

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-05

Дипломний проект

здобувача освіти денної форми навчання
КГ.05.05.000.ДП

**ГРАБОВОГО
ОЛЕКСАНДРА
ВАЛЕНТИНОВИЧА**

м. Одеса
2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **123 «Комп'ютерна інженерія»**

Освітня програма: **«Комп'ютерна графіка і Web-дизайн»**

Група: **4КГ-05**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка веб-сайту курсів іноземних мов

Проектний матеріал складається з пояснювальної записки на 60 сторінках та графічного (презентаційного) матеріалу на 10 аркушах (слайдах).

Дипломник _____ (Грабовий О.В.)

Керівник _____ (Краснієнко Н.В.)

Консультанти:

з економічної частини _____ (Копайгородська Т.Г.)

з охорони праці _____ (Чорновол Н.І.)

з дотримання вимог ЄСКД _____ (Петрашова В.І.)

старший консультант _____ (Скорнякова О.В.)

До захисту допущений

Голова циклової комісії _____ (Скорнякова О.В.)

Завідувач відділення _____ (Суліма Ю.Ю.)

Захист « » _____ 2022 р. Протокол ДКК № _____

Оцінка ДКК _____

Секретар ДКК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та Ш
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна графіка і Web-дизайн»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І.В.

“ _____ ” _____ 2022 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Грабовому Олександр Валентиновичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка веб-сайту курсів іноземних мов

затверджена наказом по коледжу від “ 13 ” 12 2021 р. № 306-А2-ОД

2. Термін здачі закінченого проекту (роботи) 21.06.21

3. Вихідні данні до проекту (роботи) Обґрунтування технології програмування.
Розробка макету головної сторінки у HTML-редакторі. Визначення
структури та формування контенту веб-сайту. Розробка програмного
забезпечення та верстка веб-сайту

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Вступ.

2. Конструкторський розділ. Розробка структури веб-сайту. Обґрунтування
вибору програмного забезпечення. Розробка програмного зазпечення та
верстка веб-сайту

3. Економічний розрахунок.

4. Охорона праці. Висновки. Перелік використаних джерел

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Презентація. 10 слайдів

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1-2	Краснієнко Н.В.	16.05.22	
3	Копайгородська Т.Г.	16.05.22	
4	Чорновол Н.І	16.05.22	
ЄСКД	Петрашова В.І.	16.05.22	

7. Дата видачі завдання _____

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Робота над вступом.	16.05.22	Виконано
2	Аналіз теми ДП та огляд літературних джерел	22.05.22	Виконано
3	Розробка конструкторського розділу ДП	23.05.06	Виконано
4	Обґрунтування вибору ПЗ	24.05.22	Виконано
3	Обґрунтування структури веб-сайту	25.05.22	Виконано
4	Розробка функціональної схеми веб-сайту.	28.05.22	Виконано
5	Розробка ПЗ	05.06.22	Виконано
6	Виконання розділу «Економічний розрахунок»	06.06.22	Виконано
7	Розрахунок економічних показників проекту	06.07.22	Виконано
8	Виконання розділу «Охорона праці»	08.06.22	Виконано
9	Перевірка якості виконання розділу «Охорона праці»	12.06.21	Виконано
10	Виконання пояснювальної записки ДП	13.06.22	Виконано
11	Перевірка якості виконання пояснювальної записки ДП	14.06.22	Виконано
12	Виконання графічної частини ДП	15.06.22	Виконано
13	Перевірка якості виконання графічної частини ПЗ	16.06.22	Виконано
14	Малий захист	17.06.22	Виконано

Дипломник

(підпис)

Керівник

(підпис)

ЗМІСТ

ВСТУП	5
1. ТЕХНОЛОГІЧНИЙ РОЗДІЛ	7
1.1 Аналіз технічного завдання	7
1.2 Вибір основних технологій для розробки	7
1.3 Вибір додаткових технологій для розробки	12
1.4 Висновки	16
2. ОСНОВНА ЧАСТИНА	17
2.1 Етапи розробки веб-сайту	17
2.2 Первісна підготовка та налаштування усього проекту	19
2.3 Первісна підготовка та налаштування сторінки	25
2.4 Додавання модуля або його створення	26
2.5 Висновки	41
3. ЕКОНОМІЧНИЙ РОЗРАХУНОК	45
3.1 Витрати на розробку сайту	46
3.2 Витрати на впровадження та експлуатацію сайту	49
3.3 Розрахунок економічної ефективності	50
3.4 Функціональна та соціальна ефективність	51
3.5 Висновки	51
4. ОХОРОНА ПРАЦІ	52
4.1 Розробка заходів з охорони праці	53
4.2 Пожежна безпека	56
4.3 Висновки	57
ВИСНОВОК	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТОК	60

					КГ05.05.000.00 ДП ПЗ	Арк.
						4
Ізм.	Лист	№ докум.	Підпис	Дата		

ВСТУП

З розвитком і поширенням мережі Інтернет все більше підприємств, виробництв, учбових та інших закладів використовують її для поширення інформації. Це не дивно, адже на даний момент кількість смартфонів перевищила кількість людей на нашій планеті, кількість користувачів цієї мережі становить близько 4,2 млрд., що становить приблизно 52% населення нашої планети. В середньому, людина витрачає 7 годин на день, користуючись мережею Інтернет. Якщо говорити фінансовою статистикою, то за даними за 2020 рік, ринок електронної комерції в Україні досягає 4 млрд. доларів, за останні 5 років він виріс у 3 рази. Отже, переведення існуючих підприємств до цієї мережі є і буде актуальним, як і буде актуальним створення нових, які з самого початку будуть інтернет-орієнтованими.

За допомогою інтернет-орієнтованих систем, що являють собою веб-сайти, можливо розповсюдити будь-яку інформацію, докладаючи мінімум зусиль. А головне, що сучасні пошукові системи, донесуть цю інформацію заінтересованим в ній людям (цільовій аудиторії), звісно, якщо веб-сайт був правильно налаштований. Це новий інструмент для ведення бізнесу та більш повного задоволення потреб, дієвий засіб зниження витрат. В свою чергу, для споживачів це нові можливості взаємодії та новий ефективний засіб комунікації з підприємствами та організаціями, а також між собою. Завдяки розвитку мережі Інтернет та пов'язаних з нею технологій, для користувачів з'явилося нове інформаційне джерело про товари і послуги. Це нова можливість для задоволення своїх потреб, що зумовлює актуальність теми дипломного проектування.

Сфера розробки веб-сайтів буквально за кілька років зазнала еволюційних змін і пройшла шлях від примітивних статичних сторінок до багатофункціональних систем. Це відбулося завдяки широкому розповсюдженні мови програмування JavaScript, а також її швидкому розвитку.

					КГ05.05.000.00 ДП ПЗ	Арк.
						5
Ізм.	Лист	№ докум.	Підпис	Дата		

За даними за 2020 рік, JavaScript є найпопулярнішою мовою програмування, серед розробників в Україні, і становить понад 18,5%. Також, вона є майже монополістом в сфері розробки веб-сайтів, на її основі розроблена велика кількість фреймворків, які також є дуже популярними (наприклад: Vue, React, Angular). На її основі навіть зроблені мови програмування, такі як TypeScript та CoffeeScript, а також NodeJs, який перетворює JavaScript з вузькоспеціалізованої мови на мову загального призначення.

Метою дипломного проекту є створення веб-сайту для курсів іноземних мов, використовуючи нові можливості мови програмування JavaScript, а також HTML, CSS та інші допоміжні технології.

					КГ05.05.000.00 ДП ПЗ	Арк.
						6
Ізм.	Лист	№ докум.	Підпис	Дата		

1. ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1. Аналіз технічного завдання

Згідно технічного завдання на дипломне проектування основним завданням веб-сайту для курсів іноземних мов є: донести до потенційного клієнта переваги цих курсів, перелік мов для вивчення, можливі варіанти навчання, їх ціну, фізичну адресу курсів та іншу контактну інформацію. Також реалізувати можливість замовити зворотній дзвінок, записатися на пробне заняття та сплатити онлайн (реалізувати лише клієнтську частину). Необхідно обов'язково реалізувати підтримку мобільних пристроїв, такі як смартфони та планшети.

1.2. Вибір основних технологій для розробки

Так як я реалізую веб-сайт, то для цього необхідно зробити розмітку, це можливо за допомогою мови розмітки гіпертексту - HTML.

Hyper Text Markup Language або **HTML** - це стандартизована мова гіпертекстової розмітки для документів, які створюються для показу у веб-браузері. Часто використовується у поєднанні з такими технологіями як Cascading Style Sheets (CSS) та скриптовими мовами програмування, наприклад JavaScript. Веб-браузери отримують HTML документи з веб-серверу або локального середовища, після чого рендерять (будують) документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки, її семантику, а також може включати опис зовнішнього вигляду (стили). HTML елементи є будівельними блоками HTML сторінок. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. У HTML є можливість створювати структуровані сторінки, це означає

					КГ05.05.001.00 ДП ПЗ	Арк.
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

можливість використовувати спеціальні структурні блоки, наприклад: header (верхня частина сайту), footer (нижня частина сайту), параграфи, цитати та інше. Елементи HTML окреслені тегами, написаними з використанням кутових дужок. Такі теги, як `` або `<input />` вставляють різноманітний контент прямо у сторінку. А теги на кшталт `<p>` відображають текстову інформацію та можуть містити у собі інші теги. Браузер не показує теги, він показує лише інформацію всередині них. HTML може містити в собі програми, написані на скриптових мовах програмування, таких як JavaScript, які мають вплив на поведження та контент веб-сторінки. Додання CSS впливає на зовнішній вигляд та шари контенту.

Розмітка. Розмітка HTML сторінки містить у собі різні теги (та їхні атрибути), базові типи даних, символічні мнемоніки та декларації типу документа. Частіш за все, HTML теги йдуть парами, наприклад `<h1></h1>`, але деякі теги не містять у собі інші, тому не мають пари (наприклад ``). У парах перший тег називається відкриваючим, а другий закриваючим. Також, важливим компонентом HTML є декларація типу документа (DOCTYPE), яка відповідає за режими рендерінгу сторінки.

Елементи. Документи HTML мають структуру вкладених елементів HTML. Вони вказуються в документі тегами, огорнутими в кутові дужки таким чином: `<p>`. У простому, загальному випадку, елемент вказується парою тегів: «початковий тег» `<p>` і «кінцевий тег» `</p>`. Текстовий вміст елемента, якщо такий є, розміщується між цими тегами. Теги також можуть містити додаткову розмітку тегів між початком і кінцем, включаючи суміш тегів і тексту. Це вказує на подальші (вкладені) елементи як дочірні елементи батьківського елемента. Початковий тег також може включати атрибути елемента всередині тегу. Вони вказують іншу інформацію, таку як ідентифікатори розділів у документі, ідентифікатори, які використовуються для прив'язки інформації про стиль оформлення документа, а для деяких тегів, таких як ``, що використовуються для вбудовування зображень, посилання на ресурс зображення в такий формат:

					КГ05.05.001.00 ДП ПЗ	Арк.
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

``. Деякі елементи, такі як розрив рядка `
` або `
`, не дозволяють додавати будь-який вміст, текст або інші теги. Для них потрібен лише один порожній тег (як початковий тег) і не потрібно використовувати кінцевий тег. Багато тегів, зокрема закриваючий кінцевий тег для дуже часто використовуваного елемента абзацу `<p>`, є необов'язковими. Браузер HTML або інший агент може зробити висновок про закриття кінця елемента з контексту та структурних правил, визначених стандартом HTML. Проте, правильне написання розмітки є правилом гарного тону серед програмістів, а також робить код документи більш зрозумілим для аналізу та відладки. Таким чином, загальна форма елемента HTML: `<tag attribute1="value1" attribute="value2">"content"</tag>`. Деякі елементи HTML визначаються як порожні елементи і мають форму `<tag attribute1="value1" attribute2="value2">`. Порожні елементи можуть не містити вмісту, наприклад, тег `
` або вбудований тег ``. Ім'я елемента HTML - це ім'я, яке використовується в тегах. Зауважте, що назві кінцевого тегу передуює символ косої риски /, а в порожніх елементах кінцевий тег не є ані обов'язковим, ані дозволеним.

Для оформлення зовнішнього вигляду я буду використовувати каскадні таблиці стилів (CSS), так як вони чудово підтримуються HTML-документами (більш того, саме для них вони були створені), у них є можливість розділення форматування відповідно до пристрою (наприклад, для смартфонів та комп'ютерів може бути різний дизайн), а також велика кількість пре-процесорів, розроблених для роботи з CSS.

Cascading Style Sheets (CSS) - це мова таблиць стилів, яка використовується для опису представлення документа, написаного мовою розмітки, як наприклад HTML. CSS є однією з основних технологій всесвітньої мережі, поряд з HTML і JavaScript. CSS розроблено, щоб дозволити розділити оформлення та вміст, включаючи макет, кольори та шрифти. Цей поділ може покращити доступність вмісту; забезпечити більшу гнучкість і контроль у специфікації характеристик

					<i>КГ05.05.001.00 ДП ПЗ</i>	Арк.
						9
Ізм.	Лист	№ докум.	Підпис	Дата		

оформлення; дозволити кільком веб-сторінкам спільно використовувати форматування, вказавши відповідний CSS в окремому файлі .css, що зменшує складність і повторюваність структурного вмісту; і увімкнути кешування файлу .css, щоб покращити швидкість завантаження стилів між сторінками, які спільно використовують файл, та його форматування. Розділення форматування та вмісту також дає можливість представити одну й ту саму сторінку розмітки в різних стилях для різних методів відтворення, наприклад, на екрані, у друкованому вигляді, голосом (через мовленнєвий браузер або програму зчитування з екрана) та тактильні пристрої на основі Брайля. CSS також має правила альтернативного форматування, якщо доступ до вмісту здійснюється на мобільному пристрої. Каскадність назв походить від заданої схеми пріоритету, щоб визначити, яке правило стилю застосовується, якщо більше одного правила відповідає певному елементу. Ця каскадна схема пріоритетів є передбачуваною та зрозумілою. На додаток до HTML, інші мови розмітки підтримують використання CSS, включаючи XHTML, звичайний XML, SVG і XUL.

