ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп’ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

**КУРСОВА РОБОТА**

з дисципліни

«Web-дизайн та web-програмування»

на тему:

Розробка веб-сайту «Figma land»

Студента І курсу групи ІПЗс-11

спеціальності

121 – Інженерія програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Моспанюк Р.І.

(підпис)

Керівник: асистент А.В. Вознюк

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

**Луцьк 2023**

**Луцький національний технічний університет**

***Факультет комп’ютерних та інформаційних технологій***

***Кафедра інженерії програмного забезпечення***

Ступінь вищої освіти: *бакалавр*

Спеціальність: *121 «Інженерія програмного забезпечення»*

Дисципліна: *«Web-дизайн та web-програмування»*

Курс *І* Група *ІПЗс-11* Семестр *І*

***ЗАВДАННЯ***

**НА КУРСОВУ РОБОТУ СТУДЕНТУ**

\_ Моспанюк Роман Ігорович \_\_

(прізвище, ім’я, по батькові)

1. Тема роботи Розробка веб-сайту «Figma land»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник роботи: *асистент Вознюк Анастасія Вадимівна*

2. Строк подання студентом роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Вихідні дані до роботи: figma, html, scss, css, telegram, api\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) створити макет сайту у програмі Figma

обрати технології що будуть використовуватися при розробці

створити html розмітку та оформити її за допомогою scss стилів, реалізувати форму зворотнього зв’яку, протестувати сайт та виправити можливі помилки.\_\_\_

5. Перелік графічного матеріалу Презентація

1. Дата видачі завдання 04.11.2023

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ з/п** | **Назва етапів курсової роботи** | **Строк**  **виконання**  **етапів роботи** | **Примітка** |
| 1 | Отримання теми курсової роботи | 04.11.2023 |  |
| 2 | Узгодження постановки задачі з керівником | 04.11.2023 |  |
| 3 | Пошук та вивчення літератури з питань курсової роботи | 18.11.2023 |  |
| 4 | Вибір макета сайту | 18.11.2023 |  |
| 5 | Узгодження макета сайту та функціоналу з керівником | 18.11.2023 |  |
| 6 | Розробка сайту та розміщення сайту на хостингу | 16.12.2023 |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 | Підготовка пояснювальної записки | 16.12.2023 |  |
| 13 | Здача курсової роботи на перевірку | 23.12.2023 |  |
| 14 | Захист | 29.12.2023 |  |

|  |  |  |
| --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) | Моспанюк Роман (прізвище на ініціали) |
| Керівник роботи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) | Анастасія Вознюк |

**АНОТАЦІЯ**

Моспанюк Р.І. Розробка веб-сайту про старап. – Курсова робота з дисципліни «Web-дизайн та web-програмування» студента спеціальності 121 «Інженерія програмного забезпечення». – Луцький національний технічний університет. – Луцьк, 2023. – с.

Пояснювальна записка до курсової роботи містить: 25 стор., 4 рис., 10 джерел.

Мета курсової роботи полягає в розробці сайту «Figma land».

Пояснювальна записка до курсової роботи складається з двох розділів: аналіз предметної області, розробка веб-сайту . В розділі «Аналіз предметної області» проаналізовано програми для розробки веб-сайту, етапи розробки, веб-хостинги для розміщення продукту. У розділі «Розробка веб-сайту «Figma land»» описано етапи створення сайту.

Ключові слова: Front end, старап, HTML, JavaScript, Sass, Telegram.

ЗМІСТ

[ВСТУП 6](#_Toc155278641)

[РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 7](#_Toc155278642)

[1.1 Огляд та аналіз сучасних Front End технологій. Вибір технологій для розробки веб-сайту новин про технології 7](#_Toc155278643)

[1.2 Постановка завдання на курсову роботу 12](#_Toc155278644)

[РОЗДІЛ 2 РОЗРОБКА ВЕБ-САЙТУ ПРО «FIGMA LAND» 14](#_Toc155278645)

[2.1 Обґрунтування вибору технологій і засобів вирішення поставленого завдання 14](#_Toc155278646)

[2.2 Етапи розробки веб-сайту про стартап 16](#_Toc155278647)

[2.4 Практична реалізація веб-сайту «Figma land» 17](#_Toc155278648)

[ВИСНОВКИ 24](#_Toc155278649)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛА 25](#_Toc155278650)

[ДОДАТКИ 26](#_Toc155278651)

# **ВСТУП**

В сучасній цифровій ері веб-дизайн та web-програмування є ключовими елементами успішної взаємодії інтернет-користувачів з різноманітними сервісами. Розробка власного веб-сайту не тільки є засобом представлення інформації, але й можливістю виявити та реалізувати творчі та технічні здібності. У даній курсовій роботі розглядається процес створення веб-сайту новин про технології, який був реалізований за допомогою таких технологій, як HTML, SCSS, JavaScript та Figma.

Вибір правильного макету перед початком розробки веб-сайту є визначним етапом, що визначає подальший успіх проекту. Макет визначає структуру та зовнішній вигляд веб-сайту, віддзеркалюючи його цільову аудиторію та функціональні вимоги. Відповідно до макету визначається логіка розташування контенту, навігація та загальна інтерактивність.

