн, але попри це на ній присутнє зображення динозавра, яке рухається вздовж нижньої частини сайту.

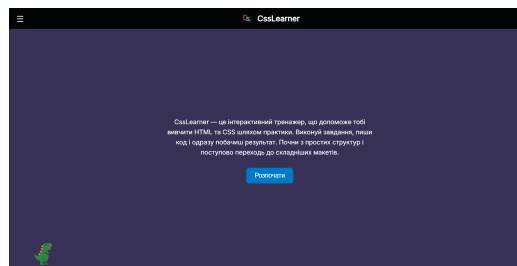


Рис. 2.12. Вигляд головної сторінки

Після того як буде натиснута кнопка «Розпочати» відкривається меню завдань (рис. 2.13) в якому будуть всі завдання даного тренажера, яких 30 і кожне завдання має свою тему, але вони між собою взаємопов'язані. Ці завдання були створені на основі шкільного курсу інформатики (зміст чинних підручників інформатики), наприклад завдання 4 «Списки», де описано створення списків мовою програмування HTML [3]. Завдання підвищеної

складності «Відсоткові розміри», яке передбачає використання команди `width width` [4]. Кожне завдання має унікальні підказки, що відповідають його меті, та власну систему перевірки. Це означає, що код, написаний для одного завдання, не буде коректним для іншого, оскільки функція перевірки використовує специфічні дані для кожного завдання. Крім того, кожне завдання розміщено на окремій сторінці.

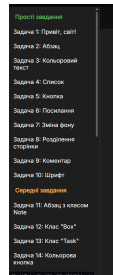


Рис. 2.13. Меню завдань

Перейдемо до сторінки із завданням. Вона має свій стиль внаслідок того, що в ній використовується більше елементів, це і редактор коду, і поле для відображення виконаного коду, і різні кнопки для виконання переходів між сторінками та редакторами коду HTML та CSS. Наповненість сторінки завдань, на відміну від головної сторінки, досить повна, адже майже немає пустого місця (рис. 2.14).

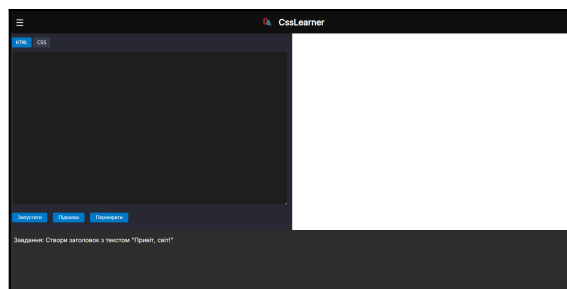


Рис. 2.14. Сторінка завдання

Завдання поділено на три рівні для ускладнення вивчення CSS, щоб користувач «не застряг» на одному рівні знань. Перший рівень це «Простий рівень», він містить 20 завдань, присвячених різним темам, від простого

написання тексту до створення ліній та зміни кольору фону. Другий рівень, «Середній рівень» містить також 15 завдань, які також мають взаємодію з текстовим оформленням сторінки в html, але також є завдання зі створення кнопок, зміни їх стилів, класів та динамічних змін кольорів. Останній рівень – «Складний рівень», містить 10 завдань, він складається із завдань взаємодії з FlexBox, а також створення прихованих блоків, анімацій та створення псевдокласів.

Кожне завдання передбачає перевірку правильності виконання за допомогою JavaScript-коду, що запускається натисканням кнопки «Перевірка». Наприклад, у першому завданні реалізовано порівняння введеного в редакторі тексту з еталонним значенням – «<h1>Привіт, світ!</h1>» (рис. 2.15). Такий принцип використовується в усіх завданнях. Варто зазначити, що деякі завдання навмисно не враховуватимуть стилізацію або точний текстовий вміст, надаючи користувачеві можливість для експериментів з кодом. Від того, яка відповідь буде введена, користувач отримає повідомлення «Правильно!» або «Спробуй ще...». Якщо відповідь невірна – користувачу потрібно переробити код в редакторі, якщо відповідь буде вірною, то з'явиться кнопка «Далі», яка дасть можливість користувачу перейти до наступного завдання.

```
if (h1 && h1.textContent.trim().toLowerCase() === 'привіт, світ!') {
  result.textContent = 'Правильно!';
}
```

Рис. 2.15. Код для порівняння правильної відповіді

Окрім традиційних завдань з написання коду, тренажер також включає інтерактивні вправи, аналогічні тим, що використовуються у тренажері Flexbox Froggy [11]. Ці вправи представлені інтерактивними завданнями під назвою «Hunger Dino». Сторінка із цими завданнями складається з двох основних зон: зони для написання коду та зони візуалізації результату (рис. 2.16).