Можливість користувача взаємодіяти з сайтом (переглядати ціну в залежності від фільтрів, замовити зворотній дзвінок та інше) я реалізую за допомогою мови програмування JavaScript через гарну підтримку сучасними та старими десктопними браузерами, а також браузерами мобільних пристроїв.

JavaScript часто скорочується як JS, це мова програмування, яка є однією з основних технологій всесвітньої мережі, поряд з HTML і CSS. Понад 97% веб-сайтів використовують JavaScript на стороні клієнта для поведінки веб-сторінок, часто включають сторонні бібліотеки. Усі основні веб-браузери мають спеціальний механізм JavaScript для виконання коду на пристроях користувачів. JavaScript — це мова високого рівня, яка відповідає стандарту ECMAScript. Він має динамічне введення тексту, орієнтацію на об'єкт на основі прототипу та першокласні функції. Він багатопарадигмальний, підтримує керування подіями, функціональний та імперативний стилі програмування. Він має інтерфейси

					<i>КГ05.05.001.00 ДП ПЗ</i>	Арк.
						10
<i>Ізм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

прикладного програмування (API) для роботи з текстом, датами, регулярними виразами, стандартними структурами даних і об'єктною моделлю документа (DOM). Механізми JavaScript спочатку використовувалися лише у веб-браузерах, але тепер є основними компонентами деяких серверів і різноманітних програм. Найпопулярнішою системою виконання для такого використання є Node.js.

Використання у клієнтській веб-частині. JavaScript є домінуючою мовою сценаріїв на стороні клієнта в Інтернеті. Скрипти вбудовуються в документи HTML або включаються у них і взаємодіють з DOM. Усі основні веб-браузери мають вбудований механізм JavaScript, який виконує код на пристрої користувача. Приклади сценарної поведінки:

- Завантаження нового вмісту веб-сторінки без перезавантаження сторінки через AJAX або WebSocket. Наприклад, користувачі соціальних мереж можуть надсилати та отримувати повідомлення, не залишаючи поточної сторінки.
- Анімація веб-сторінки, наприклад зникнення та поява об'єктів, зміна розміру та переміщення їх.
- Браузерні ігри.
- Керування відтворенням потокового медіа.
- Створення спливаючих оголошень.
- Перевірка введених значень веб-форми перед відправкою даних на веб-сервер.
- Реєстрація даних про поведінку користувача та надсилання їх на сервер. Власник веб-сайту може використовувати ці дані для аналітики, відстеження реклами та персоналізації.
- Перенаправлення користувача на іншу сторінку.

Бібліотеки та фреймворки. Понад 80% веб-сайтів використовують сторонні бібліотеки JavaScript або веб-фреймворк для написання сценаріїв на стороні клієнта. jQuery є найпопулярнішою бібліотекою, яку використовують понад 75%

					<i>КГ05.05.001.00 ДП ПЗ</i>	Арк.
						11
<i>Ізм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

веб-сайтів. Facebook створив бібліотеку React для свого веб-сайту, а пізніше випустив її як відкритий код; інші сайти, включаючи Twitter, тепер використовують його. Аналогічно, фреймворк Angular, створений Google для своїх веб-сайтів, включаючи YouTube і Gmail, тепер є проектом з відкритим кодом, яким користуються інші. На відміну від цього, термін «Vanilla JS» був придуманий для веб-сайтів, які не використовують жодних бібліотек чи фреймворків, а натомість повністю покладаються на стандартну функціональність JavaScript. Мій сайт є представником так званих «Vanilla JS».

1.3. Вибір додаткових технологій для розробки

Для спрощення розробки стилів документа та додавання нових можливостей під час написання та відладки коду, я буду використовувати скриптову метамову, яка інтерпретується в каскадні таблиці стилів (CSS), під назвою SASS (SCSS).

SASS — це мова таблиць стилів, скомпільована в CSS. Вона дозволяє використовувати змінні, вкладені правила, міксини, функції та багато іншого, і все це має повністю сумісний із CSS синтаксис. SASS допомагає добре організувати великі таблиці стилів і полегшує обмін ними всередині та між проектами. Мова SASS має два синтаксиси:

- SASS - відрізняється відсутністю фігурних дужок, у ньому вкладені елементи реалізовані за допомогою відступів;
- SCSS (Sassy CSS) – використовує фігурні дужки, як і сам CSS.

Можливості SCSS:

- Змінні. Класичний CSS не підтримує використання змінних, але SCSS реалізує цю можливість. У них можна зберігати змінні кольори, стеки шрифтів або будь-які інші значення CSS, які я захочу використовувати. Щоб створити змінну в SCSS, потрібно використовувати символ \$:
- Вкладеність. Класичний CSS не підтримує вкладеність, розробник має окремо писати кожен селектор, якщо хоче додати до нього стилі. SCSS дозволяє вкладати CSS селектори так само, як і у візуальній ієрархії HTML.

					КГ05.05.001.00 ДП ПЗ	Арк.
						12
Ізм.	Лист	№ докум.	Підпис	Дата		

Але необхідно пам'ятати, що надмірна кількість вкладеностей робить документ менш читабельним, що вважається поганою практикою.

- Фрагментування. Можливість створювати фрагменти SCSS-файлу, які міститимуть у собі невеликі уривки CSS, які можна використовувати в інших SCSS-файлах. Це чудовий спосіб зробити ваш CSS модульним, а також полегшити його обслуговування. Фрагмент — це простий файл SCSS, ім'я якого починається з нижнього підкреслення, наприклад, `_partial.scss`. Нижнє підкреслення в імені SCSS-файлу говорить компілятор про те, що це тільки фрагмент і він не повинен компілюватися в CSS. Фрагменти SCSS підключаються директивою `@import`. Особливо цінною ця можливість показую себе разом с БЕМ-методологією, яку я теж використовуватиму.
- Міксини (домішки). Деякі речі в CSS дуже стомлює писати, особливо в CSS3, де плюс до всього часто потрібно використовувати велику кількість вендорних префіксів. Міксини дозволяють створювати групи декларацій CSS, які можна використовувати кілька разів на сайті. Можливо навіть передавати змінні в міксини, щоб зробити їх більш гнучкими. Також добре використовувати міксини для вендорних префіксів.
- Розширення/Спадкування. Це одна з найкорисніших функцій SCSS. Використовуючи директиву `@extend`, можна успадковувати набори властивостей CSS від одного селектора іншому. Це дозволяє тримати SCSS-файл у «чистоті».
- Математичні оператори. SCSS додає підтримку деяких стандартних математичних операторів, таких як `+`, `-`, `*`, `/` та `%`.

Для можливості використання модулів, які будуть додаватися до кожної сторінки, я буду використовувати БЕМ методологію.

БЕМ (Блок, Елемент, Модифікатор) – компонентний підхід до веб-розробки. У його основі лежить принцип поділу інтерфейсу на незалежні блоки. Він

					КГ05.05.001.00 ДП ПЗ	Арк.
						13
Ізм.	Лист	№ докум.	Підпис	Дата		

дозволяє легко та швидко розробляти інтерфейси будь-якої складності та повторно використовувати існуючий код, уникаючи «Copy-Paste».

Блок - функціонально незалежний компонент сторінки, який можна повторно використати. HTML блоки представлені атрибутом class. Блоки можна вкладати один в одного. Допустима будь-яка вкладеність блоків.

Елемент - складова частина блоку, яка не може використовуватись у відриві від нього. Елементи можна вкладати один одного. Допустима будь-яка вкладеність елементів. Елемент завжди частина блоку, а не іншого елемента. Ім'я блоку задає простір імен, що гарантує залежність елементів від блоку (block__elem). Елемент – необов'язковий компонент блоку. Не у всіх блоків мають бути елементи.

Модифікатор - сутність, що визначає зовнішній вигляд, стан та поведінку блоку або елемента. Назва модифікатора характеризує зовнішній вигляд ("який розмір?", "яка тема?" і т. п. - "розмір": size_s, "тема": theme_islands), стан ("чим відрізняється від інших?" - "відключено"): disabled, "фокусований": focused) і поведінка ("як поводитьсья?", "як взаємодіє з користувачем?" - "напрямок": directions_left-top). Ім'я модифікатора відокремлюється від імені блоку або елемента одним підкресленням (_).

Для створення слайдеру, я буду використовувати плагін SwiperJS, через його велику функціональність, подальший розвиток спільноту, а також підтримку пристроїв із сенсорними екранами.

Swiper — це найсучасніший безкоштовний мобільний сенсорний слайдер з апаратним прискоренням переходів і дивовижною нативною поведінкою. Він призначений для використання на мобільних веб-сайтах, мобільних веб-програмах і мобільних нативних/гібридних програмах. Для його використання, необхідно підключити до мого документу стилі слайдеру, а також його програмний JavaScript-код. Після цього правильно оформити структуру HTML

					КГ05.05.001.00 ДП ПЗ	Арк.
						14
Ізм.	Лист	№ докум.	Підпис	Дата		

згідно документації. Це можна зробити за допомогою спеціальних «вбудованих» класів, а також правильною вкладеністю елементів.

Після цього необхідно ініціалізувати слайдер за допомогою вбудованого класу `Swiper`. У ньому ж можна налаштувати його поведінку. У подальшому, я можу оперувати методами, викликаючи їх на створеному об'єкті класу `Swiper`.

Для резервного копіювання та спрощення роботи я буду використовувати розподілену систему контролю `Git`. Основна відмінність `Git` від будь-якої іншої системи контролю версій — це підхід до роботи із своїми даними. Концептуально, більшість інших систем зберігають інформацію як списку змін у файлах. Ці системи представляють збережену інформацію у вигляді набору файлів та змін, зроблених у кожному файлі, за часом (зазвичай це називають контролем версій, заснованим на відмінностях). `Git` не зберігає та не обробляє дані таким способом. Замість цього, підхід `Git` до збереження даних більше схожий на набір знімків мініатюрної файлової системи. Щоразу, коли ви робите коміт, тобто зберігаєте стан свого проекту в `Git`, система запам'ятовує, як виглядає кожен файл у цей момент, та зберігає посилання на цей знімок. Для підвищення ефективності, якщо файли не були змінені, `Git` не запам'ятовує ці файли знову, а лише створює посилання на попередню версію ідентичного файлу, який вже збережено. `Git` представляє свої дані, як, скажімо, потік знімків.

Також для роботи більшості операцій у `Git` достатньо локальних файлів та ресурсів — в основному, системі не потрібна жодна інформація з інших комп'ютерів у вашій мережі. Оскільки вся історія проекту зберігається прямо на вашому локальному диску, більшість операцій здаються мало не миттєвими.

Git - розподілена система керування версіями. Проект був створений Лінусом Торвальдсом для управління розробкою ядра `Linux`, першу версію випущено 7 квітня 2005 року. На сьогоднішній день його підтримує Джуніо Хамано. Система спроектована як набір програм, спеціально розроблених з урахуванням їх використання у сценаріях. Це дозволяє зручно створювати спеціалізовані системи

					КГ05.05.001.00 ДП ПЗ	Арк.
						15
Ізм.	Лист	№ докум.	Підпис	Дата		

контролю версій на базі Git або інтерфейси користувача. Наприклад, Cogito є таким прикладом оболонки до репозиторія Git, а StGit використовує Git для керування колекцією виправлень (патчів). Git підтримує швидкий поділ та злиття версій, включає інструменти для візуалізації та навігації з нелінійної історії розробки. Як і Darcs, BitKeeper, Mercurial, Bazaar та Monotone, Git надає кожному розробнику локальну копію всієї історії розробки, зміни копіюються з одного репозиторію до іншого. Віддалений доступ до репозиторію Git забезпечується git-демоном, SSH або HTTP-сервером. TCP-сервіс git-daemon входить до дистрибутиву Git і є поряд з SSH найбільш поширеним і надійним методом доступу. Метод доступу по HTTP, незважаючи на ряд обмежень, дуже популярний у контрольованих мережах, тому що дозволяє використовувати наявні конфігурації мережевих фільтрів.

1.4. Висновки

Отже, веб-сайт буде розроблятися за допомогою HTML, CSS та JS (ES6). Такі технології, як SCSS, БЕМ оптимізують процес розробки, а система контролю версій Git допоможе створювати резервні копії.

					<i>КГ05.05.001.00 ДП ПЗ</i>	Арк.
						16
<i>Ізм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

2. ОСНОВНА ЧАСТИНА

2.1. Етапи розробки веб-сайту

Нижче наведено етапи розробки мого веб-сайту.