Веб-сайт новин про технології розроблявся з урахуванням сучасних тенденцій веб-дизайну та забезпечення зручності користувача на різних пристроях. Адаптивний дизайн гарантує зручний доступ до інформації навіть на невеликих екранах. Сайт структурований логічно та інтуїтивно зрозуміло, кожен розділ відповідає конкретній темі.

Мета курсової роботи – створення веб-сайту новин про технології з допомогою сучасних технологій веб-розробки.

Для реалізації цієї роботи поставлено наступні завдання:

* обрати макет сайту у програмі Figma;
* обрати технологій що будуть використовуватися при розробці;
* створити html розмітку сайту та оформити його за допосогою scss стилів;
* реалізувати форму зворотнього зв’яку;
* адаптувати сайт під пізні пристрої
* протестувати сайт та виправити знайдені помилки.

# **РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

## 1.1 Огляд та аналіз сучасних Front End технологій. Вибір технологій для розробки веб-сайту новин про технології

В наш час сфера веб-розробки прогресує неперервно, відкриваючи перед розробниками безліч інструментів та технологій для створення високоякісних веб-додатків. Зокрема, фронтенд-розробникам необхідно майстерно володіти різноманітними технологіями для успішної імплементації стильних та функціональних веб-сайтів. Основою залишаються HTML, CSS та JavaScript, що визначають структуру, дизайн і функціонал веб-сторінок.

У даному розділі ми детально розглянемо ключові компоненти фронтенд-розробки, обгрунтовуючи вибір конкретних інструментів для розробки веб-сайту новин про технології. HTML, що означає "Мова гіпертекстової розмітки", використовується для створення структури та відображення документів веб-сторінок. Перехід на HTML5 став актуальним у зв'язку зі зростанням його популярності та підтримки у сучасних браузерах. Крім того, розглянута варіація XHTML застосовує ті ж принципи, проте із більш жорсткими правилами синтаксису.

Стандарти розвитку цих технологій визначає консорціум всесвітньої веб-сторінки (W3C), сприяючи сумісності та стандартизації. Це дозволяє розробникам створювати веб-додатки, що працюють у різних браузерах і на різних пристроях [1].

CSS буквально означає "каскадні таблиці стилів" і являє собою мову програмування, призначену для створення стилістичного синтаксису для створення веб-сайтів і веб-додатків.

HTML сама по собі визначає такі речі, як структура. Наприклад, HTML може визначати розміщення панелі навігації та варіанти, які користувач може вибрати на панелі навігації.

На противагу цьому, CSS визначає стилістичні властивості веб-сайту, такі як кольори, макети сторінок і шрифти. Таким чином, HTML забезпечує функціональну основу, а CSS - дизайн і естетичні елементи. Однак CSS взаємодіє з HTML, щоб виконувати свою функцію додавання елементів стилю.

CSS є галузевим стандартом для стилізації HTML-сайтів. Він залишається незмінним протягом багатьох років. Однак деякі розробники все ще можуть покладатися виключно на HTML для стилізації своїх веб-сайтів. Це набагато складніший підхід, оскільки ви повинні вручну модулювати веб-сайт за допомогою HTML, тоді як CSS робить це за вас.

CSS також використовується для оптимізації сторінок для адаптивного веб-дизайну і для створення інших просунутих візуальних ефектів, таких як ефекти наведення.

Адаптивні веб-сторінки змінюються в макеті залежно від розміру екрану пристрою користувача. Останнім часом можливість створювати адаптивні сайти набуває все більшого значення, оскільки кількість потенційних варіантів розмірів екрану для користувачів стрімко зростає. Фактично, за останні кілька років мобільний трафік зріс настільки, що зараз більшість інтернет-трафіку припадає на мобільні пристрої [2,3].

Sass (скорочення від syntactically awesome style sheets - синтаксично дивовижні таблиці стилів) - це мова сценаріїв препроцесора, яка інтерпретується або компілюється в каскадні таблиці стилів (CSS). SassScript - це власне мова сценаріїв.

Sass складається з двох синтаксисів. Оригінальний синтаксис, який називається "синтаксис з відступами", використовує синтаксис, схожий на Haml. Він використовує відступи для розділення блоків коду і символи нового рядка для розділення правил. Новіший синтаксис, SCSS (Sassy CSS), використовує форматування блоків, подібне до CSS. Він використовує фігурні дужки для позначення блоків коду і крапку з комою для розділення правил всередині блоку. Файли синтаксису з відступами і SCSS традиційно мають розширення .sass і .scss відповідно.

CSS3 складається з серії селекторів і псевдоселекторів, які групують правила, що застосовуються до них. Sass (у ширшому контексті обох синтаксисів) розширює CSS, надаючи кілька механізмів, доступних у більш традиційних мовах програмування, зокрема об'єктно-орієнтованих, але недоступних у самому CSS3. Коли SassScript інтерпретується, він створює блоки правил CSS для різних селекторів, визначених у файлі Sass. Інтерпретатор Sass перекладає SassScript в CSS. Крім того, Sass може стежити за файлом .sass або .scss і перекладати його у вихідний файл .css щоразу, коли файл .sass або .scss зберігається.

Відступний синтаксис - це метамова. SCSS є вкладеною метамовою і надмножиною CSS, оскільки коректний CSS є коректним SCSS з тією ж семантикою.