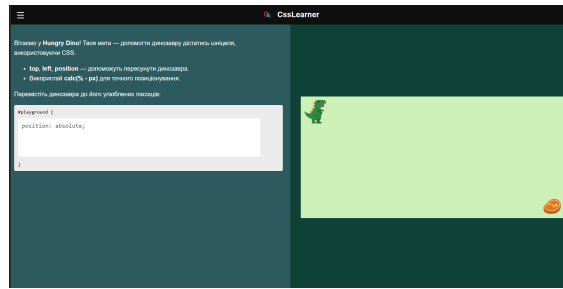


Рис. 2.16. Інтерактивна вправа «Hunger Dino»

У зоні написання коду розміщено текстовий опис завдання та поле для введення коду. Опис завдання містить підказки щодо команд, які необхідно використати для його виконання (див. Додаток Б).

У зоні візуалізації результату відображається «ігрове поле» та два зображення: динозавра і шніцеля (рис. 2.17–2.18). Динозавр розташований у верхньому лівому куті ігрового поля, а шніцель – у протилежному куті.



Рис. 2.17. Зображення динозавра



Рис. 2.18. Зображення шніцеля

Завдання користувача в цій вправі полягає в тому, щоб перемістити зображення динозавра до шніцеля для переходу до наступних завдань. Основна ідея цих вправ схожа на завдання у Flexbox Froggy [11]. Однак, у нашому тренажері рух зображень реалізовано шляхом написання повного CSS-коду, а не за допомогою команд flexbox. Таке рішення було прийнято з метою уникнення запозичення авторських розробок та плагіату.

Крім завдань, сайт також має невеликий довідник з описом та прикладами команд, які може зустріти користувач у даному тренажері (рис. 2.19). Цей довідник був створений для того, щоб користувач не витрачав час на пошук інформації про команди на інших сайтах, а одразу міг знайти те, що йому потрібно для виконання завдань у цьому тренажері.

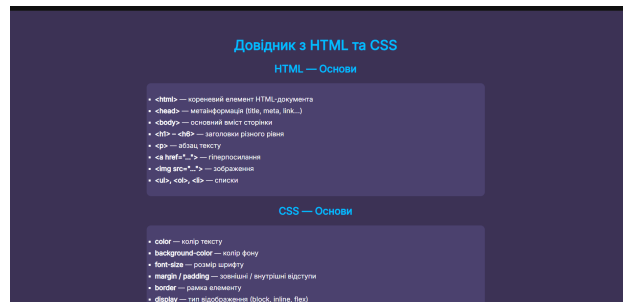


Рис. 2.19 Довідник команд

2.3 Тестування

Перейдемо до тестування сайту, під час завантаження сайту відкривається головна сторінка, на якій є привітальне слово, та кнопка «Розпочати» (рис. 2.20).

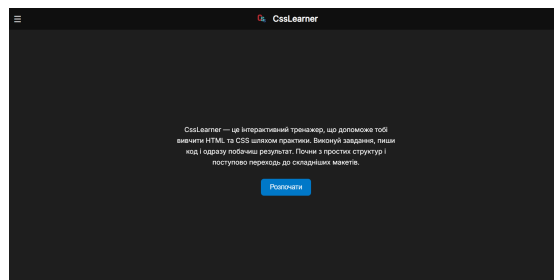


Рис. 2.20. Вигляд головної сторінки

Після натискання кнопки «Розпочати» відкривається меню (Див. рис. 2.13) з 30 завданнями, що поділені на три рівні складності. Перейдемо, наприклад, до завдання №2 (рис. 2.21). У даному завданні потрібно написати код в якому створюється абзац з текстом – «Я вивчаю HTML». Для перевірки працездатності кнопки запуску введемо, наприклад, такий код: «<h1>Привіт світ!</h1>» (рис. 2.22). Як бачимо, кнопка запуску коду працює коректно. Таким чином, ми також перевірили функціональність поля редагування коду та поля відображення результату.

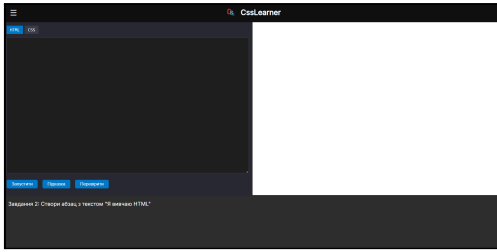


Рис. 2.21. Завдання № 2

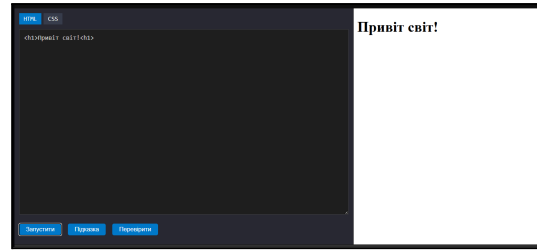


Рис. 2.22. Перевірка кнопки

Далі нам необхідно перевірити працездатність поля CSS. Для цього натиснемо кнопку «CSS», щоб перейти до відповідного поля редактора. У полі «CSS» введемо код: `<h1 {color: blue;}>`, натиснемо кнопку запуску та переглянемо результат (рис. 2.23). Як бачимо, текст у полі відображення результату став синім, що свідчить про коректну роботу поля редагування CSS.