1. Первісна підготовка та налаштування усього проекту.
2. Первісна підготовка та налаштування сторінки.
3. Додавання модуля або його створення, якщо він ще не зроблений
 - а) Створення розмітки модуля.
 - б) Створення стилів для модуля
 - в) Програмування модуля (за потреби)
 - г) Реалізація адаптивності модуля

Пункт 2 виконується 4 рази, адже веб-сайт використовує 4 сторінки. Пункт 3 виконується велику кількість разів, адже кожна сторінка складається з багатьох модулів. Взагалі, розробку можна представити наступною блок-схемою (Рисунок 2.1.1 (а)):

					<i>КГ05.05.002.00 ДП ПЗ</i>	Арк.
						17
<i>Ізм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

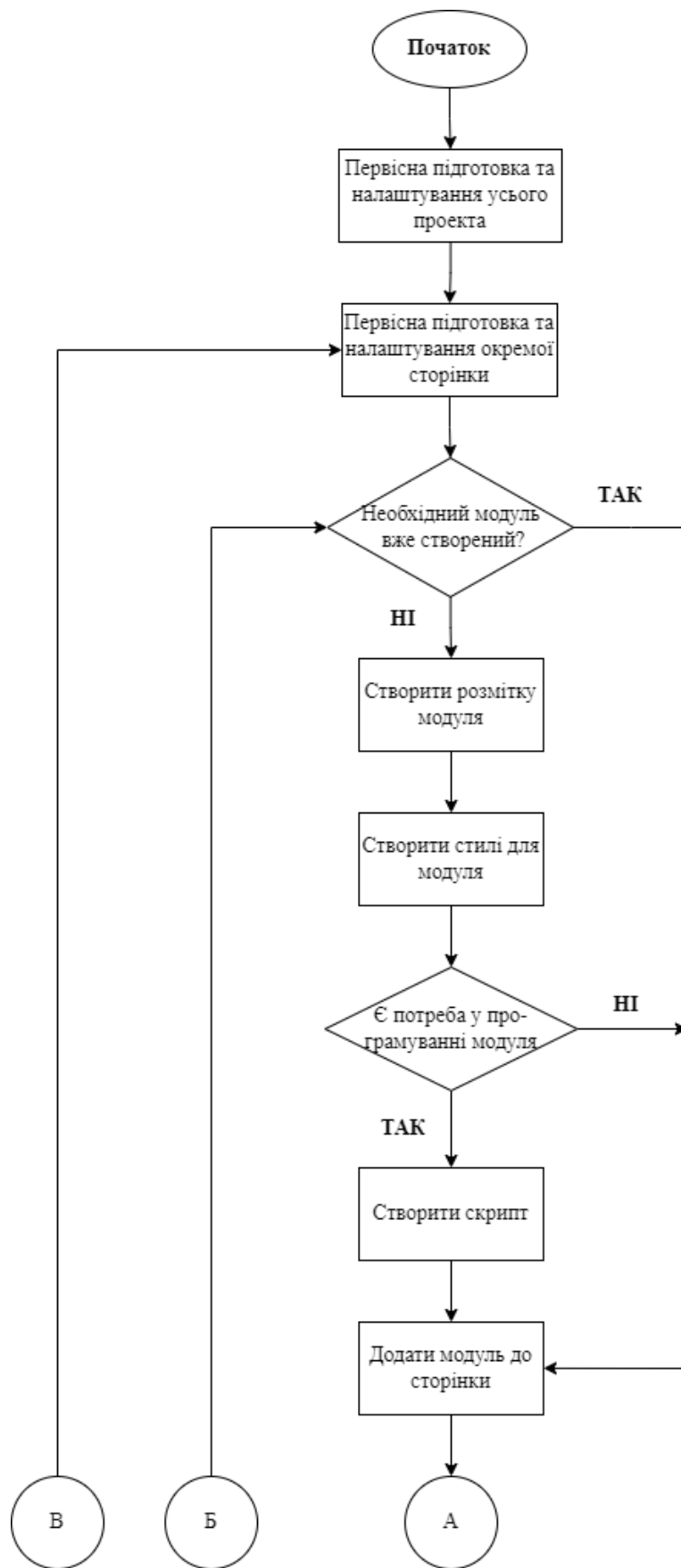


Рисунок 2.1.1 (а) - Блок-схема алгоритму розробки сайту

Ізм.	Лист	№ докум.	Підпис	Дата

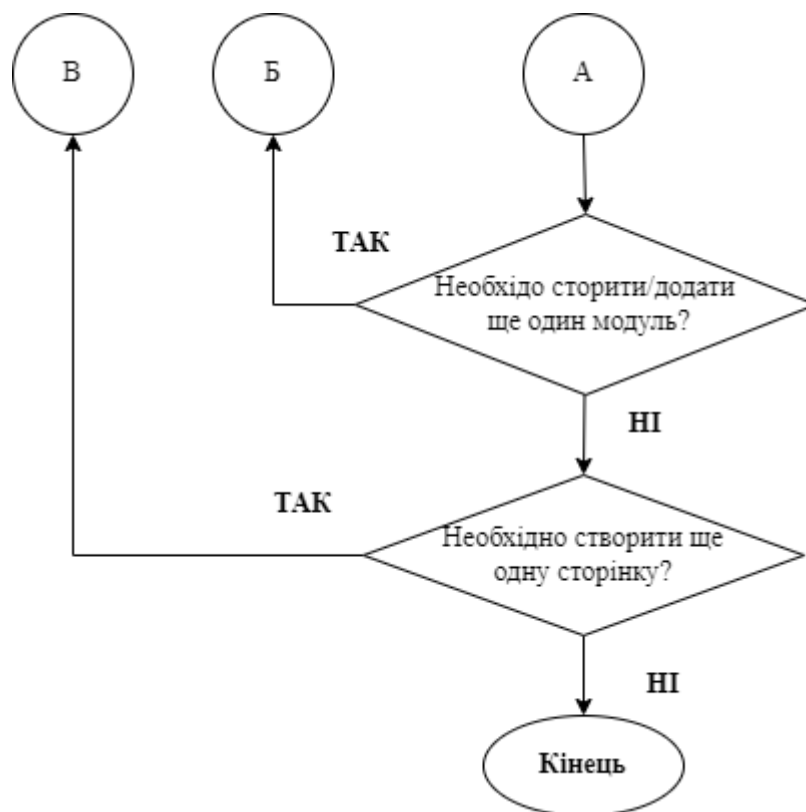


Рисунок 2.1.1 (б) - Блок-схема алгоритму розробки сайту

2.2. Первісна підготовка та налаштування усього проекту

Первісна підготовка - це перший етап створення мого сайту. На цьому етапі я створюю необхідні каталоги, додаю плагін SwiperJS, створюю допоміжні SCSS-файли, а також налаштовую Git.

Створю папку для фав-іконок та назву її «favicons». Фав-іконка - це невелика картинка, яка відображається у вкладці разом із назвою сторінки. Створити її не важко, але у кожного браузера, операційної системи свої вимоги до фав-іконки, вона має бути певного розміру, формату. Це можна зробити самому, а можна скористатися онлайн-сервісами, так званими генераторами фав-іконок. Вони приймають зображення, частіше всього у форматах .svg, .png, .jpeg та розміром до 160x160. А повертають файл з HTML-кодом підключення, так папку з різними

форматами та розмірами цією іконки. Необхідно лише додати цей HTML-код до сторінки, правильно прописуючи шлях до файлів.

Далі, я створю папку «fonts». Як нескладно здогадатися, в цій папці будуть зберігатися шрифти. Але, крім текстових, я також створю шрифт-іконок. Я це зроблю по причині того, що деякі svg-зображення маленького розміру (наприклад: значки Facebook, Instagram, телефону, стрілка вниз) зручніше використовувати не як окремі svg-файли, а як частину шрифта-іконок. Після цього я зможу додавати стилі, як до звичайного тексту, що буде більш універсальним підходом, ніж створювати окремі класи та стилі для цих маленьких зображень. Створити шрифт-іконок можна також за допомогою онлайн-сервісів, вони приймають svg-файл, дають можливість невеликого їх корегування, після чого дають можливість завантажити їх як звичайні шрифти, а також файл з CSS-кодом їх підключення та класи для їх використання.

Для того щоб додати шрифт-ікону, необхідно лише додати пару тегів (саме які можна обрати на сайті генератора, частіше за все це пара `<i></i>`) та додати відповідний клас цієї іконки (назву для класів я також обираю на сайті генератора)

```
<a class="social-link main-nav__social-link" href="#" alt="Facebook">  
  <i class="icon-facebook"></i>  
</a>
```

Рисунок 2.2.1 - Приклад використання шрифт-іконок

Варто зауважити, що тут я також додав звичайний клас «social-link», який відповідає за стилізацію тексту. Якби я використовував іконку як зображення, то мені прийшлося б створювати окремі класи, а також налаштовувати позиціонування відносно тексту.

```
<div class="content-block">  
    
  <p class="text">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>  
</div>
```

Рисунок 2.2.2 - Приклад використання іконки як окремого зображення

					КГ05.05.002.00 ДП ПЗ	Арк.
						20
Ізм.	Лист	№ докум.	Підпис	Дата		

Далі, я створю папку «js-lib». У неї я буду додавати створені js-файли, які я буду часто використовувати, але вони не є частиною якогось модуля. Також, у цій папці я створю каталог «Swiper». В нього я додам файли скрипта SwiperJS, а саме swiper-bundle.min.js та _swiper-bundle.min.scss.

Далі, я створю папки «module» та «pages», де, відповідно, я буду зберігати модулі та сторінки.

Далі, я створю папку «scss-elements». В цій папці я буду зберігати файли, які не є блоками (тобто модулями). Їх можна використовувати у будь-якому модулі, але не можна використовувати окремо. По своїй суті це класи з стилями, завдяки яким я можу швидко додати необхідний зовнішній вигляд блокам або елементам блока.

Далі, я створю папку «scss-lib», де я буду зберігати спеціальні «технічні» файли SCSS. Це не стилі, вони не мають значення, якщо їх використовувати окремо від SCSS. Це всього три файли: змінні кольорів, загальний файл шрифтів, міксини. Я детально опишу кожен з них.

Перший, це файл змінних кольорів. З самого початку створення сайту, я додав сюди змінні зі значенням кольорів з макету. Це полегшує написання стилів та є більш інтуїтивно зрозумілим.

```
$dark-red: #B60D00;
$dark-red-active: #a50b00;
$salmon: #EE746A;
$purple: #6B356E;
$purple-active: #4e2650;
$light-pink: #F8C4ED;
$light-red: #F77E72;
$light-orange: #FFBDA2;
$light-lemon: #FFFFD7;
$dark-grey: #363636;
$black: #000000;
$white: #FFFFFF;
```

Рисунок 2.2.3 - Файл _color.scss

					КГ05.05.002.00 ДП ПЗ	Арк.
						21
Ізм.	Лист	№ докум.	Підпис	Дата		

Другий, це загальний файл шрифтів. Так як кожна сторінка використовує одні й ті самі шрифти, щоб кожен раз не писати або копіювати код, я просто імпортую цей файл у загальний scss-файл.

Третій файл - це файл міксинів. Це шаблони для швидкого написання CSS-коду. Наприклад, на рисунку 2.2.7 зображено код міксинів для медіа-виразів. Замість того, щоб писати «@media (max-width: 1360px)», я можу просто написати «@include media-l». Також, замість того, щоб запам'ятовувати цифри, я використовую букви.

```
@mixin media-l {
  @media (max-width: 1360px) { @content; }
}

@mixin media-m {
  @media (max-width: 1024px) { @content; }
}

@mixin media-s {
  @media (max-width: 768px) { @content; }
}

@mixin media-xs {
  @media (max-width: 576px) { @content; }
}
```

Рисунок 2.2.4 - Приклад медіа-міксинів

Також у цьому файлі є міксини для флекс-боксів (flex-box), замість того щоб писати «display: flex; flex-direction: column; justify-content: center; align-items: center;», я просто прописую «@include flex-column-center-center».

					КГ05.05.002.00 ДП ПЗ	Арк.
						22
Ізм.	Лист	№ докум.	Підпис	Дата		

```

@mixin flex-center {
  display: flex;
  justify-content: center;
}

@mixin flex-center-center {
  @include flex-center;
  align-items: center;
}

@mixin flex-column-center {
  display: flex;
  flex-direction: column;
  align-items: center;
}

@mixin flex-column-center-center {
  @include flex-center-center;
  flex-direction: column;
}

@mixin flex-builder($direction, $justify, $align) {
  display: flex;
  flex-direction: $direction;
  justify-content: $justify;
  align-items: $align;
}

```

Рисунок 2.2.5 - Приклад міксинів флекс-боксів

Далі, я маю налаштувати компіляцію SCSS-коду у CSS-код. Є багато варіантів вирішення цього питання. Можна використовувати окремі прт-модулі, або модулі у WebPack, розширення до Visual Studio Code (саме в ньому я працюю), або скористатися спеціальною програмою. Я буду використовувати безкоштовну програму Prepros. Мені потрібно лише указати файл, який необхідно компілювати, та місце, де буде створено CSS-файл на його основі. А також додаткові опцію, такі як мініфікація коду, додання вендорних префіксів.

І останнє, це налаштування системи контролю версій Git. Спочатку, мені необхідно ініціалізувати Git-репозиторій у моєму каталозі. Це можна зробити командою «git init». Не всі файли мають бути занесені до версійного контролю, наприклад файл конфігурації prepros.congif або файли прт-модулів (якщо використовується WebPack чи інші модулі). Для цього необхідно створити файл .gitignore. У ньому я маю описати за допомогою регулярних виразів та інших правил файли, які не треба відстежувати.

Правила .gitignore:

					КГ05.05.002.00 ДП ПЗ	Арк.
						23
Ізм.	Лист	№ докум.	Підпис	Дата		

- Порожні рядки, а також рядки, що починаються із #, ігноруються.
- Стандартні шаблони є глобальними та застосовуються рекурсивно для всього дерева каталогів.
- Щоб уникнути рекурсії, використовуйте символ слеш (/) на початку шаблону.
- Щоб виключити каталог, додайте слеш (/) до кінця шаблону.
- Можна інвертувати шаблон, використовуючи знак оклику (!) як перший символ.

Працюючи з гіт, я буду користуватися гілками, довгоіснуючими та тематичними. Багато розробників, які використовують Git, дотримуються саме такого підходу, залишаючи повністю стабільний код тільки у гілці «master» - можливо, тільки той код, який був або буде випущений. При цьому існує паралельна гілка з ім'ям «develop» або «next», призначена для роботи та тестування стабільності; вона не обов'язково повинна бути завжди стабільною, але при досягненні стабільного стану її вміст можна злити у гілку «master». Вона використовується для злиття завершених завдань із тематичних гілок щоб гарантувати, що ці завдання проходять тестування і не роблять помилок. Загалом це можна у вигляді накопичувачів, у яких набори комітів переміщуються більш стабільний рівень лише після повного тестування.

