

Міністерство освіти і науки України  
Гусятинський фаховий коледж  
Тернопільського національного технічного університету імені Івана Пуллюя

*Циклова комісія програмування*

ЗВІТ

до лабораторної роботи №1 з  
дисципліни «Вебтехнології»

Тема: «Вступ до курсу « Вебтехнології». Ознайомлення та встановлення  
редактора коду. »

Виконала:  
студентка групи П-31  
**Чавурська Єлизавета**

Гусятин, 2025

## **ЛАБОРАТОРНА РОБОТА №1**

Тема: Вступ до курсу «Веб-технології». Ознайомлення та встановлення редактора коду. Аналіз сучасних інструментів веб-розробки та стану веб-технологій на сьогоднішній день.

Мета: Отримати поняття, що таке веб-розробка, ознайомитись із сучасними тенденціями розробки веб-застосувань, а також отримати практичні навики роботи з різними редакторами коду.

### **ХІД РОБОТИ**

#### **Завдання 1**

Frontend – це публічна частина web-додатків (веб-сайтів), з якою користувач може взаємодіяти і контактувати напряму. Розробники фронтенду відповідають за ефективне впровадження візуальних компонентів на веб-сайті. Вони також виконують важливі завдання щодо розробки веб-сайтів, наприклад навігацію, кнопки або будь-що, що допомагає підвищити загальну видимість веб-сайту. Для цього HTML, JavaScript і CSS часто використовуються для забезпечення безперебійної роботи вашого веб-сайту. Це дозволяє користувачам вільно та комфортно взаємодіяти з веб-сайтом.

Backend – це програмно-апаратна частина проекту. Бекенд-розробники працюють над серверною розробкою. Вони зосереджуються на базах даних, сценаріях та архітектурі вебсайтів. Ці професіонали також стежать за тим, як працює сайт, як вносять зміни й оновлюють усі закулісні функції.

Оскільки Backend – це програмно-апаратна частина проекту, Frontend ж є клієнтською стороною призначеного для користувача інтерфейсу до програмно-апаратної частини проекту, тобто до бекенду. Іншими словами бекенд – це все те, що відбувається на стороні сервера і що залишається невидимим користувачеві (сам сервер теж є частиною бекенду, тільки апаратного). Звідси і назва front – це видиме спереду, back – це те, що приховано позаду, невидиме.

Параметр	Front-End (Клієнтська частина)	Back-End (Серверна частина)
Визначення	Відповідає за ефективне впровадження візуальних компонентів і взаємодію з користувачем.	Працює над серверною розробкою, зосереджується на базах даних, сценаріях та архітектурі.
Набори навичок	HTML, CSS і JavaScript.	Внутрішні мови програмування, такі як PHP, Java, .Net, а також знання бази даних, сервера, API тощо.
Професіонали	Проектують зовнішній вигляд веб-сайту, враховуючи дані користувачів та змінюючи їх шляхом тестування.	Розробляють програму, яка підтримує інтерфейс. Відповідають за підтримку, безпеку та керування вмістом.
Автономна служба	Не може бути запропонована окремо.	Може бути запропонована як окрема послуга в BaaS (Backend як послуга).
Мета	Переконатися, що веб-сайт доступний для всіх користувачів і залишається чуйним (адаптивним) на всіх пристроях.	Створити програму, що підтримує інтерфейс, та переконатися, що веб-сайт функціонує належним чином.
Повинен мати	Розуміння, як ефективно працювати над дизайном і UI/UX веб-сайту чи програми.	Вміння впроваджувати алгоритми та вирішувати проблеми, пов'язані з системою.
Найпопулярніші засоби	jQuery та HTML5.	MySQL і PHP.
Середня зарплата	\$104,405 на рік	\$120,798 на рік