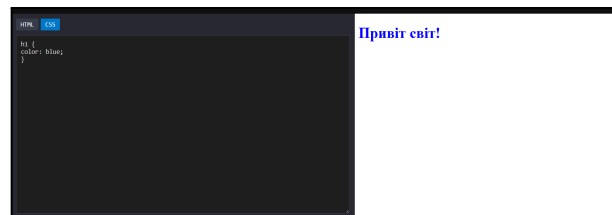


Рис. 2.23. Робота поля CSS

Перевіримо роботу кнопки «Підказка». При її натисканні має з'явитися повідомлення з підказкою по правильному виконанні завдання (рис. 2.24). Бачимо що виводиться текст жовтого кольору, що підтверджує працездатність цієї кнопки.

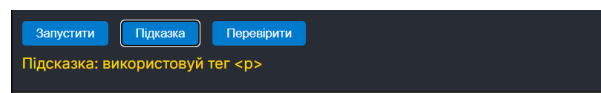


Рис. 2.24. Виведення підказки

Далі перевіримо роботу кнопки «Перевірка» та реакцію системи на правильний і неправильний код. Залишимо у вікні коду попередній код, натиснемо кнопку «Перевірка» (рис. 2.25). Бачимо повідомлення «Спробуй ще», що свідчить про те, що код не пройшов перевірку і спрацювала

відповідна дія для некоректного варіанту. Тепер введемо правильний код: «<p>Я вивчаю HTML</p>», натиснемо кнопку «Перевірка» та оцінімо результат (рис. 2.26). Як бачимо, результат виконання коду є вірним, і з'являється кнопка «Далі» для переходу до наступного завдання, а також повідомлення «Правильно!».

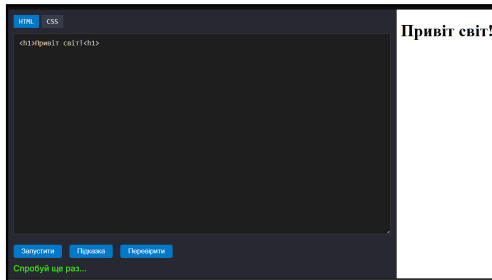


Рис. 2.25. Не правильний код

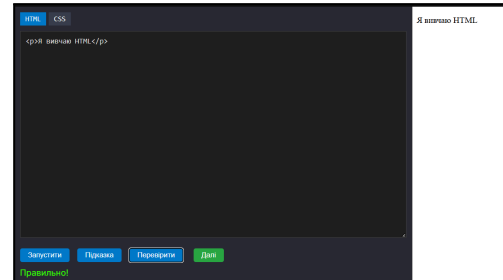


Рис. 2.26. Правильний код

Після перевірки усіх інших завдань можна зробити висновок про працездатність сайту та коректне функціонування всіх його елементів.

2.4 Інструкція щодо використання

Для того, щоб запустити тренажер потрібно запустити браузер та ввести в адресний рядок наступну адресу: <https://vitaliyi-21.github.io/>.

Найкраще використовувати такі браузери як:

- MS Edge;
- Google Chrome;
- Opera;
- Mozilla Firefox;

Після завантаження сторінки, відкриється проєкт, який використовує хостинг GitHub [17]. Після завантаження посилання відкривається головна сторінка тренажера. У верхній частині сторінки розташована «шапка» сайту, яка не змінюється в кожній сторінці та складається з кнопки «меню» з випадаючим списком завдань, а також логотипом і назвою тренажера.

На головній сторінці присутнє привітальне слово, яке потрібно

прочитати, щоб ознайомитись із тренажером. Після того як буде прочитано привітальне слово, потрібно натиснути на кнопку «Розпочати», яка відкриє список завдань. Список завдань складається із 31 завдання з написання коду, які поділені на три рівні, простий, середній та складний, а також 5 інтерактивних завдань.

Далі після відкриття меню завдань користувач може обрати рівень завдання, в кожному рівні завдань, є по 10 задач, кожна з яких охоплює вивчення HTML та CSS. Рекомендується починати з першого завдання, щоб ознайомитись з інтерфейсом сайту та з'ясувати як працює редактор коду.

Перейшовши до завдання на екрані буде зображено два поля – чорне та біле (рис. 2.27). Чорне поле – це поле для вводу коду програми, воно може змінювати свій розмір для цього достатньо потягнути за «трикутник» зображений в правому нижньому кутку поля. Біле поле – це поле, яке виводить результат виконання коду, який був написаний в чорному полі.