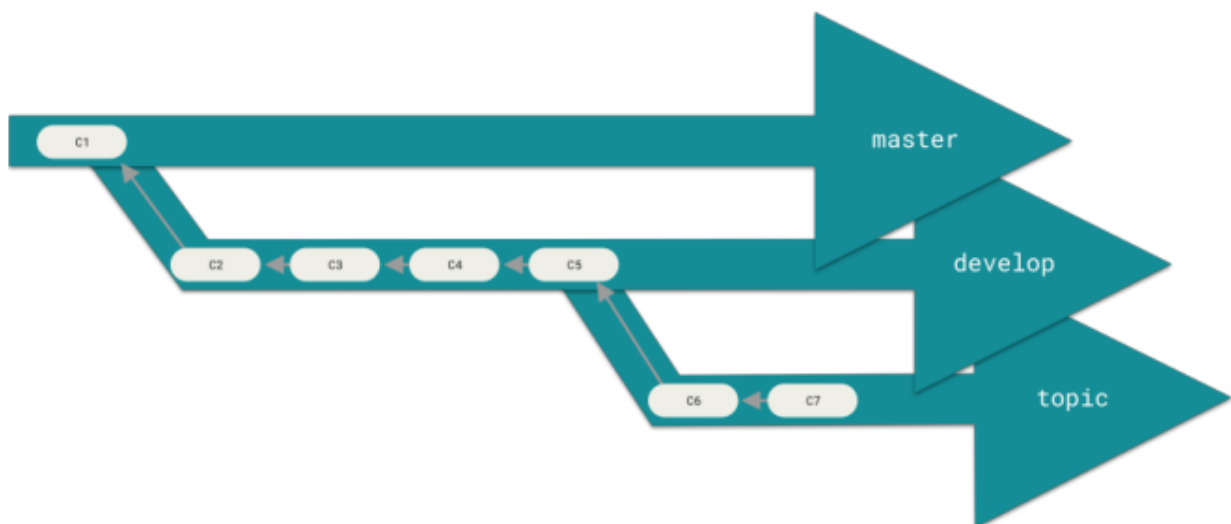


Рисунок 2.2.6 - Діаграма стабільності гілок у вигляді багаторівневого накопичення

Також, однією з причин, з якої я використовую Git - це гарна кооперація з сервісом GitHub, адже крім локальних резервних копій, необхідно мати копії у хмарному сховищі. Для цього мені необхідно створити аккаунт GitHub (якщо немає), створити віддалений репозиторій на сайті GitHub та підключити його локально у моєму репозиторії за допомогою команди «git remote add <name> <url>». Після цього я можу завантажувати проект у хмарне сховище.

Отже, я налаштував проект і готовий до подальшої розробки.

2.3. Первісна підготовка та налаштування сторінки

На другому етапі необхідно здійснити налаштування окремої сторінки. Спочатку у папці «pages» необхідно створити папку для моєї сторінки. Саме у ній я буду працювати. Я створюю файл index.scss. Це головний SCSS-файл, саме в ньому я імпортую всі необхідні SCSS-файли модулів. Його можна поділити на 4 частини. У першій частині я відміню базові стилі браузеру, щоб не виникало конфліктів з моїми. У другій я імпортую файли з папки «scss-libs» - змінні кольорів, міксини, файл шрифтів, а також стилі плагіна SwiperJS. Ці дві частини у всіх сторінок однакові. У третій частині я імпортую необхідні елементи з папки scss-elements, а у четвертій необхідні стилі модулів.

Далі, у програмі Prepros, як було написано раніше, необхідно вказати який файл SCSS компілювати та куди.

Наступне, що потрібно зробити, це створити файл HTML, зробити базову розмітку (рисунок 2.3.2).

					КГ05.05.002.00 ДП ПЗ	Арк.
						25
Ізм.	Лист	№ докум.	Підпис	Дата		

```

<!DOCTYPE html>
<html lang="uk">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<body>

</body>

</html>

```

Рисунок 2.3.1 - Базова (початкова) розмітка

Далі, вказати <title>, тобто назву сторінки, підключити фав-іконки (як було описано раніше), та підключити скомпільований CSS-файл. Залишається лише створити тематичну гілку сторінки у Git за допомогою команди «git branch <branch-name>» та переключитися на неї за допомогою команду «git checkout <branch-name>». Тепер можна створювати коміти та відправляти зміни на сервер, а коли робота з цією сторінкою буде закінчена, я її з'єднаю зі стабільною гілкою «dev». Тепер можна створювати та підключати модулі.

2.4. Додавання модуля або його створення

Процес створення та додання модуля до сторінки я продемонструю на основі модуля модально вікна «Замовити зворотній дзвінок». Спочатку, у папці «modules» необхідно створити папку модуля. У ній я створюю папку «scss», де буде зберігатися SCSS-файл. У цьому модулі використовуються три SVG-зображення, тому я також створюю папку «svg», де вони будуть зберігатися.

Тепер я роблю розмітку. Так як сама форма виклику не займає всю ширину браузеру, я створю семантичний тег <section>, у якому буде форма, та додам до

					КГ05.05.002.00 ДП ПЗ	Арк.
						26
Ізм.	Лист	№ докум.	Підпис	Дата		

нього клас, який додає відступи по бокам, а також центрує форму посередині. (Рисунок 2.4.1). До того ж, сам тег `<forma>` є також семантичним.

```
<section class="request-call-form-modal-window">
  <form class="request-call-form-wrapper">
  </form>
</section>
```

Рисунок 2.4.1 - Тег `<section>` всередині якого `<form>`

Далі, я додам блоки `<div>`, їм буде задані розміри та задній фон. Так, можна не використовувати задній фон, а просто додати зображення за допомогою тегу ``, але в цьому випадку заднім фоном буде зручніше маніпулювати при додаванні адаптивності, тобто підтримки мобільних пристроїв.

```
<section class="request-call-form-modal-window">
  <form class="request-call-form-wrapper">
    <div class="request-call-form-wrapper__background-left"></div>
    <div class="request-call-form-wrapper__background-right"></div>
  </form>
</section>
```

Рисунок 2.4.2 - Блоки для заднього фону

Далі, я додам `<div>` блок який буде містити увесь інтерактивний контент форми. У ньому я додам назву форми, поля введення інформації, а також кнопку «Надіслати».

```
<section class="request-call-form-modal-window">
  <form class="request-call-form-wrapper">
    <div class="request-call-form-wrapper__background-left"></div>
    <div class="request-call-form-wrapper__content">
      <p class="title title_small request-call-form-wrapper__title_small">
        Замовити дзвінок
      </p>
      <input class="input request-call-form-wrapper__input" type="text" name="user-name" placeholder="Ім'я">
      <input class="input request-call-form-wrapper__input" type="tel" name="user-tel" placeholder="Телефон">
      <button class="button button_shorter request-call-form-wrapper__button_shorter">Надіслати</button>
    </div>
    <div class="request-call-form-wrapper__background-right"></div>
  </form>
</section>
```

Рисунок 2.4.3 - Форма з інтерактивними елементами

					КГ05.05.002.00 ДП ПЗ	Арк.
						27
Ізм.	Лист	№ докум.	Підпис	Дата		

Як можна побачити, крім елементів блоку «request-call-form-wrapper», я використовую окремі класи (елементи), такі як «input», «button», «title». Про них було описано раніше, вони не є частиною якогось блоку, а також не можуть використовуватись окремо. Їх можна міксувати з елементами блока (або з усім блоком). Ну і останнє, це кнопка закриття модального вікна. Я його зробив за допомогою стилізації елементів <div> та їх трансформування. Звичайно, можна просто використати іконку, це лише один із варіантів реалізації.

```
<section class="request-call-form-modal-window">
  <form class="request-call-form-wrapper">
    <div class="request-call-form-wrapper__background-left"></div>
    <div class="request-call-form-wrapper__content">
      <p class="title title_small request-call-form-wrapper__title_small">
        Заповіти дзвінок
      </p>
      <input class="input request-call-form-wrapper__input" type="text" name="user-name" placeholder="Ім'я">
      <input class="input request-call-form-wrapper__input" type="tel" name="user-tel" placeholder="Телефон">
      <button class="button button_shorter request-call-form-wrapper__button_shorter">Надіслати</button>
    </div>
    <div class="request-call-form-wrapper__background-right"></div>
    <button class="button-close request-call-form-wrapper__button-close" data-toggle-modal="request-call-form">
      <div class="button-close__line-1"></div>
      <div class="button-close__line-2"></div>
    </button>
  </form>
</section>
```

Рисунок 2.4.4 - Повністю готовий HTML-код блоку

До речі, кнопка закриття також є окремим елементом.

Тепер необхідно створити стилі для цього модуля. Задам стилі для всієї форми, а також для блоків із заднім фоном.

					КГ05.05.002.00 ДП ПЗ	Арк.
						28
Ізм.	Лист	№ докум.	Підпис	Дата		

Я додав основну стилізацію для цього блоку, але необхідно також реалізувати відображення для смартфонів, планшетів та інших пристроїв. Це можна зробити за допомогою медіа-виразів. Їх легко використовувати, необхідно лише описати до якої, або після якої ширини екрану використовувати певні стилі, і описати ці стилі. А з вже готовими міксинами це буде набагато швидше.

```
.request-call-form-wrapper {  
  width: 860px;  
  height: 490px;  
  display: flex;  
  overflow: hidden;  
  background: $light-lemon;  
  border-radius: 70px;  
  transition: all .3s ease-in-out;  
  position: relative;  
  
  @include media-m {  
    width: 100%;  
    @include flex-builder(row, space-between, none);  
  }  
  
  @include media-s {  
    height: auto;  
    @include flex-builder(column, none, center);  
  }  
}
```

Рисунок 2.4.7 - Використання медіа-міксинів для реалізації підтримки мобільних пристроїв

На рисунку 2.4.7 можна побачити стандартні стилі, та стилі які будуть примінятися на певній ширині екрану. Вони не повністю перезаписують стилі, а лише доповнюють/оновлюють існуючі. В даному випадку на ширині «М», що дорівнює максимальній ширині в 1024 пікселі, форма почне займати всю доступну ширину екрана, а також зміниться порядок елементів у ній - вони будуть розташовуватися колоною. А на ширині «S», що дорівнює максимальній ширині в 768 пікселів, відмінюється фіксована висота блоку, а також змінюється порядок розташування елементів.

Тепер, необхідно додати інтерактив до цього блоку. Так як форма зворотного виклику не єдина на сторінці, то скрипт буде універсальним та підтримувати будь-які форми у будь-якій кількості. Загальні відомості такі: форма прозора та знаходиться за екраном, натискаючи на кнопку, вона за допомогою анімації стає

					КГ05.05.002.00 ДП ПЗ	Арк.
						30
Ізм.	Лист	№ докум.	Підпис	Дата		

непрозорою та по осі Y рухається до середини екрану. Прокрутка сторінки блокується, але якщо форма не вміщається на екрані, то тільки їй додається прокрутка. Далі, якщо натиснути на кнопку закриття, або на задній фон за межами форми, то вона рухається по осі Y за екран та стає прозорою. Прокрутка сторінки знов дозволяється. Є декілька варіантів реалізації: можна створити звичайну функцію, можна використати функцію-конструктор та додати методи у її властивість «prototype», я ж використаю новий ES6 синтаксис, а саме - класи. Також, я буду маніпулювати браузерним API для керування DOM-елементами сторінки.

Спочатку необхідно налаштувати функцію-конструктор у класі. Вона буде приймати елемент (в даному випадку форму зворотного зв'язку), який необхідно перетворити у модальне вікно, кнопки, які показують/закривають модальне вікно, а також CSS-класи: клас, який перетворює елемент у модальне вікно (налаштовує позиціонування), клас, який робить вікно прозорим, та клас, який відповідає за анімацію пересування по осі Y.

```
class Modal {
  constructor(options) {
    this.modal = document.querySelector(options.modal_selector);
    this.buttons = document.querySelectorAll(options.toggle_button_selector);
    this.child = this.modal.firstChild;

    this.hide_class = options.hide_class;
    this.modal_class = options.modal_class;
    this.animation_class = options.animation_class;
  }
}
```

Рисунок 2.4.8 - Клас та функція-конструктор у ньому

Після передачі всіх необхідних даних, необхідно зробити налаштування. Воно здійснюється один раз при створенні об'єкту класу. Увесь код буде у функції «setup», яка буде викликатися у функції конструкторі. Ця функція додасть формі клас, який перетворює її у модальне вікно, клас, який робить вікно прозорим, додасть слухач подій «клік» на необхідні кнопки закриття/відкриття, який при

										Арк.
										31
Ізм.	Лист	№ докум.	Підпис	Дата						

випадку забороняється прокрутка всієї сторінки, та на місце скроллу ставиться відступ, бо без нього сторінка «скачне», що не дуже гарно. Ширину відступу надає сетер «scrollWidth».

```
get scrollWidth() {
    return window.innerWidth - document.documentElement.clientWidth;
}

checkContentOverflow() {
    let clientHeight = document.documentElement.clientHeight;
    let childHeight = (getComputedStyle(this.child).height).match(/\d+/g)[0];

    childHeight > clientHeight ? this.contentOverflow = true : this.contentOverflow = false;
}
```

Рисунок 2.4.11 - Гетер «scrollWidth» та функція «checkContentOverflow» всередині класу

```
show() {
    this.modal.classList.remove(this.hide_class);
    this.child.classList.remove(this.animation_class);

    this.checkContentOverflow();

    if (this.contentOverflow) {
        this.modal.style.overflow = "scroll";
        this.modal.style.alignItems = "start";
        this.modal.style.paddingTop = "5%";
        this.modal.style.paddingBottom = "5%";
    }

    let scrollWidth = this.scrollWidth;

    document.body.style.overflow = "hidden";
    document.body.style.paddingRight = scrollWidth + "px";
}
```

Рисунок 2.4.12 - Функція «show» всередині класу

Функція «hide», робить зовсім протилежне, додає клас, який робить вікно прозорим, додає анімацію яка пересуває вікно по осі Y за екран, дає можливість

прокручувати сторінку, а також якщо у модального вікна була своя прокрутка, то її відмінняє.

```
hide() {
  this.modal.classList.add(this.hide_class);
  this.child.classList.add(this.animation_class);

  if (this.contentOverflow) {
    this.modal.style.overflow = "";
    this.modal.style.alignItems = "";
    this.modal.style.paddingTop = "";
    this.modal.style.paddingBottom = "";

    this.contentOverflow = false
  }

  document.body.style.overflow = "";
  document.body.style.paddingRight = "";
}
```

Рисунок 2.4.13 - Функція «hide» всередині класу

Далі, функція «backdropClose», яку викликає клік на модальному вікні. Вона перевіряє, чи на задньому фоні відбувся клік, і якщо так, то викликає функцію «hide», яка закриває модальне вікно.

```
backdropClose(e) {
  e.preventDefault();

  if (e.target === this.modal) {
    this.hide();
  }
}
```

Рисунок 2.4.14 - Функція «backdropClose» всередині класу

Ну і останнє, це функція «orientationChange», яку викликає подія перевертання пристрою, наприклад коли користувач змінює положення смартфона з вертикального на горизонтальне. Ця функція перевіряє чи відкрите модальне вікно і якщо так, то закриває його, при цьому вся введена інформація залишається. Це