Завдання 2

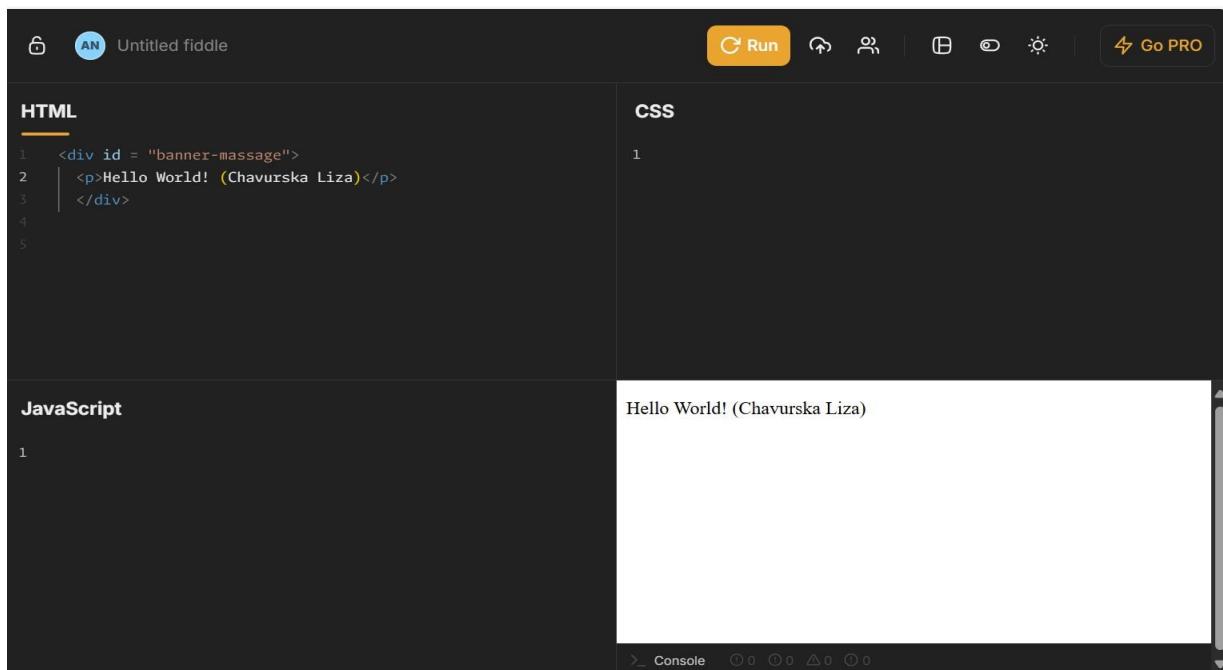
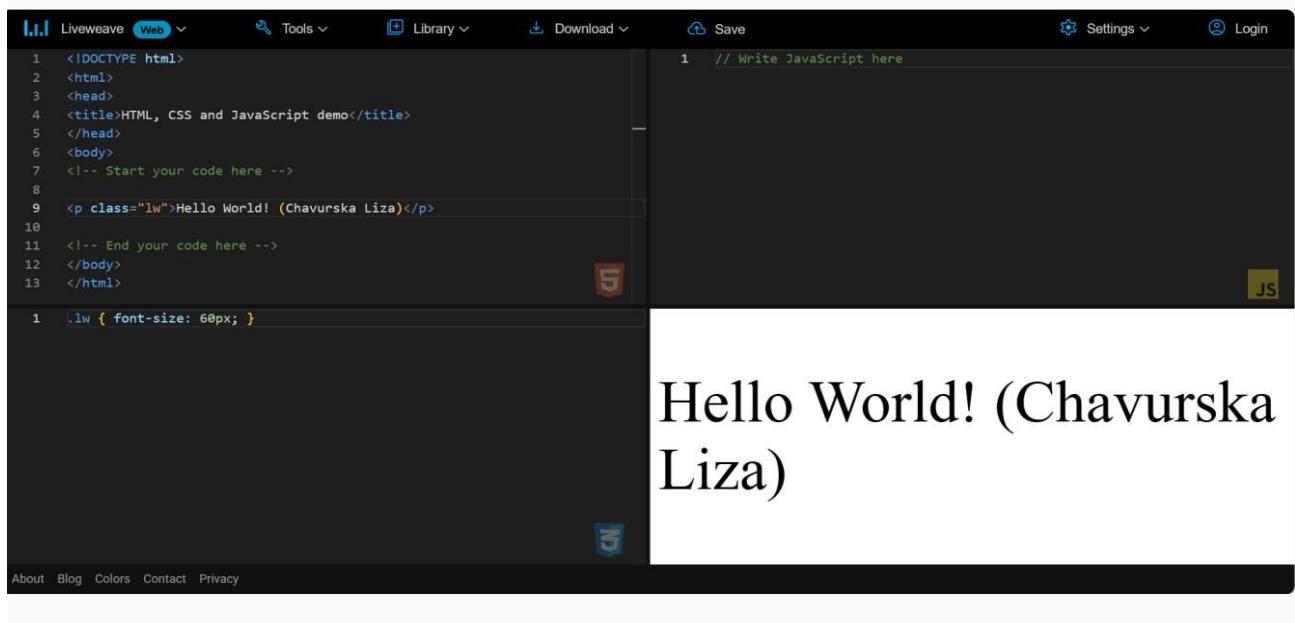