Рис. 2.27. Поля для вводу та виведення коду

Поле для написання коду має два режими — «HTML» та «CSS», які можна перемикає за допомогою кнопок у верхній частині (рис. 2.28). У режимі HTML користувач вводить html-код, який буде виконуватись у правому полі, а перейшовши у режим CSS, користувач вводить код, який буде змінювати оформлення його коду, написане в іншому режимі.

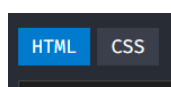


Рис. 2.28. Кнопки режимів

Окрім кнопок, які змінюють режим програмування також є 4 кнопки: «Запустити», «Перевірити», «Підказка» та «Далі» (рис. 2.29). Кнопка «Запустити» запускає виконання коду, який написаний у полі для вводу у двох режимах одночасно. Тобто якщо у полі HTML написаний код, а в полі CSS використовується особливий стиль для коду HTML то виконається одночасно два цих коди та відобразиться у полі виконання коду.

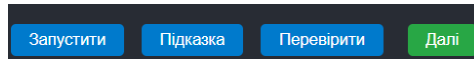


Рис. 2.29. Кнопки керування

Кнопка «Підказка» надає допомогу користувачеві, якщо він не знає, як виконати завдання. При її натисканні з'являється повідомлення жовтого кольору, що підказує, в якому напрямку потрібно виконувати розв'язання поточного завдання. Даний функціонал не вставляє частини коду, а лише допомагає розібратись, що потрібно писати та який напрямок правильніше буде обрати.

Кнопка «Перевірка» перевіряє чи виконано завдання правильно, тобто перевіряє чи написаний код, відповідає тій умові, яка зазначена в завданні. Дана кнопка виводить повідомлення «Правильно» або «Спробуй ще...» для того, щоб користувач зрозумів чи написаний ним код відповідає умові завдання.

Кнопка «Далі» з самого початку невидима, і для її появи потрібно виконати завдання правильно.

Під кнопками керування є поле де написане завдання для даної роботи, користувач одразу бачить, що потрібно зробити для того аби перейти далі.

Для переходу до наступного завдання користувач має правильно написати код, та натиснути кнопку «Перевірка», якщо код написано правильно, з'явиться кнопка «Далі», яка дозволить користувачу перейти до наступного завдання.

Користувач також може повернутись на головну сторінку натиснувши

на назву сайту або перейти до інших завдань натиснувши на кнопку меню.

Також, для того, щоб пригадати або вивчити нову інформацію про команди html або css користувач може натиснути в меню кнопку «Довідник».

2.5 Висновки до розділу 2

Для розробки сайту було обрано середовище програмування Visual Studio Code (далі – VS Code). Це середовище забезпечує підтримку широкого спектра мов програмування, зокрема HTML та JavaScript, необхідних для створення сайту-тренажера. Окрім підтримки багатьох мов, VS Code вирізняється зручною роботою з файлами, що значно полегшує створення структури сайту, яка може складатися з багатьох файлів. Кожен файл, залежно від його типу (мови програмування), має відповідну іконку, що спрощує навігацію між великою кількістю файлів.

Для ефектного та простого освоєння основ CSS у розробці сайту спробували об'єднати два типи тренажерів ігровий та редактор з вільним кодуванням. Наш проєкт отримав назву «CssLearner». Ця назва підкреслює простоту CSS та заохочує до його вивчення. Крім того, для ідентифікації сайту потрібен логотип. Щоб виділити наш ресурс серед інших, створили власний логотип (Див. рис. 2.7). Він вирізняється лаконічністю та складається з трьох літер, що відображають основну тематику сайту – вивчення CSS. Кожна літера має індивідуальний колір і розмір, що надає логотипу оригінальності та стильності. Під час тестувань тренажера проблеми, які виникали було усунуто, але вони були незначні, в основному це проблеми з розташуванням об'єктів. Усі інші функції та програмні особливості працюють без збоїв. Для того, щоб запустити тренажер потрібно запустити браузер та ввести в адресний рядок наступну адресу: <https://vitaliyi-21.github.io/>.