лише один із варіантів реалізації, при перегортанні можна також повністю перелаштовувати модальне вікно або сторінку.

```
orientationChange() {
    if (!this.isClose) {
        this.hide();
    }
}
```

Рисунок 2.4.15 - Функція «orientationChange» всередині класу

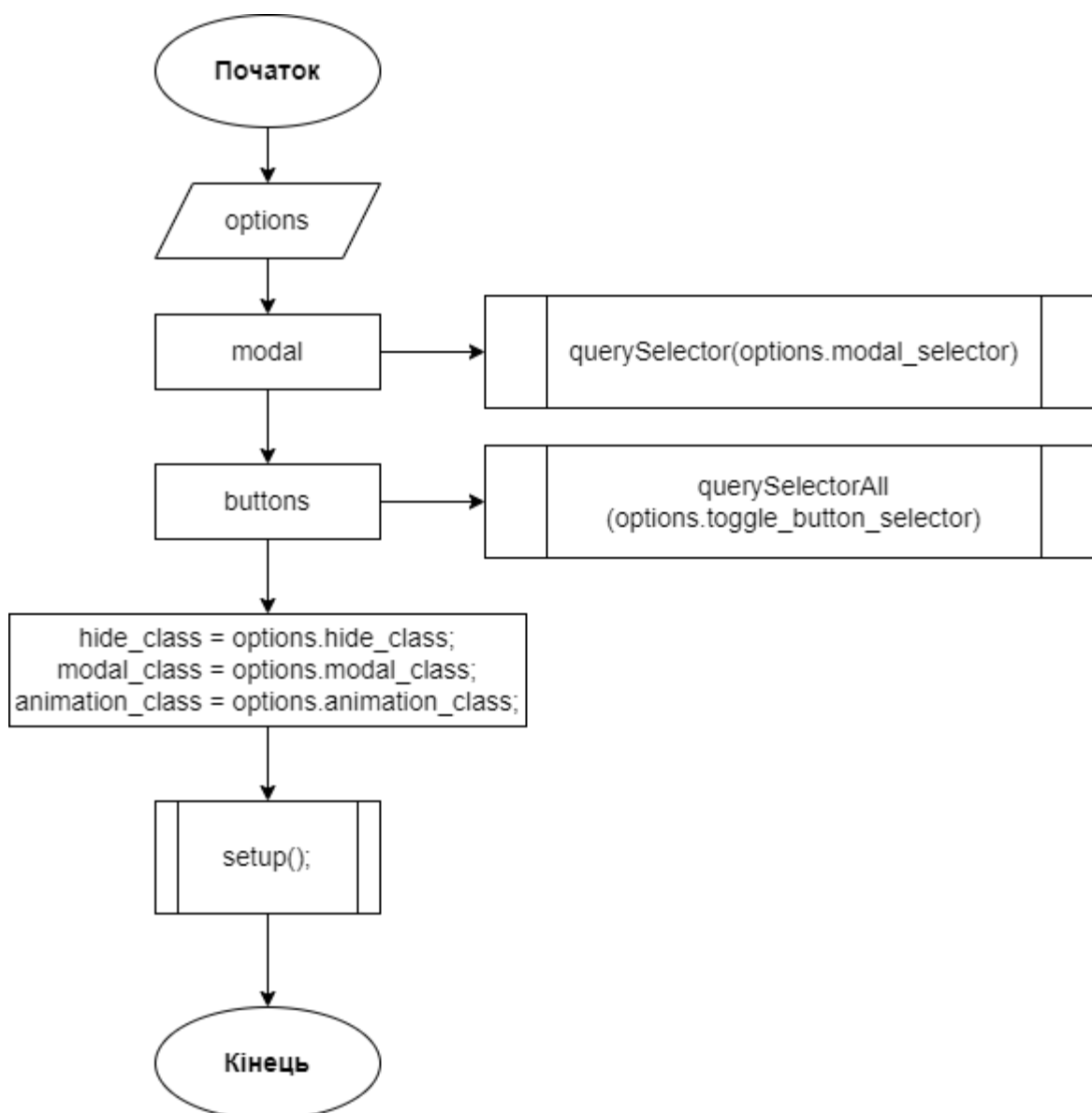


Рисунок 2.4.16 - Блок-схема алгоритму функції «constructor» у класі

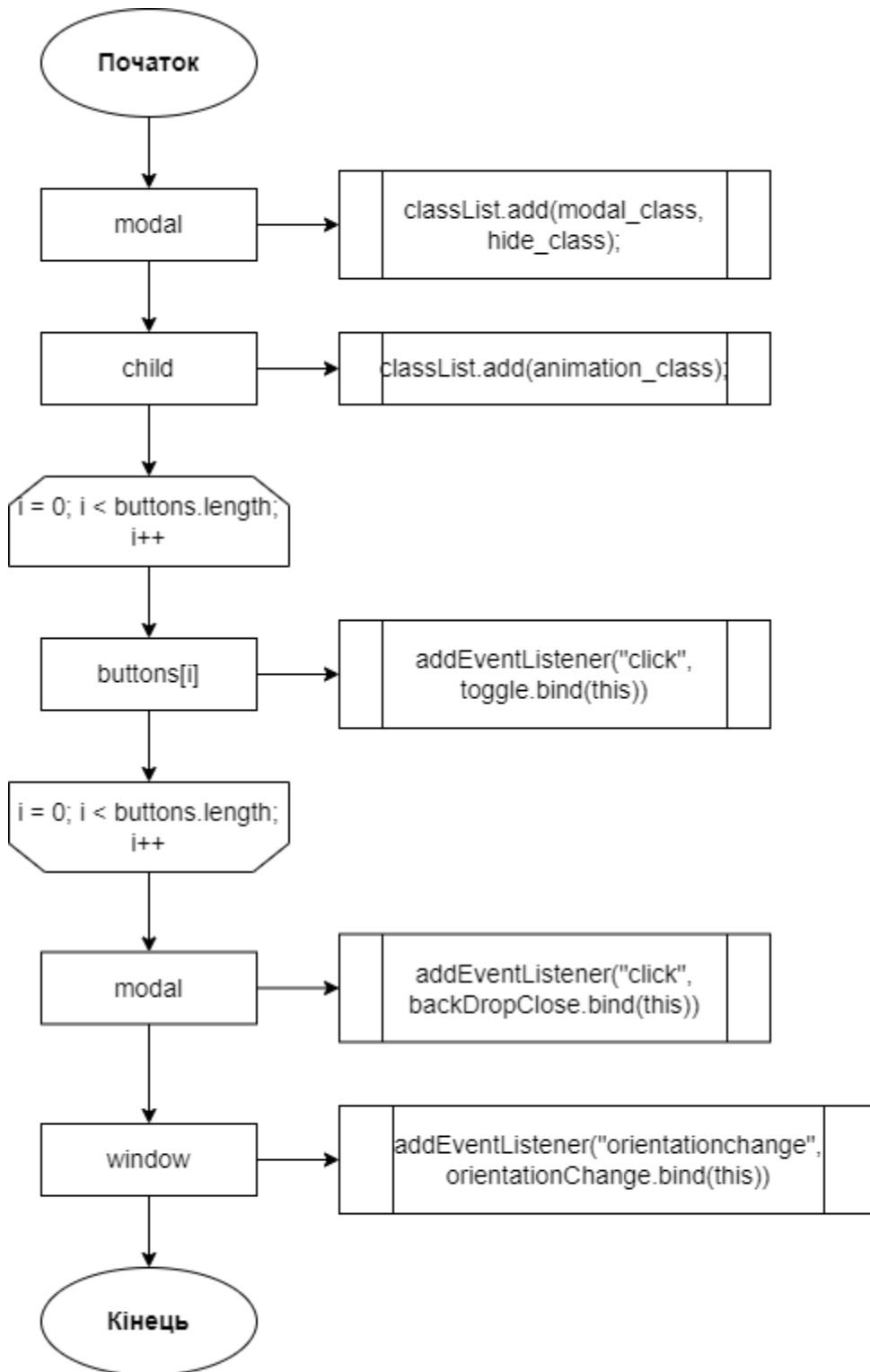


Рисунок 2.4.17 - Блок-схема алгоритму функції «setup» у класі

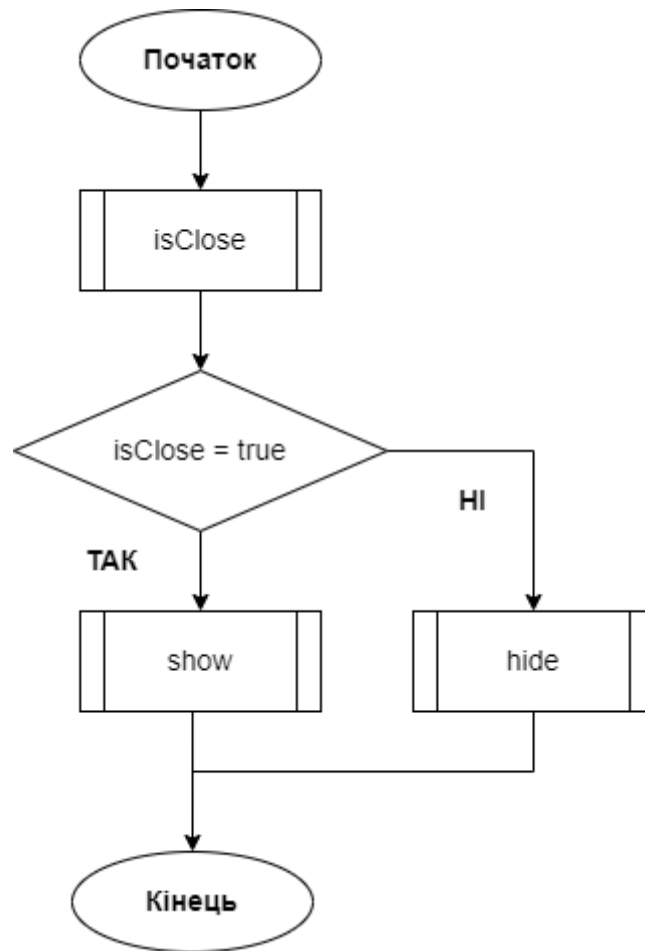


Рисунок 2.4.18 - Блок-схема алгоритму функції «toggle» у класі

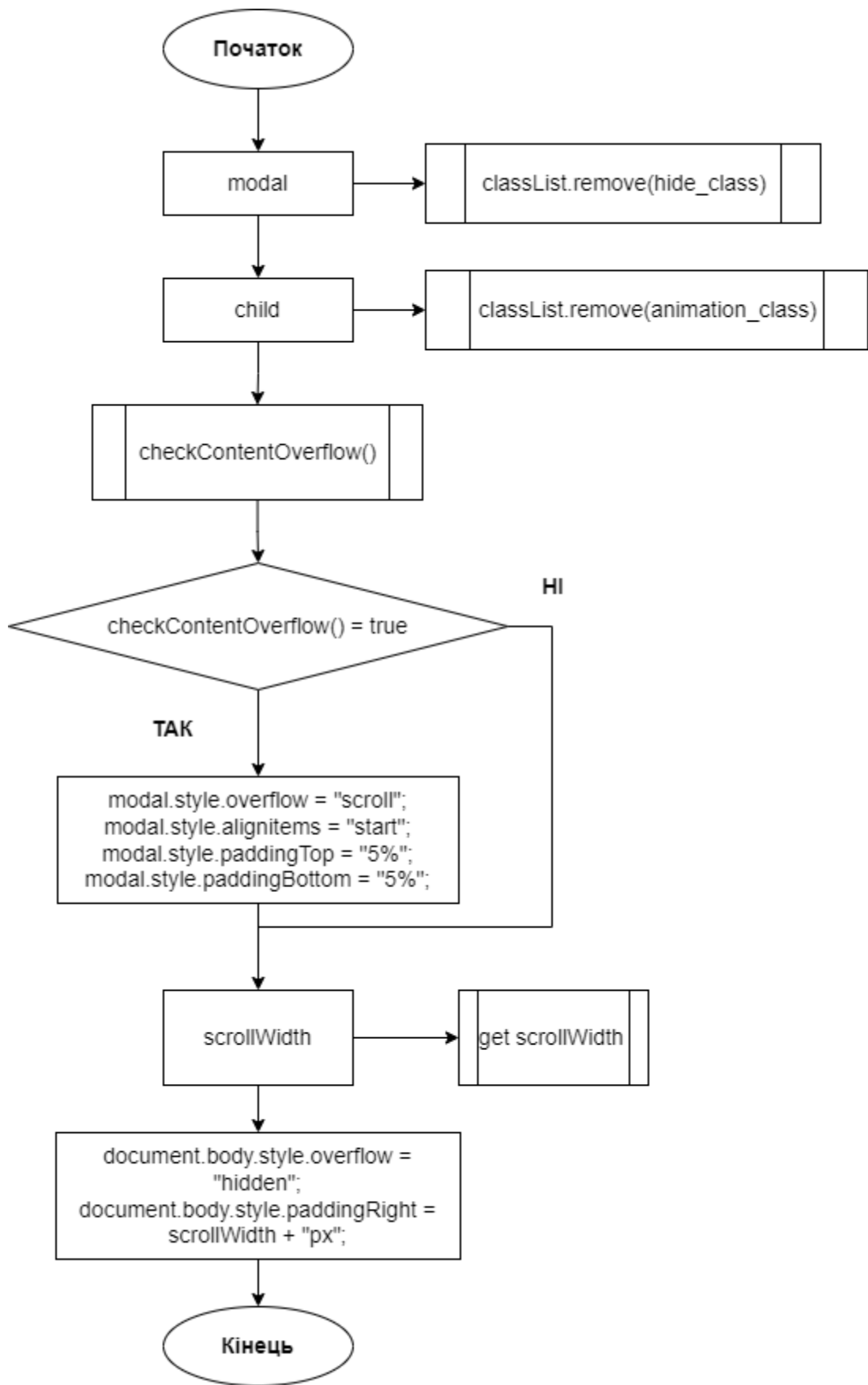


Рисунок 2.4.19 - Блок-схема алгоритму функції «show» у класі

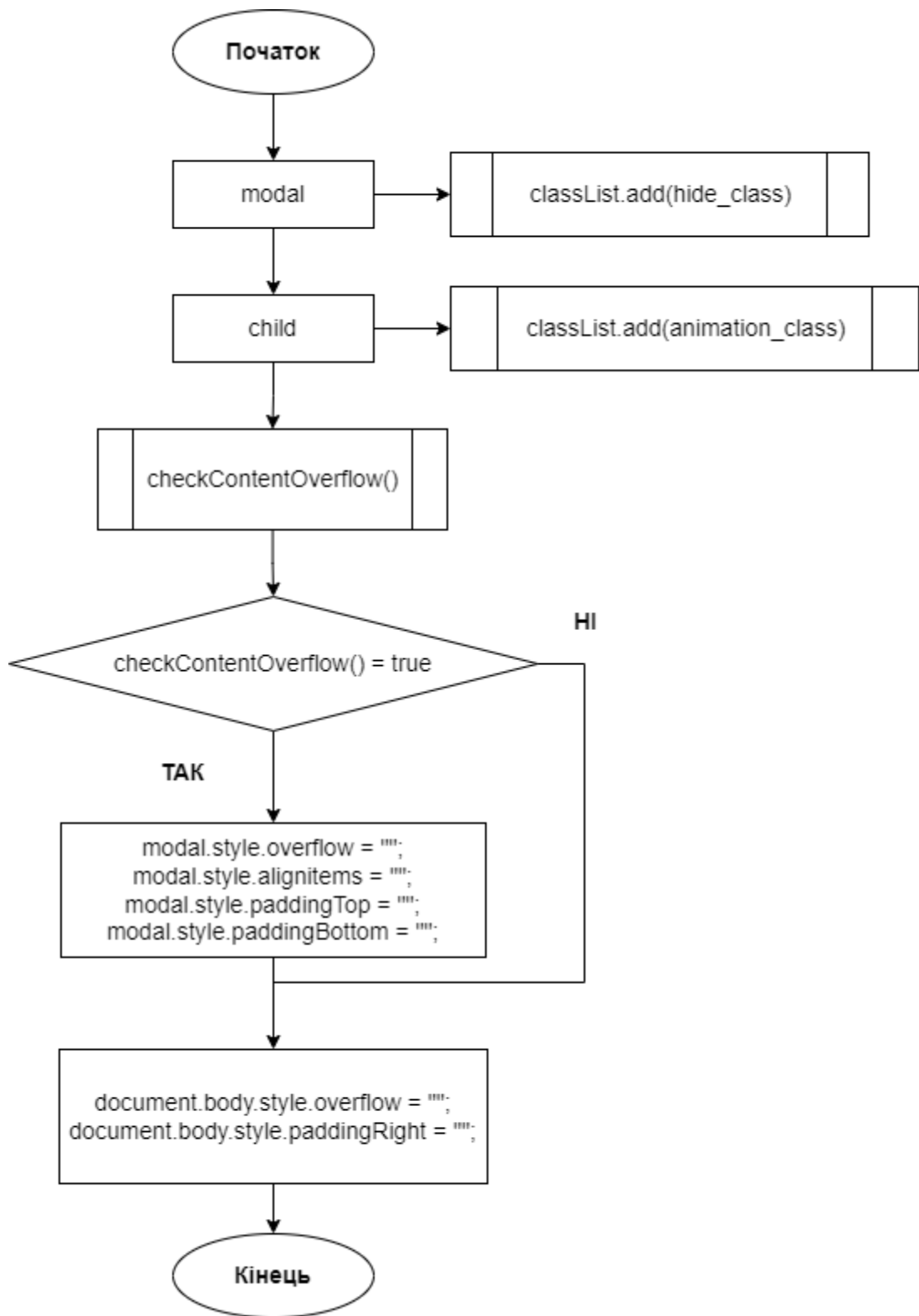