Рисунок 1.2 – Вікно інтерфейсу JSFiddle

JSFiddle - це те, на що це схоже: пісочниця, де можна пограти з JavaScript. Але якщо подивитися, як JavaScript йде рука об руку з HTML і CSS, можна редагувати всі три за допомогою інтерфейсу редагування JSFiddle - і взагалі пропустити JavaScript.

Що добре в JSFiddle, це що можна додати зовнішні запити в бічну панель, яка дозволяє включати сторонні файли JavaScript і CSS для поліпшення HTML. Також кнопка Tidy, яка автоматично очищає відступи коду, і кнопка Collaborate, яка дозволяє працювати над тим же кодом з кимось ще в режимі реального часу.



## 1.2 – Вікно інтерфейсу Liveweave

Liveweave має більш приємний інтерфейс. Як і JSFiddle, Liveweave дозволяє працювати в режимі реального часу і підключатися до певних сторонніх бібліотек, таким як jQuery. Але у нього є кілька унікальних функцій. Наприклад, генератор Lorem Ipsum для створення тексту риби на поточній позиції курсору. CSS Explorer надає візуальний редактор WYSIWYG для створення стилів. Color Explorer допоможе підібрати ідеальні кольори. А з допомогою Vector Editor можна створити векторну графіку для сайту.