ВИСНОВКИ

Усі завдання кваліфікаційного проєкту були виконані, а саме:

1. **Проаналізовані аналоги та складені технічні вимоги проєкту.** Проведено порівняльний аналіз 5 популярних онлайн-тренажерів для вивчення CSS, та виявлено їхні сильні сторони (інтерактивність, гейміфікація) та недоліки (обмеженість у навчальному матеріалі, відсутність можливості вільного кодування). Визначено вимоги до розроблюваного тренажера, що враховують ці аспекти: поєднання інтерактивності з можливістю введення власного коду, розширений набір тем для вивчення.
2. **Обрані інструменти розробки.** Обґрунтовано вибір технологій HTML5, CSS3 та JavaScript для розробки інтерфейсу користувача. Обрано VS Code як середовище розробки, що забезпечує зручне редагування коду та інтеграцію з інструментами контролю версій.
3. **Спроектовані структура та інтерфейс.** Розроблено схему навігації сайту, що включає головну сторінку, сторінку з вибором завдань, сторінку з інтерактивним тренажером та сторінку з інструкцією користувача. Створено макети інтерфейсу для кожної сторінки, з акцентом на зручність користувача та інтуїтивність розташування елементів.
4. **Реалізований основний функціонал тренажера.** Реалізовано режим навчання з покроковим вивченням CSS-властивостей, починаючи з базових і поступово переходячи до складніших. Розроблено інтерактивні вправи, що дозволяють користувачам застосовувати отримані знання на практиці (наприклад, переміщення об'єктів, зміна стилів). Створено систему автоматичної перевірки коду, що надає користувачам зворотний зв'язок про правильність виконання завдання. Забезпечено можливість перегляду підказок та прикладів коду для полегшення процесу навчання.

5. **Розроблена інструкція користувача.** Створено посібник користувача, що описує всі функціональні можливості тренажера та надає рекомендації щодо його використання. Інструкція містить текстові описи та скриншоти інтерфейсу для кращого розуміння.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основи CSS. URL:
<https://w3schoolsua.github.io/css/index.html#gsc.tab=0>.
2. Етапи створення вебсайтів URL:
<https://webcase.com.ua/uk/blog/iz-chego-sostoit-razrabotka-sajta/>.
3. Інформатика. Підручник для 8 кл. закладів загальної середньої освіти. Н. В. Морзе, О. В. Барна. Київ: УОВЦ «Оріон», 2021. 192 - 217 с.
URL: <https://pidruchnyk.com.ua/839-informatyka-8-klas-morze-2016.html>.
4. Інформатика. Підручник для 8-го класу закладів середньої освіти. О. В. Коршунова, І. О. Завадський. Київ: Видавничий дім «Освіта» 2021. 170-195 с. URL: <https://pidruchnyk.com.ua/1627-informatiya-8-klas-korshunova.html>.
5. Mayer, R. E. (2009). Multimedia learning. Cambridge University Press, 2009. 305 с. URL:
<https://www.cambridge.org/core/books/multimedia-learning/7A62F072A71289E1E262980CB026A3F9>.
6. Roblyer, M. D., & Doering, A. H. (2013). Integrating educational technology into teaching. Pearson Education, 2013. 492 с. URL:
<https://anyflip.com/ixqwb/ozrv/basic/>.
7. Clark, R. C., & Mayer, R. E. (2016). E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning. John Wiley & Sons, 2016. 492 с. URL:
<https://onlinelibrary.wiley.com/doi/book/10.1002/9781119239086>.
8. Best code editors for web development. 2024. URL:
<https://happyaddons.com/best-code-editors-for-web-development/>.
9. Free code editors for web development. 2024. URL:
<https://daily.dev/blog/10-best-free-code-editors-for-web-development-2024>.
10. The best place to build, test, and discover front-end code.
[Електронний ресурс] URL: <https://codepen.io>.

11. Learn to code, build projects, earn certifications. URL: <https://www.freecodecamp.org>.
 12. CSS Diner. [Электронный ресурс] URL: <https://flukeout.github.io>.
 13. Flexbox Froggy. URL: <https://flexboxfroggy.com/#uk>.
 14. Grid Garden. URL: <https://cssgridgarden.com>.
 15. CSSBattle. URL: <https://cssbattle.dev>.
 16. Googleapis. URL: <https://developers.google.com/apis-explorer>.
 17. GitHub. URL: <https://github.com>.
 18. Colors.co. URL: <https://colors.co/000505-3b3355-5d5d81-6c7d47-96a>
- 13a.
19. Литвиненко В. А. CssLearner. [Электронный ресурс]. URL: <https://vitaliyi-21.github.io/>.

ДОДАТКИ

Додаток А

Код файлу Main.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
  <title>CssLearner – Головна</title>
  <link rel="stylesheet" href="main.css" />
  <link
href="https://fonts.googleapis.com/css2?family=Inter&display=swap"
rel="stylesheet">
</head>
<body>
  <header>
    <button class="menu-button" onclick="toggleMenu()">≡</button>
    <div class="logo-container"
onclick="location.href='../Main.html'">
      