SassScript надає наступні механізми: змінні, вкладеність, міксини та успадкування селекторів [4].

JavaScript (JS) - це легка інтерпретована (або скомпільована просто в часі) мова програмування з першокласними функціями. Хоча вона найбільш відома як мова сценаріїв для веб-сторінок, багато небраузерних середовищ також використовують її, наприклад, Node.js, Apache CouchDB та Adobe Acrobat. JavaScript - це багатопарадигмальна, однопотокова, динамічна мова, що базується на прототипах, підтримує об'єктно-орієнтовані, імперативні та декларативні (наприклад, функціональне програмування) стилі.

Динамічні можливості JavaScript включають конструювання об'єктів під час виконання, списки змінних параметрів, змінні функцій, динамічне створення сценаріїв (за допомогою eval), самоаналіз об'єктів (за допомогою утиліт for...in та Object) та відновлення вихідного коду (функції JavaScript зберігають свій вихідний текст і можуть бути відновлені за допомогою toString()).

Стандартами для JavaScript є специфікація мови ECMAScript (ECMA-262) і специфікація API інтернаціоналізації ECMAScript (ECMA-402). Як тільки один з браузерів реалізує якусь функцію, ми намагаємося її задокументувати. Це означає, що у випадках, коли деякі пропозиції щодо нових можливостей ECMAScript вже реалізовані в браузерах, документація і приклади в статтях MDN можуть використовувати деякі з цих нових можливостей. Здебільшого це відбувається між етапами 3 і 4, і, як правило, до офіційної публікації специфікації [5].

Swiper – це потужний та гнучкий інструмент для створення сучасних і мобільно-дружніх слайдерів та каруселей на веб-сайтах, що базується на JavaScript. Він виділяється своєю простотою використання та високою швидкістю, що дозволяє легко розробляти інтерактивні та адаптивні елементи веб-інтерфейсу, такі як слайдшоу, галереї зображень та каруселі для прокрутки контенту.

Основні особливості Swiper включають респонсивний дизайн, сенсорне управління, вбудовану підтримку клавіатурного керування та різноманітні опції налаштувань. Крім того, бібліотека надає готові теми оформлення, що полегшує її інтеграцію в різноманітні дизайнерські концепції.

Swiper широко використовується у веб-розробці для створення динамічних та візуально привабливих інтерфейсів, де надання користувачам ефективного способу взаємодії з вмістом має важливе значення. Незалежно від складності проекту, Swiper забезпечує ефективну реалізацію слайдерів та каруселей, зроблюючи веб-сайти більш привабливими та динамічними.

Сучасні фахівці в області Front-end розробки все частіше використовують різноманітні фреймворки та бібліотеки, такі як React, Vue та Angular, для створення складних компонентів та ефективного управління станом веб-сайтів. Ці інструменти спрощують розробку та підтримку проектів, дозволяючи швидко та ефективно створювати високоякісні веб-додатки [6].

Штучний інтелект (ШІ) – це передова технологія у сфері веб-розробки, здатна ефективно обробляти обширні обсяги даних, приймати обґрунтовані рішення та робити точні прогнози. Із завдань, які раніше були виключною компетенцією людських програмістів, штучний інтелект вносить революцію у сферу створення, оптимізації та завершення коду.

Однією з ключових переваг є здатність ШІ генерувати код на основі природної мови та відповідати на підказки програміста. Наприклад, система Codex від OpenAI може автоматично створювати фрагменти коду, але, звісно, необхідно провести ручну перевірку результатів. Також ШІ може аналізувати існуючий код та пропонувати оптимізації для підвищення його швидкості та ефективності.

Сучасний ШІ широко використовується для забезпечення якості та тестування веб-розробок, а також у розробці та дизайні веб-сайтів. Цей варіант використання ШІ, хоча вже досить популярний в автоматизованому тестуванні, набуває нового розмаху завдяки зростанню технічної потужності. Тепер інструменти ШІ можуть використовуватися для аналізу даних та прогнозування можливих проблем, що дозволяє вирішувати ризики та нюанси на етапі їхнього виникнення, запобігаючи серйозним проблемам у процесі розгортання веб-проектів [7].

Figma – це інструмент для колективного дизайну веб-інтерфейсів, який дозволяє користувачам спільно працювати над створенням живих та інтерактивних прототипів. Відзначившись в 2016 році, Figma швидко завоювала популярність серед фахівців у сфері веб-дизайну та спільнот онлайн. Завдяки Figma користувачі можуть ефективно взаємодіяти та обмінюватися шаблонами, дизайнами та віджетами з мільйонами користувачів у всьому світі.

З використанням інструментів векторної графіки команди дизайнерів можуть легко створювати складні каркаси для веб-сайтів, які адаптуються для різних екранів. Крім того, користувачі можуть додавати інтерактивні елементи, такі як прокрутка та наведення курсору миші, для створення сучасних веб-сторінок.

Figma дозволяє користувачам створювати робочі прототипи веб-сторінок, які легко тестувати, дозволяючи отримати швидкий зворотній зв'язок. При потребі дизайнери можуть експортувати певну кількість коду з прототипів для передачі розробникам.

Однією з ключових переваг Figma є можливість працювати над проектами з будь-якого місця та пристрою без обов'язкового встановлення додатка. Програма доступна на веб-сайті Figma, вимагаючи лише реєстрації та входу в систему.