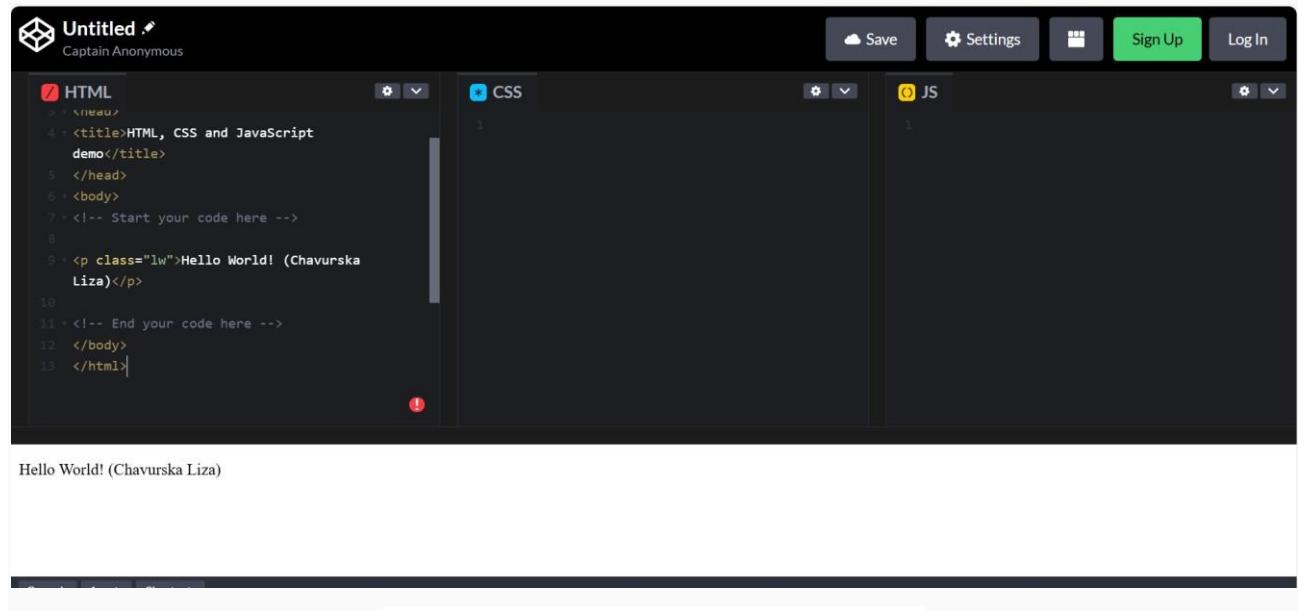


Рисунок 1.3 – Вікно інтерфейсу CodPen

На відміну від мінімалістичних інструментів (як-от JSFiddle), CodePen має сильний фокус на спільноті. Розробники публікують свої роботи, які називаються «Pens» (ручки/твори), як елементи портфоліо, змагаються та надихають один одного. Інтерфейс складається з трьох основних редакторів: HTML, CSS та JavaScript, а також панелі "Result" (Результат), де код виконується в режимі реального часу.

Підтримка Препроцесорів: CodePen підтримує майже всі популярні препроцесори для HTML (Pug/Jade), CSS (Sass, Less, Stylus) та JavaScript (Babel, TypeScript), що є великою перевагою для професійної розробки. Дозволяє легко підключати зовнішні бібліотеки та фреймворки (React, Vue, Three.js тощо) через налаштування кожного "Pen".

Онлайн-редактор JSdo.it є одним із ранніх і менш поширених, але схожих за функціональністю на інші "пісочниці" коду, такі як CodePen і JSFiddle. Як і інші платформи,

JSdo.it був створений для швидкого тестування написання та обміну Front-End кодом (HTML, CSS, JavaScript).

Платформа орієнтована на спільноту та дозволяє користувачам публікувати свої фрагменти коду (часто називаються "Рецептами") для демонстрації та взаємного навчання. Свого часу він був популярний серед розробників, які створювали невеликі креативні проекти та ефекти, подібно до CodePen. Як правило, пропонує стандартний інтерфейс з окремими панелями для HTML, CSS і JavaScript та вікном для результату виконання коду. Хоча JSdo.it був інноваційним на початку, він значно поступився популярністю таким гігантам, як CodePen і JSFiddle, і на сьогодні вважається менш активним та оновлюваним інструментом у порівнянні з основними конкурентами.