      <span class="logo-text">CssLearner</span>
    </div>
  </header>

  <div class="menu-panel" id="taskMenu">
    <div style="color: #5cb85c; font-weight: bold; padding: 8px 10px
2px;">Прості завдання</div>
    <div style="height: 2px; background-color: #5cb85c; margin: 2px
10px 8px;"></div>
    <a href="Q1/Q1.html">Задача 1: Привіт, світ!</a>
    <a href="Q2/Q2.html">Задача 2: Абзац</a>
    <a href="Q3/Q3.html">Задача 3: Кольоровий текст</a>
    <a href="Q4/Q4.html">Задача 4: Список</a>
    <a href="Q5/Q5.html">Задача 5: Кнопка</a>
    <a href="Q6/Q6.html">Задача 6: Посилання</a>
    <a href="Q7/Q7.html">Задача 7: Зміна фону</a>
    <a href="Q8/Q8.html">Задача 8: Розділення сторінки</a>
    <a href="Q9/Q9.html">Задача 9: Коментар</a>
    <a href="Q10/Q10.html">Задача 10: Шрифт</a>
```

```

    <div style="color: #f0ad4e; font-weight: bold; padding: 12px 10px
2px;">Середні завдання</div>
    <div style="height: 2px; background-color: #f0ad4e; margin: 2px
10px 8px;"></div>
    <a href="Q11/Q11.html">Задача 11: Абзац з класом Note</a>
    <a href="Q12/Q12.html">Задача 12: Клас "Box"</a>
    <a href="Q13/Q13.html">Задача 13: Клас "Task"</a>
    <a href="Q14/Q14.html">Задача 14: Кольорова кнопка</a>
    <a href="Q15/Q15.html">Задача 15: Вирівнювання тексту</a>
    <a href="Q16/Q16.html">Задача 16: Рамка об'єкта</a>
    <a href="Q17/Q17.html">Задача 17: Динамічна зміна кольору</a>
    <a href="Q18/Q18.html">Задача 18: Закруглення</a>
    <a href="Q19/Q19.html">Задача 19: Внутрішній відступ</a>
    <a href="Q20/Q20.html">Задача 20: Спільний стиль</a>

    <div style="color: #d9534f; font-weight: bold; padding: 12px 10px
2px;">Складні завдання</div>
    <div style="height: 2px; background-color: #d9534f; margin: 2px
10px 8px;"></div>
    <a href="Q21/Q21.html">Задача 21: Контейнер Flexbox</a>
    <a href="Q22/Q22.html">Задача 22: Селектор</a>
    <a href="Q23/Q23.html">Задача 23: Відсоткові розміри</a>
    <a href="Q24/Q24.html">Задача 24: Приховані блоки</a>
    <a href="Q25/Q25.html">Задача 25: Додавання тіней</a>
    <a href="Q26/Q26.html">Задача 26: Верхні та нижні відступи</a>
    <a href="Q27/Q27.html">Задача 27: Псевдокласи</a>
    <a href="Q28/Q28.html">Задача 28: Псевдокласи</a>
    <a href="Q29/Q29.html">Задача 29: Зміна кольору фону при
натисканні</a>
    <a href="Q30/Q30.html">Задача 30: Анімація</a>
</div>

<main>
  <p class="intro-text">
    CssLearner – це інтерактивний тренажер, що допоможе тобі вивчити
HTML та CSS шляхом практики. Виконуй завдання, пиши код і одразу
побачиш результат. Почни з простих структур і поступово переходь до
складніших макетів.
  </p>

  <button class="start-button"
onclick="toggleMenu()">Розпочати</button>
</main>

```