Рисунок 2.4.20 - Блок-схема алгоритму функції «hide» у класі

Тепер модуль готовий. Для того щоб його використовувати, необхідно лише скопіювати HTML-код в необхідну сторінку, підключити його стилі у загальному файлі «index.scss» та підключити його скрипт.

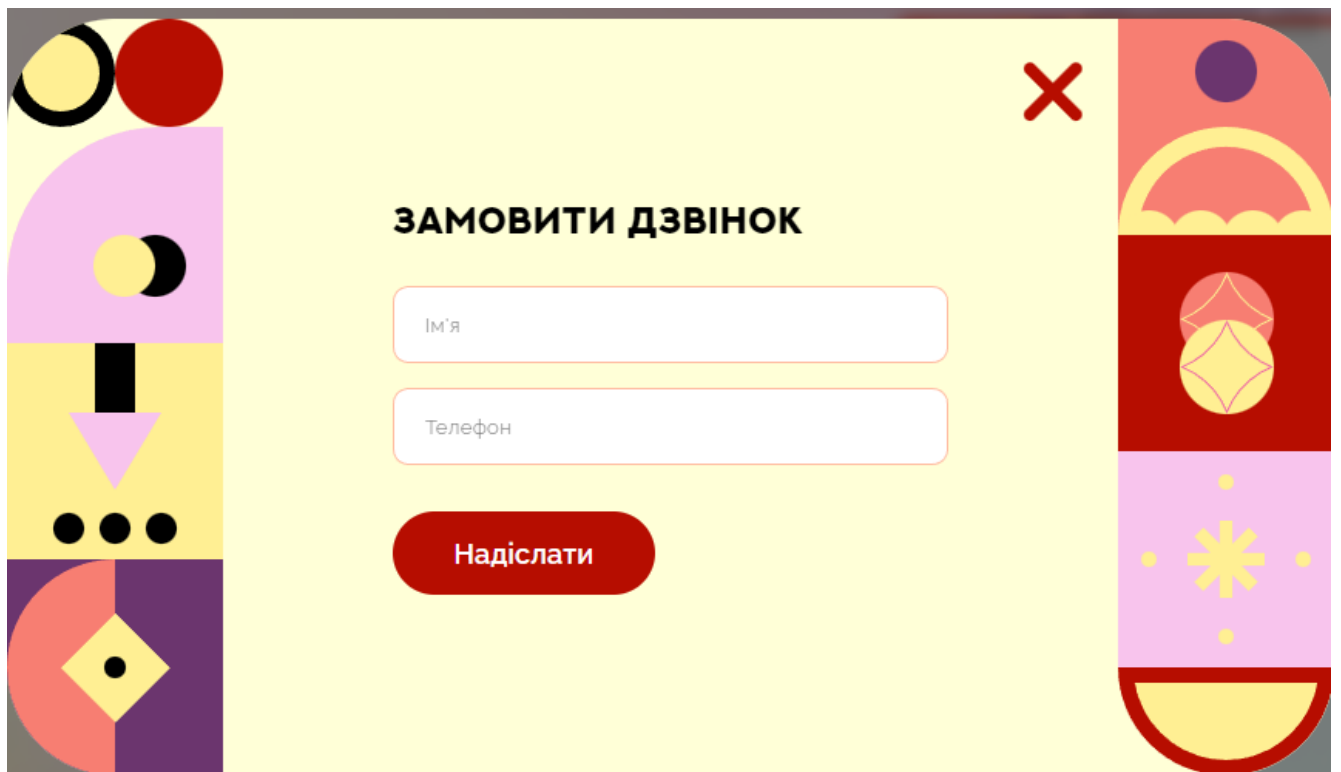


Рисунок 2.4.21 - Скриншот форми «Замовити дзвінок»

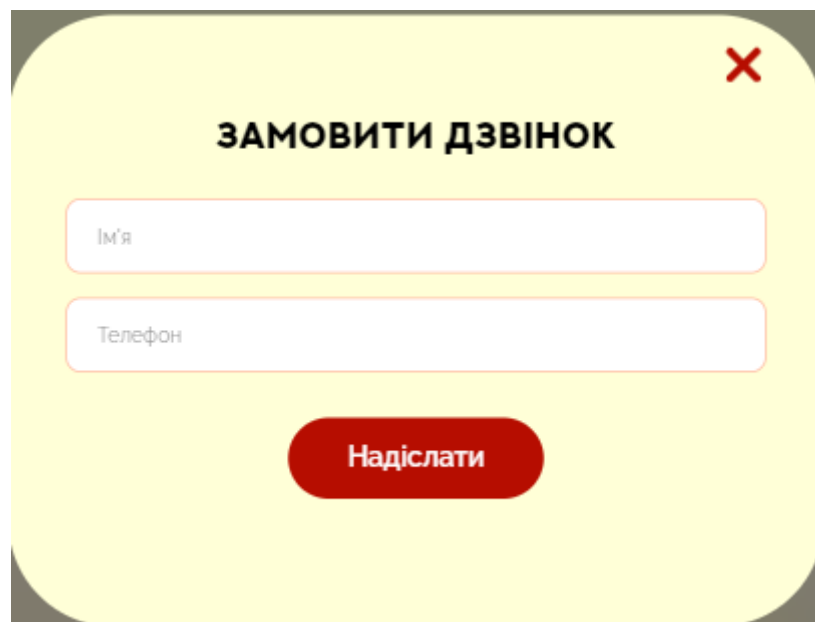


Рисунок 2.4.22- Скриншот мобільної версії форми «Замовити дзвінок»

					КГ05.05.002.00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		40

2.5 Висновки

Отже, було описано процес створення веб-сайту в цілому, та кожної сторінки. На прикладі форми «Замовити зворотній дзвінок» я продемонстрував процес розробки модуля, практичне використання описаних у першому розділі технологій.