Маючи різні варіанти ціноутворення, включаючи безкоштовний план з обмеженими можливостями, Figma надає користувачам гнучкість у виборі підписки, що відповідає їхнім конкретним потребам. З розвитком Figma як потужного інструменту для співпраці, спільнота дизайнерів і розробників постійно поповнюється новими плагінами, шаблонами та віджетами для загального використання [7,8].

Враховуючи вищезгадані технології та інструменти, вибір для розробки веб-сайту новин про технології. у цій курсовій роботі буде на користь HTML, CSS з використанням препроцесора Sass, та JavaScript для створення самого сайту та Figma­ – для зручної роботи з макетом для сайту. За допомогою цих інструментів вдалося створити веб-сайт високої якості з високою функціональністю, що відповідає найсучаснішим стандартам та вимогам користувачів.

## 1.2 Постановка завдання на курсову роботу

Враховуючи обрані технології та загальні цілі розробки веб-сайту новин про технології, було поставлено конкретні завдання, які дозволять досягти успішного виконання курсової роботи. Зазначені цілі відображають основні напрямки та вимоги проекту:

* вибір макету для сайту у програмі Figma – Обрання макету сайту у програмі Figma є важливим етапом у веб-розробці, оскільки це визначає основну структуру та зовнішній вигляд веб-сайту. Перед тим, як розпочати написання коду, було проведено детальне планування та вибір оптимального макету веб-сайту новин про технології;
* створення html розмітки сайту та створення scss стилів – ці кроки представляють собою ключові етапи веб-розробки, які взаємодіють між собою для формування структури та зовнішнього вигляду веб-сторінок;
* написання адаптиву для мобільних пристроїв – створення веб-сайту таким чином, щоб він ефективно та естетично виглядав на різних розмірах екранів, зокрема на мобільних пристроях. Цей підхід забезпечує оптимальний користувацький досвід незалежно від того, чи відбувається перегляд веб-сайту на смартфоні, планшеті чи настільному комп'ютері;
* реалізація форми зворотнього зв’яку – створення веб-форми, яка дозволяє користувачам надсилати повідомлення чи коментарі, прямо адміністратору;
* тестування та виправлення помилок – проведення систематичних тестів для виявлення та виправлення будь-яких помилок та недоліків у функціоналі та дизайні інтернет-магазину.

Це ключові завдання для успішної розробки веб-сайту новин про технології. Вони є найважливішими аспектами у процесі розробки даного веб-сайту.

# **РОЗДІЛ 2 РОЗРОБКА ВЕБ-САЙТУ ПРО «FIGMA LAND»**

## 2.1 Обґрунтування вибору технологій і засобів вирішення поставленого завдання

Під час створення веб-сайту новин про технології, вибір відповідного середовища розробки визначається як критично важливий етап для досягнення ефективності та продуктивності. На цьому етапі було вирішено скористатися інтегрованим редактором коду Visual Studio Code, оскільки він пропонує ряд переваг, які суттєво спрощують та оптимізують процес розробки.

Visual Studio Code, розроблене Microsoft, є інтегрованим середовищем розробки та творчим майданчиком, що дозволяє розробникам ефективно створювати, редагувати та налагоджувати код перед подальшим його публікацією. Цей текстовий редактор має безліч настроюваних функцій у вигляді плагінів, які дозволяють створювати індивідуалізоване середовище розробки.

Visual Studio Code є легким та потужним, легко встановлюється на будь-яку платформу. Забезпечуючи вбудовану підтримку для Node.js, TypeScript, та JavaScript, а також розширеної екосистеми для інших мов програмування, таких як C++, Java, C#, PHP, Go та Python, він виходить за рамки звичайного текстового редактора.

Visual Studio Code входить в той же ряд із відомими інструментами, такими як Atom, Sublime та Text Wrangler, проте відрізняється своїми вищими можливостями та надійністю. Дозволяючи розробникам писати, редагувати та налагоджувати код в одному місці, VS Code підтримує багато мов програмування та пропонує вбудовану багатомовну підтримку.

Розробники можуть сподіватися на Visual Studio Code як на надійний інструмент для всіх аспектів розробки, особливо якщо використовувати його у поєднанні з відповідними інструментами. Завдяки вбудованій підтримці JavaScript, Node.js та TypeScript, а також можливості додавати підтримку інших мов, таких як Python, C#, PHP, Java та інших, Visual Studio Code є невід'ємною частиною процесу розробки, забезпечуючи неймовірну зручність та функціональність на всіх платформах [10].

У наш час цифрових технологій забезпечення ефективного та зручного механізму отримання зворотного зв'язку на веб-сайті визначається як один з ключових аспектів взаємодії з користувачами. Цікавою альтернативою традиційній електронній пошті є використання месенджера Telegram для обробки повідомлень від відвідувачів.

Telegram, як месенджер, володіє надзвичайно швидким та миттєвим способом передачі повідомлень. На відміну від традиційної електронної пошти, де інтервали часу можуть бути великими, використання Telegram дозволяє негайно взаємодіяти. Користувачі можуть отримати відповіді та відправляти повідомлення без зайвої затримки, що особливо важливо в ситуаціях, де час важливий фактор.