```
<script src="main.js"></script>
</body> </html>
```

Додаток Б

Код до інтерактивної вправи “Hunger Dino”

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Dino Flexbox</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header class="main-header">
    <button class="menu-button" onclick="toggleMenu()">≡</button>
    <div class="logo-container"
onclick="location.href='../Main.html'">
      
      <span class="logo-text">CssLearner</span>
    </div>
  </header>

  <div class="menu-panel" id="taskMenu">
    <div style="color: #5cb85c; font-weight: bold; padding: 8px 10px
2px;">Прості завдання</div>
    <div style="height: 2px; background-color: #5cb85c; margin: 2px
10px 8px;"></div>
    <a href="../Q1/Q1.html">Задача 1: Привіт, світ!</a>
    <a href="../Q2/Q2.html">Задача 2: Абзац</a>
    <a href="../Q3/Q3.html">Задача 3: Кольоровий текст</a>
    <a href="../Q4/Q4.html">Задача 4: Список</a>
    <a href="../Q5/Q5.html">Задача 5: Кнопка</a>
    <a href="../Q6/Q6.html">Задача 6: Посилання</a>
    <a href="../Q7/Q7.html">Задача 7: Зміна фону</a>
    <a href="../Q8/Q8.html">Задача 8: Розділення сторінки</a>
    <a href="../Q9/Q9.html">Задача 9: Коментар</a>
    <a href="../Q10/Q10.html">Задача 10: Шрифт</a>

    <div style="color: #f0ad4e; font-weight: bold; padding: 12px 10px
2px;">Середні завдання</div>
```

```

    <div style="height: 2px; background-color: #f0ad4e; margin: 2px
10px 8px;"></div>
    <a href="../Q11/Q11.html">Задача 11: Абзац з класом Note</a>
    <a href="../Q12/Q12.html">Задача 12: Клас "Box"</a>
    <a href="../Q13/Q13.html">Задача 13: Клас "Task"</a>
    <a href="../Q14/Q14.html">Задача 14: Кольорова кнопка</a>
    <a href="../Q15/Q15.html">Задача 15: Вирівнювання тексту</a>
    <a href="../Q16/Q16.html">Задача 16: Рамка об'єкта</a>
    <a href="../Q17/Q17.html">Задача 17: Динамічна зміна кольору</a>
    <a href="../Q18/Q18.html">Задача 18: Закруглення</a>
    <a href="../Q19/Q19.html">Задача 19: Внутрішній відступ</a>
    <a href="../Q20/Q20.html">Задача 20: Спільний стиль</a>

    <div style="color: #d9534f; font-weight: bold; padding: 12px 10px
2px;">Складні завдання</div>
    <div style="height: 2px; background-color: #d9534f; margin: 2px
10px 8px;"></div>
    <a href="../Q21/Q21.html">Задача 21: Контейнер Flexbox</a>
    <a href="../Q22/Q22.html">Задача 22: Селектор</a>
    <a href="../Q23/Q23.html">Задача 23: Відсоткові розміри</a>
    <a href="../Q24/Q24.html">Задача 24: Приховані блоки</a>
    <a href="../Q25/Q25.html">Задача 25: Додавання тіней</a>
    <a href="../Q26/Q26.html">Задача 26: Верхні та нижні відступи</a>
    <a href="../Q27/Q27.html">Задача 27: Псевдокласи</a>
    <a href="../Q28/Q28.html">Задача 28: Псевдокласи</a>
    <a href="../Q29/Q29.html">Задача 29: Зміна кольору фону при
натисканні</a>
    <a href="../Q30/Q30.html">Задача 30: Анімація</a>

    <div style="color: #4f6fd9; font-weight: bold; padding: 12px 10px
2px;">Інтерактивні завдання</div>
    <div style="height: 2px; background-color: #4f6fd9; margin: 2px
10px 8px;"></div>
    <a href="dino1.html">Hungry Dino №1</a>

</div>

<div class="page-content">
    <div class="left-panel">
        <div>
            <div class="instructions">
                <p>Вітаємо у <strong>Hungry Dino</strong>! Твоя мета –
допомогти динозавру дістатись шніцеля, використовуючи CSS.</p>
                <ul>

```

```

        <li><strong>top</strong>, <strong>left</strong>,
<strong>position</strong> – допоможуть пересунути динозавра.</li>
        <li>Використай <strong>calc(% - px)</strong> для точного
позиціонування.</li>
    </ul>
    <p>Перемістіть динозавра до його улюблених ласощів:</p>
</div>
<div class="editor-box">
    <div><code>#playground &#123;</code></div>
        <textarea class="code-input" id="code"
placeholder="position: absolute;"></textarea>
    <div><code>&#125;</code></div>
</div>
</div>

<div class="right-panel">
    <div class="playground" id="playground">
        <div class="dino"></div>
        <div class="food"></div>
    </div>
</div>
</div>

<script src="script.js"></script>
</body>
</html>

```

```

const textarea = document.getElementById('code');
const playground = document.getElementById('playground');
const dino = document.querySelector('.dino');
const food = document.querySelector('.food');
const runButton = document.getElementById('run');
const hintButton = document.getElementById('hint');
const checkButton = document.getElementById('check');
const preview = document.getElementById('preview');
const htmlCode = document.getElementById('html-code');
const cssCode = document.getElementById('css-code');
const result = document.getElementById('result');
const taskMenu = document.getElementById('taskMenu');

```

```

function switchTab(tab) {
    document.querySelectorAll('.tab').forEach(btn =>
btn.classList.remove('active'));

```

```

if (tab === 'html') {
  htmlCode.style.display = 'block';
  cssCode.style.display = 'none';
  document.querySelectorAll('.tab')[0].classList.add('active');
} else {
  htmlCode.style.display = 'none';
  cssCode.style.display = 'block';
  document.querySelectorAll('.tab')[1].classList.add('active');
}
}

function goToHome() {
  window.location.href = '../main.html';
}

function toggleMenu() {
  taskMenu.classList.toggle('open');
}

window.addEventListener('click', function (e) {
  if (!taskMenu.contains(e.target)
  !e.target.classList.contains('menu-button')) {
    taskMenu.classList.remove('open');
  }
});

document.querySelectorAll('#taskMenu a').forEach(link => {
  link.addEventListener('click', () => {
    taskMenu.classList.remove('open');
  });
});

textarea.addEventListener('input', () => {
  const userStyles = textarea.value;
  applySafeStyles(dino, userStyles);
  checkPosition();
});

function applySafeStyles(element, styles) {
  const temp = document.createElement('div');
  temp.style.cssText = styles;
  const allowed = ['top', 'left', 'right', 'bottom', 'transform',
'position', 'margin', 'translate'];
  allowed.forEach(prop => {
    if (temp.style[prop]) {