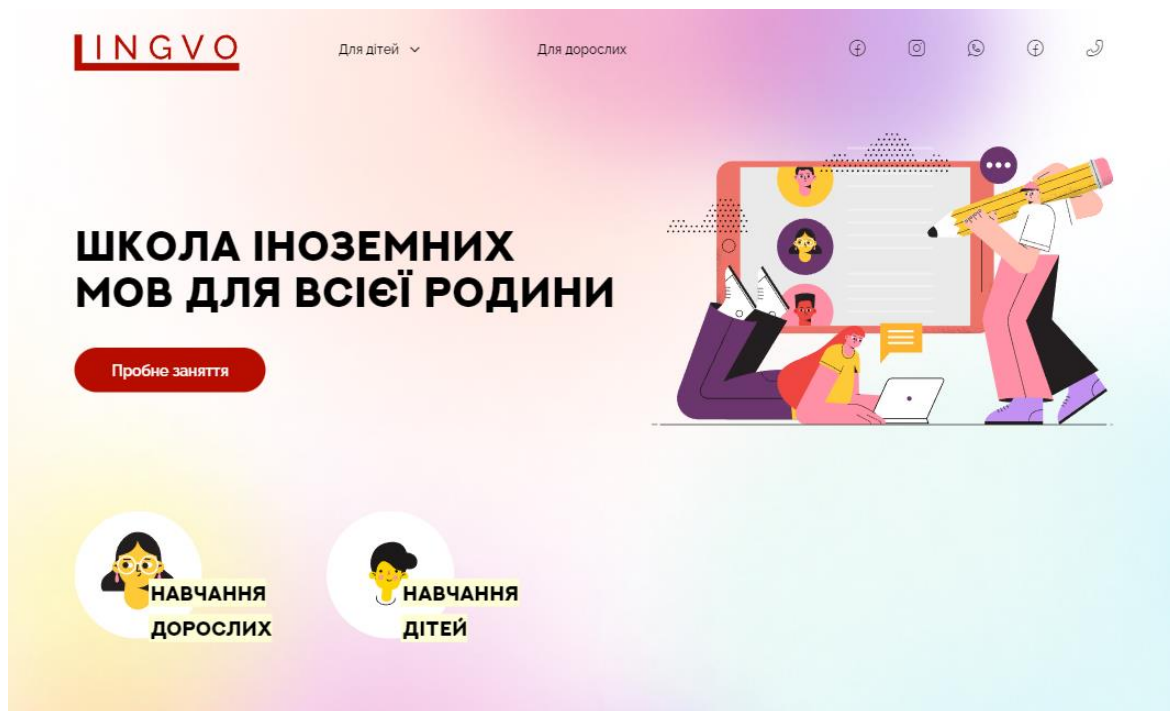


Рисунок 2.5.1 - Скриншот частини головної сторінки

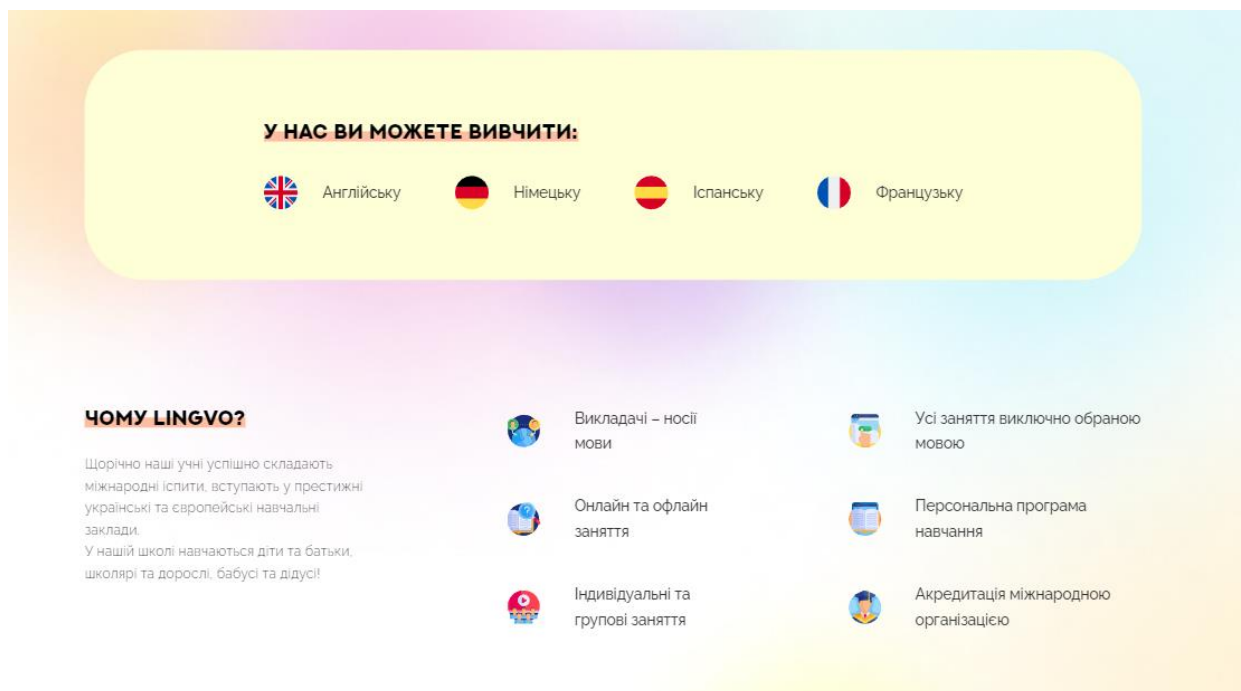


Рисунок 2.5.2 - Скриншот частини головної сторінки

					КГ05.05.002.00 ДП ПЗ	Арк.
						41
Ізм.	Лист	№ докум.	Підпис	Дата		

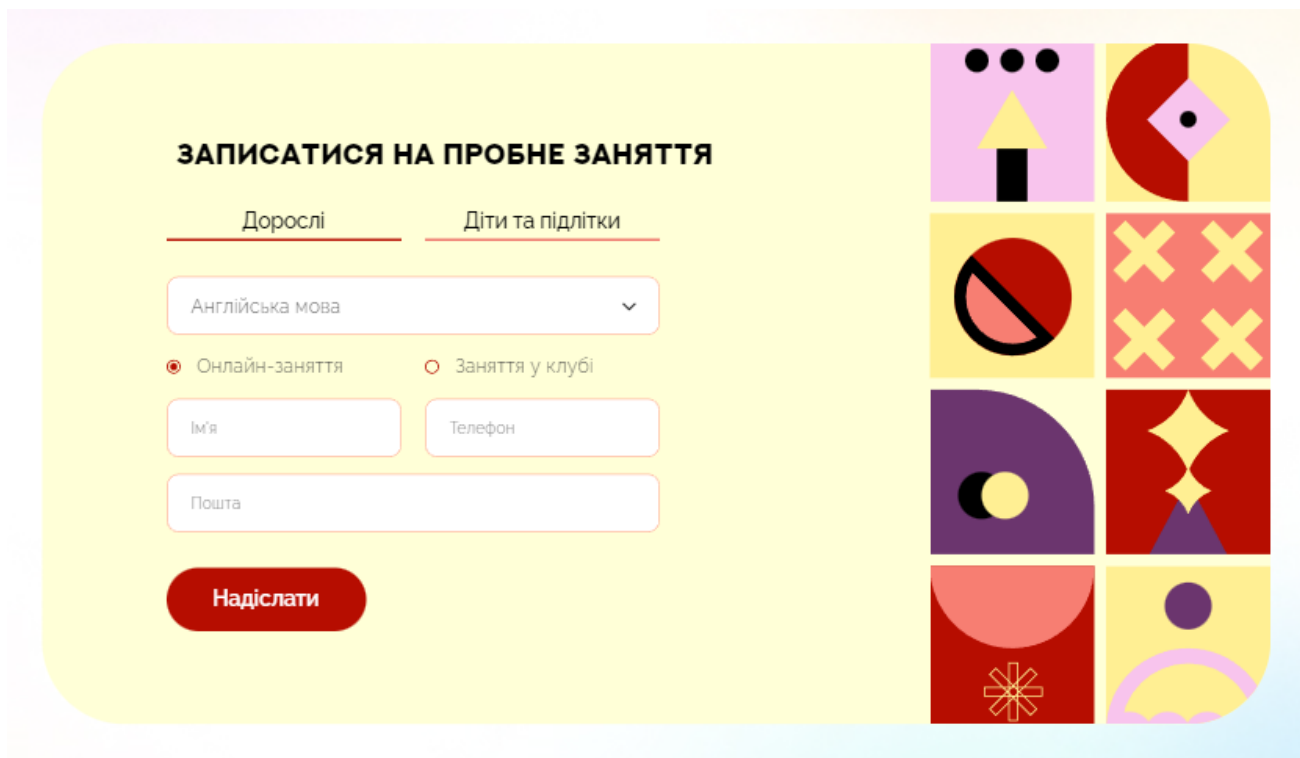


Рисунок 2.5.3 - Скриншот частини головної сторінки

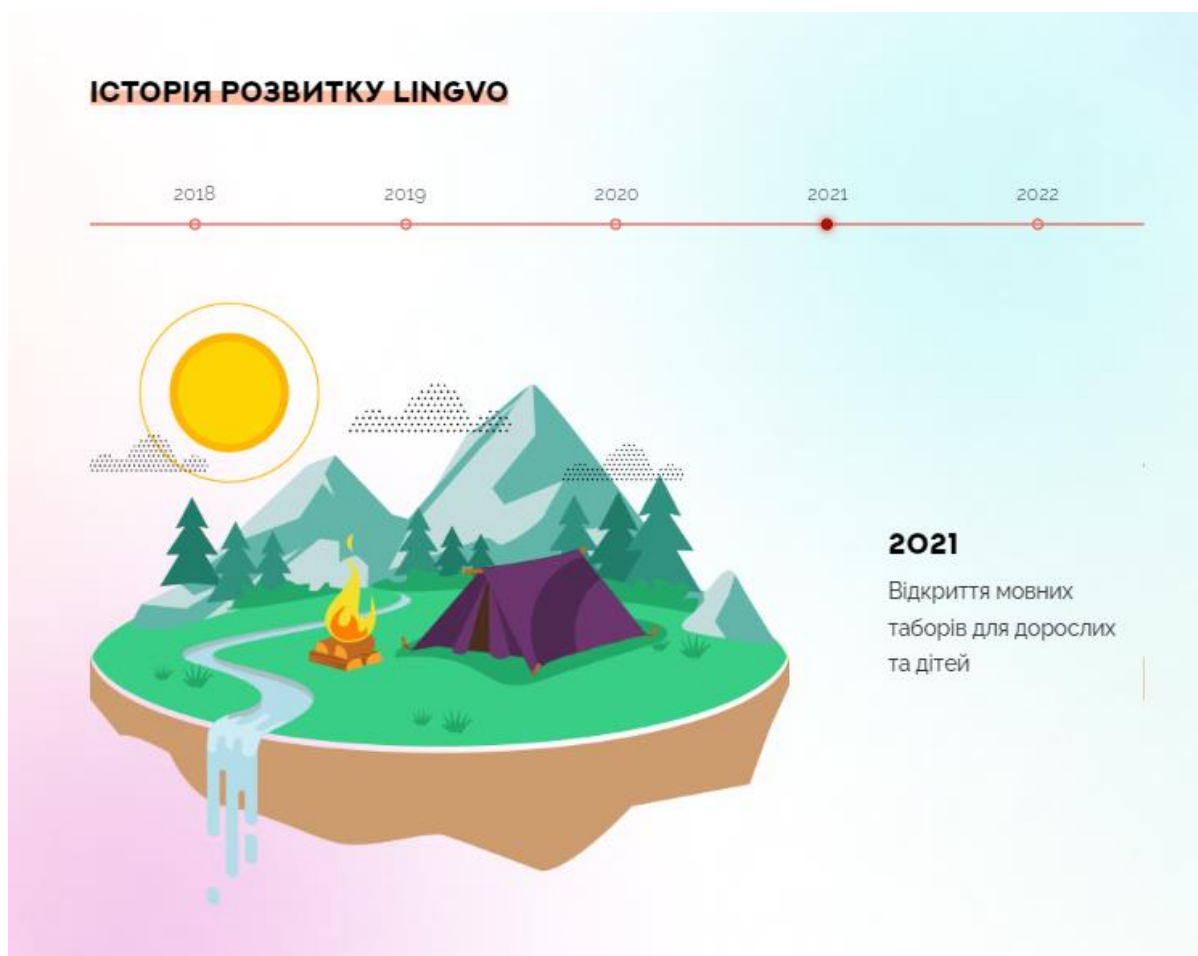


Рисунок 2.5.4 - Скриншот частини головної сторінки

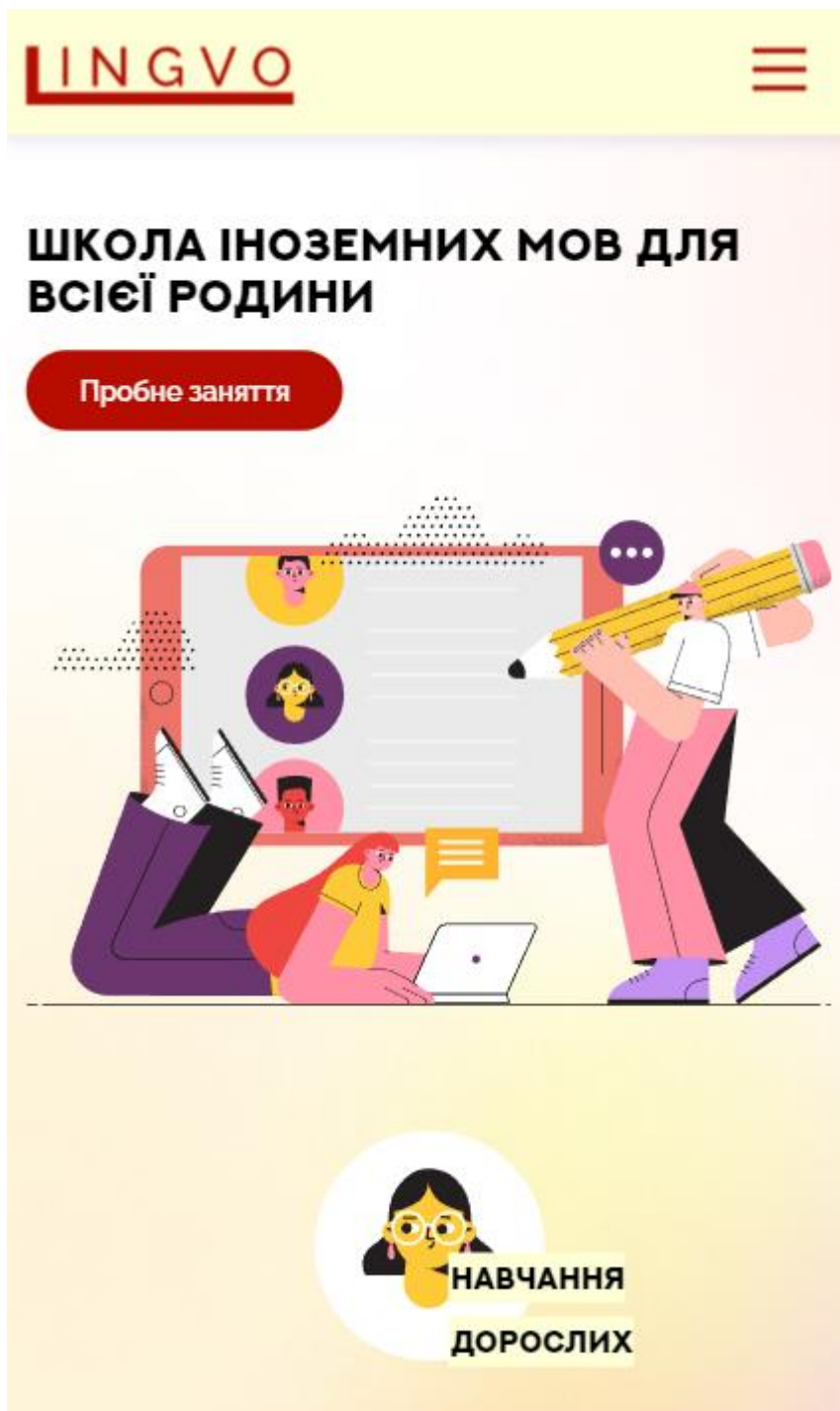


Рисунок 2.5.5 - Скриншот мобільної версії частини головної сторінки

					КГ05.05.002.00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		43



Рисунок 2.5.6 - Скриншот мобільної версії частини головної сторінки

3. ЕКОНОМІЧНИЙ РОЗРАХУНОК

ВСТУП

В дипломному проекті створений веб-сайт курсів іноземних мов для приватного навчального закладу із використанням технології HTML5.0+SCSS+CSS3+JS(ES6). Веб-сайт дає можливість користувачам обрати оптимальний пакет навчання іноземним мовам.

Як правило, веб-сайт створюється підприємством для надання основної інформації щодо його діяльності, характеристик товарів і послуг, що надаються, розміщення реклами, публікації контактів тощо. Основна мета створення веб-сайту курсів іноземних мов – це забезпечення мінімізації витрат.

При оцінці ефективності створеного сайту я виходив з того, що залежно від характеру ефекту, що досягається, визначаються наступні види ефективності сайту: економічна, функціональна та соціальна ефективність. Ці види ефективності в свою чергу залежать від технологічних, технічних, маркетингових та економічних показників веб-сайту. Між доходом від сайту і витратами на його створення існує певна залежність. Я також врахував, що на величину доходу від сайту, окрім інших чинників ефективності, впливають витрати на його оптимізацію та просування.

Розрахунок економічної ефективності розробки створеного web-сайту.

Загальні витрати (B_3) на створення сайту складаються з декількох параметрів:

$$B_3 = B_p + B_v + B_e ,$$

де B_p – витрати на розробку сайту;

B_v – витрати на впровадження сайту;

B_e – витрати на експлуатацію сайту;

					КГ05.05.003.00 ДП ПЗ	Арк.
						45
Ізм.	Лист	№ докум.	Підпис	Дата		

3.1 Витрати на розробку сайту

Витрати на розробку сайту (V_p) є одноразовими та складаються з вартості наступних видів робіт зі створення сайту:

1. Розробка дизайну сайту: розробка макетів дизайну для головної та внутрішньої сторінок сайту; логотипу.
2. Реалізація на сайті меню: звичайного та адаптивного для використання на мобільних пристроях.
3. Наповнення сайту інформацією: наповнення та форматування web-сторінок; обробка малюнків для публікації на web-сторінках, верстка (переклад в HTML-формат) web-сторінок
4. Програмна розробка сайту: створення програмного коду сайту, програмування динамічних елементів (анімаційних елементів).
5. Налаштування модулів: модуля каталогу товарів, модулів інтернет-замовлення послуг (каталог, замовлення), модуля e-mail форм і т.ін.