Однією з ключових переваг Telegram є його мобільність та універсальність. Мобільні додатки Telegram доступні для різних операційних систем, таких як iOS та Android, забезпечуючи зручність для користувачів різних пристроїв. Завдяки месенджеру, відвідувачі можуть взаємодіяти з веб-сайтом з будь-якого місця та пристрою, забезпечуючи гнучкість та доступність комунікації.

Також Telegram дозволяє взаємодіяти не тільки текстовими повідомленнями, але й різноманітним мультимедійним контентом. Можливість надсилати зображення, відео, аудіозаписи та інші формати робить комунікацію більш інформативною. Це сприяє кращому розумінню питань та дозволяє детальніше вирішувати проблеми, що виникають. Загальна взаємодія з користувачами через Telegram може бути більш ефективною завдяки високому рівню візуальної комунікації.

Для комфортної роботи з макетом проекту використовувалася Figma – інструмент для розробки дизайну та створення макетів веб-сайтів.

Figma відзначається високою ефективністю порівняно з іншими інструментами, такими як Avocode та Photoshop, завдяки великій кількості розширень та введенню нового режиму "dev-mode". Плагіни полегшують інтеграцію зовнішніх сервісів та автоматизують певні процеси, зробивши роботу більш швидкою та ефективною. Режим "dev-mode" спрощує взаємодію з розробниками, дозволяючи перевіряти та тестувати інтерфейс безпосередньо в середовищі розробки. Це значно прискорює процес внесення змін та узгодження макету з кодом.

Прекрасним інструментом для розробки є ChatGPT – мовна модель, розроблена компанією OpenAI, яка може генерувати текст, враховуючи контекст та конкретне завдання.

В рамках даної курсової роботи ChatGPT використовувався для реалізації функціоналу, пов'язаного з відправкою електронного листа замовлення. Це дозволило автоматизувати процес створення автоматичних повідомлень, спрямованих на поліпшення взаємодії з користувачами та оптимізацію обробки замовлень.

Однією з ключових областей впливу ChatGPT у веб-розробці є можливість покращення взаємодії з користувачем. Веб-сайти та додатки тепер можуть надавати більш природні та інтуїтивно зрозумілі консультації чи відповіді на запитання, спрощуючи користування та підвищуючи задоволення від використання.

ChatGPT може використовуватися для автоматичного створення контенту для веб-сайтів, включаючи статті, огляди та інші текстові матеріали. Це дозволяє заощаджувати час на написання та оптимізувати SEO-показники.

Інтелектуальні алгоритми ChatGPT можуть бути використані для аналізу поведінки користувачів та прогнозування їхніх потреб, що дозволяє персоналізувати веб-сайти, рекомендувати контент або товари, відповідати на індивідуальні запитання, забезпечуючи більш особистий досвід взаємодії. Застосування ChatGPT у веб-розробці відкриває широкі можливості для покращення користувацького досвіду, оптимізації роботи веб-сайтів та використання штучного інтелекту в різних сферах веб-індустрії.

## 2.2 Етапи розробки веб-сайту про стартап

Етапи розробки веб-сайту новин про технології включають деталізовані кроки, що описують весь процес створення веб-додатку, починаючи від вибору дизайну та закінчуючи розгортанням готового проекту на хостингу.

1. Вибір макету:

* вибір макету у Figma веб-сайту новин про технології;
* створення структури, дизайну та елементів функціоналу.

1. Розробка базової верстки:

* створення основної структури сторінок за допомогою HTML;
* використання CSS для надання стилізації та естетичного вигляду;
* використання JavaScript для впровадження інтерактивності та реалізації потрібного функціоналу.

1. Тестування та виправка помилок:

* виконання тестування для перевірки функціональності та коректності роботи сайту;
* виправлення помилок та переписування тих частин коду що конфліктують одна з одною, або не працюють так як задумувалося.

1. Розгортання на хостингу:

* вибір середовища для розгортання сайту;
* розміщення сайту.

## 2.4 Практична реалізація веб-сайту «Figma land»

Секція герой на сайті є ключовим елементом веб-дизайну, який захоплює увагу відвідувачів та створює перший враження про бренд чи продукт.

Зазвичай розташована у верхній частині головної сторінки, ця секція повинна вражати естетикою та візуальною привабливістю. Вона часто містить велике, високоякісне зображення чи відео, що ілюструє сутність бренду або передає ключовий повідомлення. Текстовий блок містить короткий та чіткий заголовок, який привертає увагу, разом із коротким описом чи запрошенням до дії. Щоб забезпечити ефективність, секція герой добре скоординована за кольорами та шрифтами, враховуючи відмінності відображення на різних пристроях. Ця секція створює враження, спонукає користувачів до дослідження сайту та визначає тон для подальшого взаємодії з контентом.

На сайті реалізовані декілька слайдерів: галерея та відгуки Один з найпопулярніших видів слайдерів - галерея, яка дозволяє стартапам чітко продемонструвати свої ключові аспекти та переваги. Наприклад, перший слайдер на сайті стартапу фокусується на елементах, які є основними в їхній роботі або продукті. Це може включати ключові функції продукту, важливі моменти його розробки або будь-які особливості, які роблять цей стартап унікальним та цікавим для потенційних клієнтів.