```

```

        element.style[prop] = temp.style[prop];
    }
});
}

function checkPosition() {
    const dinoRect = dino.getBoundingClientRect();
    const foodRect = food.getBoundingClientRect();

    const overlap = !(
        dinoRect.right < foodRect.left ||
        dinoRect.left > foodRect.right ||
        dinoRect.bottom < foodRect.top ||
        dinoRect.top > foodRect.bottom
    );

    if (overlap) {
        if (!document.getElementById('next-button')) {
            const next = document.createElement('button');
            next.id = 'next-button';
            next.textContent = '➡ Далі';
            next.style.marginTop = '10px';
            next.style.padding = '10px 20px';
            next.style.fontSize = '16px';
            next.style.fontWeight = 'bold';
            next.style.backgroundColor = '#fff';
            next.style.color = '#4caf50';
            next.style.border = '2px solid #4caf50';
            next.style.borderRadius = '6px';
            next.style.cursor = 'pointer';

            next.addEventListener('click', () => {
                window.location.href = 'next-level.html';
            });

            document.querySelector('.left-panel').appendChild(next);
        }
        else {
            const next = document.getElementById('next-button');
            if (next) next.remove();
        }
    }
}

body {

```

```
margin: 0;
font-family: Arial, sans-serif;
height: 100vh;
overflow: hidden;
display: flex;
flex-direction: column;
}

.main-header {
  display: flex;
  justify-content: center;
  align-items: center;
  background-color: #121212;
  padding: 10px 20px;
  position: relative;
}

.menu-button {
  position: absolute;
  left: 20px;
  font-size: 24px;
  color: white;
  background: none;
  border: none;
  cursor: pointer;
}

.menu-panel {
  position: fixed;
  top: 0;
  left: -260px;
  width: 250px;
  height: 100%;
  background-color: #1e1e1e;
  border-right: 1px solid #444;
  padding: 10px;
  display: flex;
  flex-direction: column;
  z-index: 1000;
  transition: left 0.3s ease-in-out;
  overflow-y: auto;
  scrollbar-color: #666 #1e1e1e;
}
```

```
.menu-panel.open {  
  left: 0;  
}  
  
.menu-panel a {  
  color: white;  
  text-decoration: none;  
  margin: 5px 0;  
  padding: 5px;  
  border-radius: 4px;  
}  
  
.menu-panel a:hover {  
  background-color: #2a2a2a;  
}  
  
.logo-container {  
  display: flex;  
  align-items: center;  
  gap: 10px;  
  cursor: pointer;  
}  
  
.logo {  
  width: 32px;  
  height: 32px;  
}  
  
.logo-text {  
  font-size: 20px;  
  font-weight: bold;  
  color: white;  
}  
  
.page-content {  
  display: flex;  
  flex: 1;  
  height: calc(100vh - 60px);  
  width: 100%;  
}  
  
.left-panel, .right-panel {  
  width: 50%;  
  height: 100%;  
}
```

```
    display: flex;
    flex-direction: column;
    box-sizing: border-box;
}

.left-panel {
    background-color: #2e5a5f;
    color: white;
    padding: 20px;
    justify-content: space-between;
}

.right-panel {
    background-color: #10433a;
    align-items: center;
    justify-content: center;
    display: flex;
}

.instructions {
    font-size: 16px;
    line-height: 1.6;
}

.instructions code {
    background-color: rgba(255, 255, 255, 0.2);
    padding: 2px 5px;
    border-radius: 4px;
    font-weight: bold;
}

.editor-box {
    background-color: #eee;
    padding: 10px;
    border-radius: 4px;
    color: #000;
    font-family: monospace;
}

.code-input {
    width: 90%;
    height: 80px;
    font-family: monospace;
    font-size: 16px;
}
```



```
border: none;
padding: 10px;
margin: 10px 0;
resize: none;
background: #fff;
}

.playground {
width: 90%;
height: 300px;
background-color: #cef0bc;
position: relative;
display: block;
padding: 10px;
}

.dino {
width: 80px;
height: 80px;
background-image: url('dino.png');
background-size: contain;
background-repeat: no-repeat;
position: absolute;
top: 0;
left: 0;
}

.food {
width: 60px;
height: 60px;
background-image: url('schnitzel.png');
background-size: cover;
background-repeat: no-repeat;
position: absolute;
right: 0;
bottom: 0;
}
```