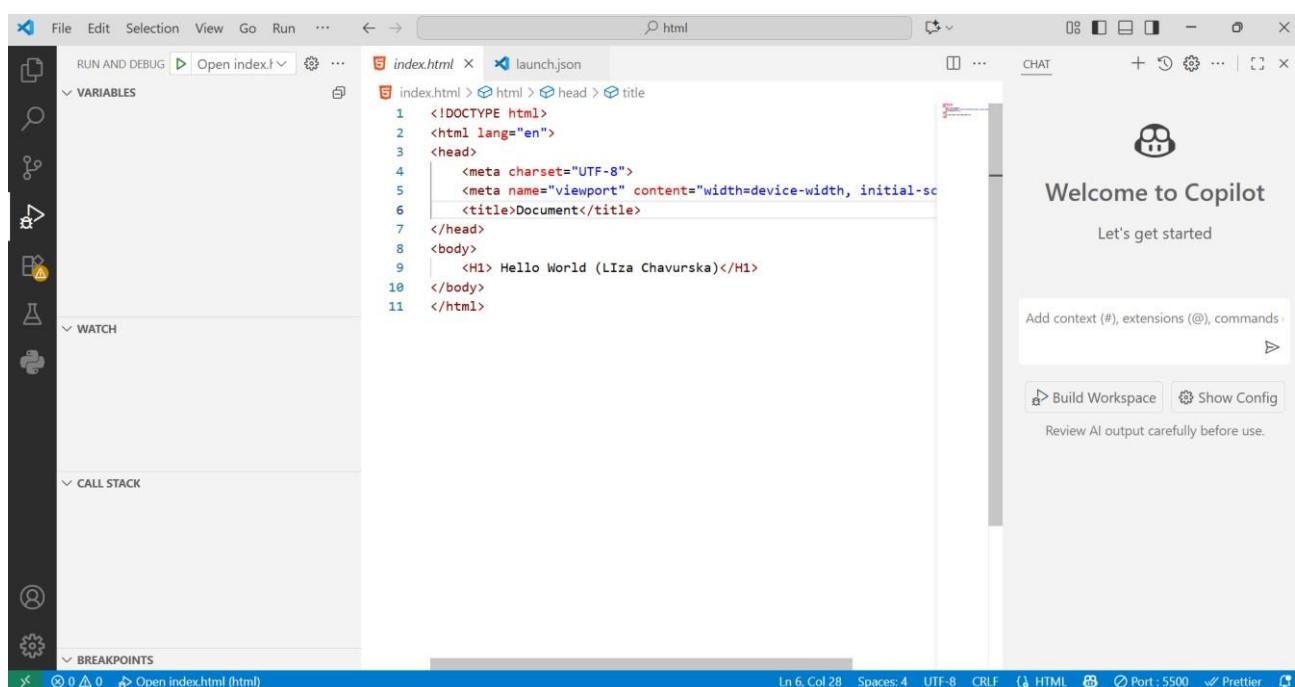


Рисунок 1.4 – Вікно інтерфейсу VS Code

Різні частини коду (теги, атрибути, значення) виділяються кольором, що значно полегшує читання та виявлення помилок. Коли ви починаєте вводити

назву тега чи атрибута, VS Code пропонує відповідні варіанти. Це прискорює кодування і мінімізує орфографічні помилки. Це вбудований інструмент, який дозволяє писати довгі HTML-структурі за допомогою коротких абревіатур. Наприклад, набравши `ul>li*4>a` і натиснувши Tab, ви миттєво отримаєте список з чотирьох елементів. VS Code може автоматично відформатувати ваш HTML, вирівнявши відступи і розташувавши теги відповідно до стандартів.

Можливість згортати блоки коду (наприклад, вміст тега `<body>`) допомагає орієнтуватися у великих файлах. При роботі зі стилями в HTML-документі, ви можете навести курсор на назву кольору, і VS Code відкриє графічний інтерфейс для його вибору.

### Завдання 3

Смарт-редактори та IDE редактори — це інструменти, які використовуються для написання програмного коду, але вони відрізняються за функціональністю та призначенням. Смарт-редактори — це легкі текстові редактори, які мають деякі можливості, наприклад підсвічування синтаксису, автодоповнення коду, перевірку помилок та можливість встановлення додаткових розширень. Вони зручні для швидкої роботи з кодом, не займають багато ресурсів і запускаються дуже швидко. Найвідомішими прикладами смарт-редакторів є Visual Studio Code, Sublime Text та Atom. Їх зручно використовувати для невеликих проектів або тоді, коли потрібно працювати з різними мовами програмування.

З іншого боку, IDE (інтегровані середовища розробки) — це більш складні інструменти, які включають у себе не тільки редактор коду, а й багато додаткових функцій: компілятор або інтерпретатор, систему налагодження (debugger), інструменти для тестування, автозбірки проекту, профілювання продуктивності, а також підтримку роботи з системами контролю версій, такими як Git. IDE створені для повного циклу розробки програм — від написання коду до його запуску, тестування та відлагодження. Прикладами IDE

є IntelliJ IDEA, PyCharm, Eclipse, Visual Studio та Xcode. Вони підходять для роботи над великими та складними проєктами, але при цьому потребують більше системних ресурсів і часу на запуск.

Отже, головна різниця полягає в тому, що смарт-редактори — це гнучкі, швидкі редактори для коду з базовими можливостями, які можна розширити, а IDE — це повноцінні середовища розробки з усім необхідним для програмування "під ключ". Вибір між ними залежить від потреб: для простих завдань краще використовувати смарт-редактор, а для серйозної розробки — IDE.

## ВИСНОВКИ

На цій лабораторній роботі я отримала поняття, що таке веб-розробка, ознайомилась із сучасними тенденціями розробки веб-застосувань, а також отримала практичні навики роботи з різними редакторами коду.

Вебтехнології, Лр1, Чавурська Єлизавета