Другий слайдер, що відображає відгуки клієнтів, має велике значення для підтримки авторитетності та відображення позитивного досвіду користувачів. Це дозволяє відвідувачам сайту отримати безпосередній вигляд на те, як інші люди вже скористалися послугами чи продуктами цього стартапу. Такий підхід створює додатковий рівень довіри до компанії, адже відгуки реальних користувачів мають значно більший вплив на прийняття рішення про співпрацю чи покупку.

Використання слайдера для новин робить веб-сайт динамічним, цікавим та легким у сприйнятті. Він стає не лише інформаційним ресурсом, але і медійним елементом, який привертає увагу та залишає позитивне враження від відвідування.

Слайдер, реалізований за допомогою JS бібліотеки "Swiper", становить важливий аспект для поліпшення інтеракції з контентом на веб-сайті, дозволяючи елегантно презентувати змінюючийся матеріал без зайвого коду. Використання технології Swiper гарантує чудову адаптивність та гнучкість, забезпечуючи плавні переходи між слайдами та покращуючи користувацький досвід.(див. ліст. 2.1)(див. рис. 2.1)

Лістинг 2.1 – реалізація слайдерів за допомогою SwiperJs

var swiper\_testimon = new Swiper(".testimonial\_\_swiper", {

slidesPerView: 'auto',

spaceBetween: 100,

centeredSlides:true,

loop:true,

pagination: {

el: ".testimonial\_\_pagination",

clickable: true,

},

});

var swiper\_gal= new Swiper(".gallery\_\_swpiper", {

slidesPerView: 'auto',

spaceBetween: 100,

centeredSlides:true,

loop:true,

pagination: {

el: ".gallery\_\_pagination",

clickable: true,

},

});

Лістинг – 2.1 swiper.

Було розроблено модальне вікно для форми зворотнього зв’язку яка отримує електронну пошту користувача та передає її з допомогою телеграм бота адміністратору для подальших дій.(див. ліст 2.2)

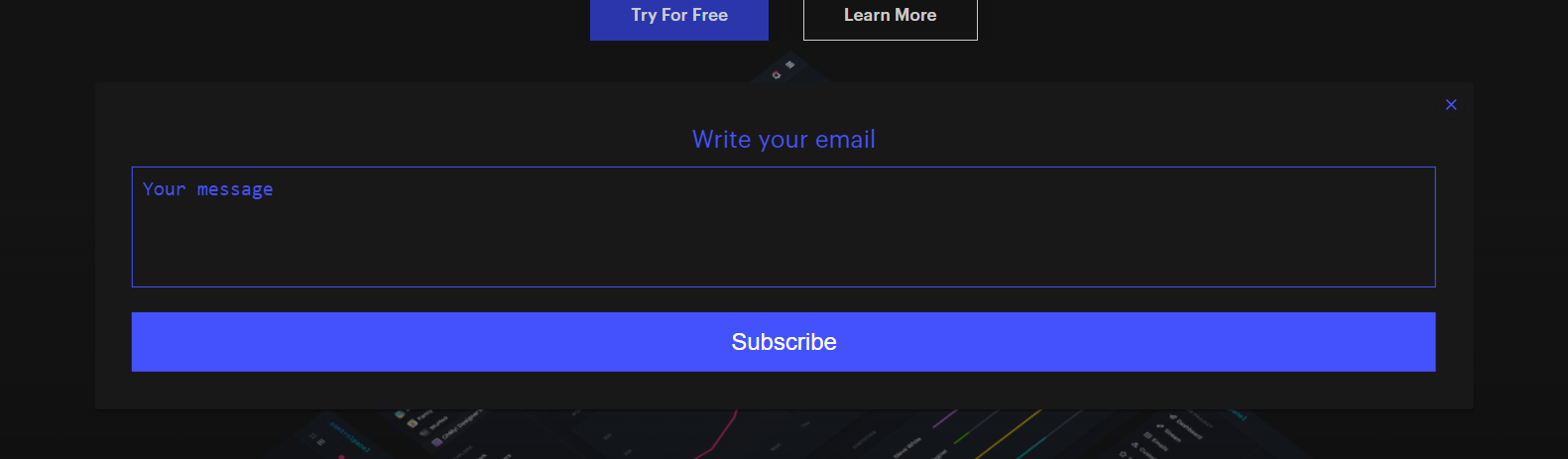


Рисунок 2.2 – Вигляд модального вікна

Для програмної реалізації модального вікна було створено функції що вимикають та вмикають можливість прокручувати сторінку. (див. ліст. 2.2)

Лістинг 2.2 – реалізація модального вікна з відправкою в телеграм

(() => {

const openModalBtns = document.querySelectorAll("[data-first-modal-open]");

const closeModalBtns = document.querySelectorAll("[data-modal-close]");

const modals = document.querySelectorAll("[data-modal]");

const body = document.body;

openModalBtns.forEach((btn, index) => {

btn.addEventListener("click", () => {

closeModal(index);

});

});

closeModalBtns.forEach((btn, index) => {

btn.addEventListener("click", () => {

closeModal(index);

});

});

function closeModal(index) {

modals[index].classList.toggle("is-hidden");

body.classList.toggle("no-scroll");

}

const forms = document.querySelectorAll("#subscribe");

forms.forEach((form) => {

form.addEventListener("submit", (e) => {

e.preventDefault();

var message = form.querySelector("#message").value;

var subscribe = `New message recieved :${message}`;

var token = "6270391808:AAEeDST2kRZacIxzisJz-GBovp8C8uRHcY0";

var chat\_id = "308432208";

var url = `https://api.telegram.org/bot${token}/sendMessage?chat\_id=${chat\_id}&text=${subscribe}&parse\_mode=html`;

let api = new XMLHttpRequest();

api.open("GET", url, true);

api.send();

console.log("Subscribe successfully sent!");

form.reset();

closeModal();

});

});

})();

Лістинг – 2.2 modal window.

Для розробки мобільного меню я використовував комбінацію HTML, CSS та JavaScript. У HTML я створив базову структуру, включаючи саме бургер-меню та його складові елементи. CSS допоміг мені стилізувати вигляд меню, забезпечити адаптивність до різних розмірів екранів та додати анімаційні ефекти для плавного розгортання та згортання.(див. рис. 2.3-2.4)



Рисунок 2.3-2.4 – Вигляд хедера на мобільному екрані

JavaScript виявився корисним для створення взаємодії між користувачем та меню. Він дозволив виконати функціонал, коли користувач натискає на бургер-іконку, щоб розгорнути чи згорнути меню, і відповідно реагувати на події та анімацію. Такий підхід дозволив створити зручне та естетичне мобільне меню для сайту, що допомагає полегшити навігацію на пристроях з невеликими екранами.(див. ліст. 2.2)

Лістинг 2.2 – реалізація модального вікна з відправкою в телеграм

const menu = document.querySelector(".menu");

const body = document.body;

const menuItems = document.querySelectorAll(".menuItem");

const hamburger = document.querySelector(".hamburger");

const closeIcon = document.querySelector(".closeIcon");

const menuIcon = document.querySelector(".menuIcon");

function toggleMenu() {

if (menu.classList.contains("showMenu")) {

body.classList.remove("no-scroll");

menu.classList.remove("showMenu");

closeIcon.style.display = "none";

menuIcon.style.display = "block";

} else {

menu.classList.add("showMenu");

body.classList.add("no-scroll");

closeIcon.style.display = "block";

closeIcon.style.position = "fixed";

closeIcon.style.right = "0";

menuIcon.style.display = "none";

}

}

hamburger.addEventListener("click", toggleMenu);

menuItems.forEach(function (menuItem) {

menuItem.addEventListener("click", toggleMenu);

});

Лістинг – 2.2 mobile menu.

2.4 Тестування веб-сайту про старап

Веб-розробка – це динамічна галузь, яка вимагає ретельного тестування, щоб забезпечити відповідність веб-сайтів найвищим стандартам функціональності, продуктивності та користувацького досвіду. Процес тестування складається з низки кроків, спрямованих на перевірку кожного аспекту дизайну та функціональності веб-сайту.

Починаючи з ретельного аналізу вимог і цілей проекту, етап тестування включає в себе стратегічне планування для визначення обсягу, ресурсів і термінів. Потім акцент переноситься на функціональне тестування, де кожна функція проходить ретельну перевірку, щоб підтвердити належну навігацію, заповнення форм і обробку даних.

Тестування на сумісність забезпечує безперебійну роботу веб-сайту в різних браузерах, операційних системах і на різних пристроях. Водночас юзабіліті-тестування оцінює користувацький інтерфейс і загальний досвід, щоб гарантувати дружній дизайн.

Тестування продуктивності стає вирішальним для оцінки швидкості, швидкості реагування та масштабованості, особливо в умовах різного навантаження трафіку. Тестування безпеки виявляє та виправляє потенційні вразливості, захищаючи від таких загроз, як SQL-ін'єкції та міжсайтовий скриптинг.

Регресійне тестування проводиться ітеративно, перевіряючи, що нові оновлення або виправлення помилок не ставлять під загрозу існуючі функціональні можливості. Автоматизоване тестування інтегрується для оптимізації повторюваних завдань і забезпечення послідовних і швидких результатів.

Тестування прийнятності для користувачів включає в себе реальні сценарії, що дозволяє кінцевим користувачам протестувати веб-сайт і надати цінний зворотній зв'язок.

Постійний моніторинг і регулярне обслуговування допомагають вирішувати проблеми, що виникають, і вдосконалювати загальну функціональність. Дотримуючись такого комплексного підходу до тестування, веб-розробники гарантують створення надійних, безпечних веб-сайтів, які забезпечують безперебійну роботу користувачів.

## ВИСНОВКИ

Під час виконання курсової роботи відбулося ретельне дослідження передових технологій веб-розробки з метою їх використання для створення інноваційного веб-сайту про технології.

Ключовим завданням було створення макету веб-сайту у програмі Figma. Цей етап дозволив ретельно проробити структуру, дизайн та функціонал майбутнього веб-сайту.

Вибір технологій, таких як HTML, SCSS та JavaScript, був обґрунтований з метою забезпечення якісного та ефективного розроблення. Використання препроцесора SCSS спростило роботу із стилями та полегшило підтримку та модифікацію коду. Реалізація форми зворотнього зв’язку та її тестування дозволили покращити взаємодію з користувачами та забезпечити високу якість веб-додатка.

Важливим етапом виявилось використання штучного інтелекту для автоматизації відправки повідомлень та поліпшення взаємодії з користувачами. Це відкрило нові можливості для впровадження інтерактивних елементів та підвищення зручності використання сайту.

Також був розроблений адаптивний дизайн для веб-сайту з метою поліпшення користувацького досвіду. Цей підхід є критичним для успішної веб-розробки, оскільки враховує різноманіття сучасних пристроїв та різні потреби користувачів.

Комплексне тестування включало в себе аналіз вимог, тестування продуктивності, регресійне тестування, тестування прийнятності, сумісність з різними платформами, продуктивності та безпеки.

Курсова робота висвітлила не лише важливі аспекти веб-розробки, але й показала, як використання передових технологій може успішно впроваджуватися у реалізацію веб-проектів, таких як веб-сайт новин про технології. У результаті виконання проекту були досягнуті всі поставлені завдання, і успішно реалізовано веб-сайт новин про технології, який інтегрується із стилістикою гри, надаючи веб-сайту привабливий зовнішній вигляд.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛА**

1. List of frontend technologies every developer should know in 2023. URL: https://merge.rocks/blog/list-of-frontend-technologies-every-developer-should-know-in-2023 (дата звернення: 21.12.2023).

2. What Is CSS? (And Why It’s Important) URL: https://www.indeed.com/career-advice/career-development/what-is-css (дата звернення: 21.12.2023).

What is CSS? A Beginner's Guide URL: https://careerfoundry.com/en/blog/web-development/what-is-css/ (дата звернення: 21.12.2023).

3. Top Front End Technologies You Must Know [2023] URL: https://www.interviewbit.com/blog/front-end-technologies/

(дата звернення: 21.12.2023).

4. Sass (style sheet language) URL: https://en.wikipedia.org/wiki/Sass\_(style\_sheet\_language) (дата звернення: 21.12.2023).

5. JavaScript URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript

6. Building modern sliders with HTML, CSS, and Swiper. URL: https://blog.logrocket.com/building-modern-sliders-html-css-swiper/,

(дата звернення: 21.12.2023).

7. Pros and Cons of Using AI for Web Development: Stumbling Blocks You Have to Know About. URL: ttps://litslink.com/blog/using-ai-for-web-development

8. What is Figma? URL: https://www.nobledesktop.com/learn/figma/what-is-figma (дата звернення: 21.12.2023).

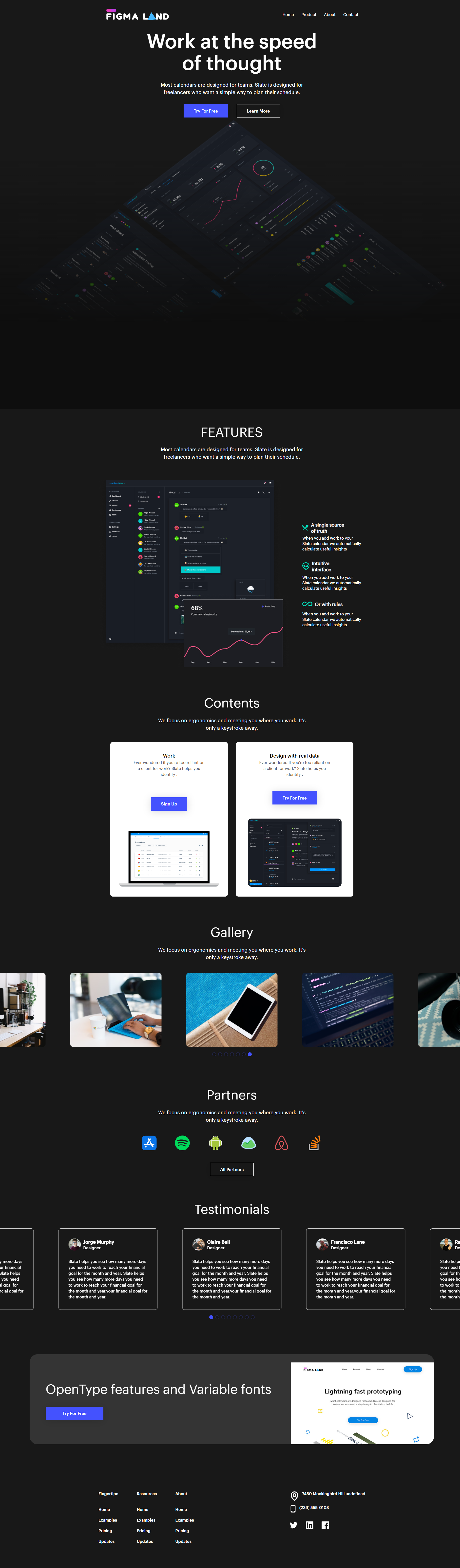
9. The Power of Figma as a Design Tool URL: https://www.toptal.com/designers/ui/figma-design-tool(дата звернення: 21.12.2023).

10. Visual Studio Code URL: https://code.visualstudio.com/docs/editor/whyvscode (дата звернення: 21.12.2023).

# ДОДАТКИ

**Додаток А**

**Головна сторінка сайту «Figma land»**



**Додаток Б**

**Результат перевірки на плагіат на сайті my.plag.com.ua**