Для визначення витрат на розробку сайту (V_p) я розрахував оплату праці виконавців, безпосередньо притягнених до її виконання. Для реалізації проекту Web-системи використовуються наступні спеціалісти: веб-розробник, веб-дизайнер, HTML-верстальник, тестувальник, веб-інженер.

Для визначення трудомісткості розробки сайту (V_p) я склав план-графік по розробці web-сайту і тривалості виконання робіт. Розподіл робіт по етапах і видах виконавців наведено в таблиці 2.1.

Таблиця 2.1 – План-графік по розробці веб-сайту курсів іноземних мов

№	Назва етапу	Час виконання (годин)	Посада виконавця
1	Аналіз вихідних даних	4	Веб-розробник
2	Створення структури меню веб-сайту	4	Веб-розробник

3	Визначення основного функціоналу проекту	4	Веб-розробник
4	Алгоритм створення веб-сайту	4	Веб-розробник
5	Розробка прототипу веб-сайту в Figma	6	Веб-дизайнер
6	Верстка HTML- коду веб-сайту	14	HTML-верстальник
7	Написання стилів веб-сайту	6	HTML-верстальник
8	Написання програмного коду веб-сайту	12	HTML-верстальник
9	Відладка і тестування роботи веб-сайту	4	Тестувальник
10	Завантаження створеного веб-сайту на хостинг провайдера	2	Веб-інженер
ВСЬОГО:		60	

Розрахунок трудомісткості я здійснив в наступній послідовності:

1. Склав перелік всіх етапів і видів робіт, які необхідно виконати в ході даної розробки. Після узгодження з керівником проекту допущено виключення, доповнення, об'єднання окремих етапів і видів робіт;
2. По кожному виду робіт визначив кваліфікаційний рівень виконавців. В разі виконання однієї роботи виконавцями різної кваліфікації, робота розподілена на ряд паралельних конкретних робіт для кожної категорії виконавця.

В умовах відсутності нормативної бази тривалість виконання окремих робіт розраховував на основі вірогідних оцінок робіт, що задаються виконавцями.

Розмір заробітної плати розраховував виходячи з чисельності різних категорій виконавців, трудомісткості, що витрачається ними на виконання різних видів робіт, а також їх середньої ставки за годину (яка дорівнює мінімальній ставці встановленій Відповідно до «Закону про Державний бюджет України»).

При визначенні вартості виконуваних робіт я орієнтувався на мінімальну заробітну плату, встановлену Відповідно до «Закону про Державний бюджет України» (станом на 1.01 поточного року), враховуючи кваліфікацію виконавців,

					КГ05.05.003.00 ДП ПЗ	Арк.
						47
Ізм.	Лист	№ докум.	Підпис	Дата		

Витрати на заробітну плату приведені в таблиці 3.2. (6500 грн – з/п, 39,60 грн - в годину).

Таблиця 2.2 – Витрати на заробітну плату

№	Персонал	Етапи розробки	Кількість робочих годин (або днів)	Погодинна ставка (або денна ставка), грн.	Заробітна плата, грн.
1	Веб-розробник	1-4	16	39,60	633,60
2	Веб-дизайнер	5	6	39,60	237,60
3	HTML-верстальник	6-8	32	39,60	1267,20
4	Тестувальник	9	4	39,60	158,40
5	Веб-інженер	10	2	39,60	79,20
ВСЬОГО:					$V_{зп}=2376,00$

До складу витрат на оплату праці також включаються податки, збори і інші обов'язкові платежі, встановлені системою оподаткування що діє. Розмір єдиного соціального внеску складає 22% від заробітної плати, я розрахував за наступною формулою:

$$V_{ссв} = V_{зп} \times 0,22$$

$$V_{ссв}=2376,00 \times 0,22=522,72 \text{ грн.}$$

Загальні витрати (V_p) на розробку веб-сайту розраховуються як сума витрат на заробітну плату праці персоналу ($V_{зп}$) та єдиного соціального внеску ($V_{ссв}$):

$$V_p = V_{зп} + V_{ссв}$$

$$V_p=2376,00+522,72=2898,72 \text{ грн}$$

3.2 Витрати на впровадження та експлуатацію сайту

Витрати на впровадження сайту (В_в) складаються з двох складових:

- витрати на реєстрацію доменного імені на 1 рік (od.ua)- 274 грн (за даними Tenet);
- витрати на реєстрацію в пошукових системах (В_{в2}) Google – 2400 грн

Витрати на експлуатацію сайту (В_е) включають вартість робіт з підтримки сайту в робочому стані і вартість послуг по продовженню доменного імені на 1 рік.

Роботи по підтримці сайту в робочому стані включають в себе:

1. Оновлення даних на сайті;
2. Видалення застарілої інформації з сайту;
3. Додавання потрібної інформації на сайт;
4. Налаштування параметрів сервера хостингу;
5. Моніторинг роботи сервера хостингу;
6. Забезпечення щомісячного захисту сайту;

Підтримка сайту в робочому стані здійснює веб-розробник.

У таблиці 2.3 я визначив постійні витрати як суму витрат на впровадження та експлуатацію сайту протягом року.

Таблиця 3.3 – Постійні витрати

№	Стаття витрат	Вартість за рік, грн.
1	Веб-розробник (обслуговування веб-сайту) 1 год *249дн*39,60 грн	9860,40
2	Домене ім'я (od.ua) (дані Tenet)	274,00
3	Хостинг (дані Tenet)	1440,00
4	Пошукова оптимізація сайту	1200,00
5	Пошукове просування сайту	1200,00

6	Доступ до мережі інтернет (дані Tenet)	285*12=3420,00
Всього:		$V_{\text{пост}} = 17394,40$

Загальні витрати (V_3) на розробку, впровадження та експлуатацію веб-сайту я розраховував за наступною формулою:

$$V_3 = V_p + (V_b + V_e) = V_p + V_{\text{пост}}$$

$$V_3 = 2898,72 + 17394,40 = 20293,12 \text{ грн}$$

3.3 Розрахунок економічної ефективності

Економічна ефективність за рік (E_p) я визначив як сукупність коштів, вивільнених за рахунок впровадження сайту:

$$E_p = E_{p1} + E_{p2}$$

Вивільнення коштів я досягнув за рахунок:

E_{p1} - скорочення коштів, що витрачаються на розміщення реклами (а саме дизайн, виготовлення та розповсюдження рекламних буклетів)

$$E_{p1} = 1200 * 12 = 14400,00 \text{ грн.}$$

(1200 грн / місяць - вартість виготовлення 1-2 тисяч рекламних буклетів + оплата дизайну + оплата розповсюдження,)

E_{p2} - скорочення засобів за рахунок рішення окремих бізнес-завдань за допомогою сайту, а саме кількість робочих годин на зменшилась на одну годину на день для одного робітника, що дозволить перерозподілити вивільнений робочий час на виконання інших видів робіт.

$$E_{p2} = 39,6 * 249 \text{ днів} = 9860,40 \text{ грн}$$

(Форма автоматичної оплати може зберегти приблизно 1 робочу годину на 1 робочий день)

					КГ05.05.003.00 ДП ПЗ	Арк.
						50
Ізм.	Лист	№ докум.	Підпис	Дата		

$E_p = 14400 + 9860,40 = 24260,40$ грн

Коефіцієнт економічної ефективності за перший рік я вирахував таким чином:

$$K_e = E_T / B_z$$
$$24260,40 / 20293,12 = 1,20$$

3.4 Функціональна та соціальна ефективність

Функціональна ефективність проявляється:

- в забезпеченні повноти, точності і доступності інформації про організацію, її діяльність, послуги у будь-який слушний для користувача час доби;
- в оптимізації технологічних процесів (процесу обробки оплати послуг)

Соціальна ефективність, або соціальний ефект виражається в наступному:

1. Розширення складу засобів просування товарів і послуг, що надаються.
2. Інформування найбільшого числа зацікавлених осіб про діяльність курсів іноземних мов та переліку послуг.
3. Оптимізація рекламної діяльності організації.
4. Пошук нових клієнтів і ділових партнерів.
5. Формування позитивного іміджу організації.

3.5 Висновки

Я розрахував необхідні витрати на розробку та впровадження веб-сайту, його експлуатацію. Зазначив та розрахував економічну ефективність сайту, з чого можна зробити висновок, що розробка та подальша експлуатація сайту здатна принести фінансову вигоду власнику. Також зазначив функціональну та соціальну ефективність веб-сайту.

					КГ05.05.003.00 ДП ПЗ	Арк.
						51
Ізм.	Лист	№ докум.	Підпис	Дата		

4. ОХОРОНА ПРАЦІ

ВСТУП

Безпека праці, як галузь практичної діяльності, спрямована на створення безпечних і нешкідливих умов праці. На сучасному етапі розвитку виробництва вона набуває все більш важливого значення. Створення безпечних і нешкідливих умов праці на виробництві вимагає значних матеріальних витрат, впровадження знань і рішень науково-дослідних робіт в галузі охорони праці. Тому роль знань з питань охорони праці інженерно-технічними працівниками має дуже велике значення. Умови праці на робочому місці веб-розробника мають відповідати вимогам безпеки праці, у відповідності з нормативно-правовим актом з охорони праці (НПАОП 0.00-7.15-18) «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» , а також НПАОП 0.00-1.28-10 "Правила охорони праці під час експлуатації електронно-обчислювальних машин", затверджені наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду від 26.03.2010 № 65. Розробка веб-сайту курсів іноземних мов виконується веб-розробником, робочим місцем якого є персональний комп'ютер в умовах офісу або домашніх умовах. Нижче буде розглянуто умови праці та забезпечення безпеки праці при виконанні основних видів робіт на робочому місці веб-розробника.

Аналіз шкідливих та небезпечних факторів виробничого середовища.

Під час праці на веб-розробника можуть впливати наступні фактори:

- підвищена або знижена температура повітря робочої зони;
- підвищений рівень шуму на робочому місці;
- підвищена або знижена вологість повітря;
- підвищена або знижена рухливість повітря;

					КГ05.05.004.00 ДП ПЗ	Арк.
						52
Ізм.	Лист	№ докум.	Підпис	Дата		

- підвищена або знижена іонізація повітря;
- підвищений рівень вібрації;
- підвищений рівень електромагнітних випромінювань;
- відсутність або недостача природного освітлення;
- недостатня освітленість робочої зони;
- нервово-психічні перевантаження;

4.1 Розробка заходів з охорони праці

Для нормальної життєдіяльності людини в умовах виробництва треба створити умови, які б дали змогу їй плідно працювати не перевтомлюючись та зберігати своє здоров'я. Для цього треба, щоб енергетичні витрати при праці компенсувалися відпочинком та умови оточуючого середовища відповідали вимогам безпеки праці у відповідності з нормативно-правовими актами.

Вимоги до приміщення

Приміщення, яке розглядається, має довжину 5м, ширину 3м, висоту стелі 2.75м, загальну площу 15м². У даному приміщенні працює один веб-розробник, тобто у приміщенні одне робоче місце. Згідно з наказом «Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин», розміщення робочих місць у підвалах заборонено, площа одного робочого місця має становити не менш ніж 6.0м², тобто, приміщення, яке розглядається відповідає цим критеріям. Також приміщення повинно мати як штучне, так і природне освітлення. Воно має бути забезпечено системою опалення, кондиціонування повітря або припливно-витяжною вентиляцією. Поверхня підлоги має бути рівною, неслизькою, з антистатичними властивостями.

					КГ05.05.004.00 ДП ПЗ	Арк.
						53
Ізм.	Лист	№ докум.	Підпис	Дата		

Організація робочого місця

Конструкція робочого місця користувача ЕОМ і ПЕОМ з ВДТ має забезпечити підтримання оптимальної робочої пози. Робочі місця з ВДТ слід так розташовувати відносно світових прорізів, щоб природне світло падало збоку переважно зліва. При розміщенні робочих столів з ВДТ слід дотримувати такі відстані між бічними поверхнями ВДТ 1,2 м, відстань від тильної поверхні одного ВДТ до екрана іншого ВДТ - 2,5 м. Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів. Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600 мм, завширшки не менше ніж 500 мм, завглибшки (на рівні колін) не менше ніж 450 мм, на рівні простягнутої ноги - ніж 650 мм. Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край - заокругленим. Регулювання за кожним із параметрів має здійснюватися незалежно, легко і надійно фіксуватися. Шаг регулювання елементів стільця має становити: для лінійних розмірів -15...20 мм, для кутових 2...5 град. Зусилля регулювання має не перевищувати 20 Н.

Освітлення робочого місця

У приміщенні, яке розглядається, використовується змішане освітлення (комбінація штучного та природного освітлення). Використовуються лампи типу ЛД, обов'язково із розсіювачами та екрануючими сітками. Робоче місце розташоване так, щоб уникнути попадання в очі прямого сонячного світла.

Згідно з ДСанПІН 3.3.2.007-98, штучне освітлення в приміщеннях з робочими місцями, обладнаними ВДТ ЕОМ та ПЕОМ, має здійснюватися системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, у разі переважної роботи з документами, допускається

					КГ05.05.004.00 ДП ПЗ	Арк.
						54
Ізм.	Лист	№ докум.	Підпис	Дата		

застосування системи комбінованого освітлення (крім системи загального освітлення додатково встановлюються світильники місцевого освітлення). Зазначення освітлення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300-500 лк. Якщо ці значення освітленості неможливо забезпечити системою загального освітлення, допускається використовувати місцева освітлення. При цьому світильники місцевого освітлення слід встановлювати таким чином, щоб не створювати бліків на поверхні екрана, а освітленість екрана має не перевищувати 300 лк. Як джерела світла в разі штучного освітлення мають застосовуватись переважно люмінесцентні лампи типу ЛБ. У разі влаштування відбитого освітлення у виробничих та адміністративно-громадських приміщеннях допускається застосування металогалогенних ламп потужністю 250 Вт. Допускається застосування ламп розжарювання у світильниках місцевого освітлення.

Мікроклімат

У приміщенні, яке розглядається, використовується загальне опалення, система кондиціонування. Для підтримання оптимальної вологості повітря використовуються прилади зволоження повітря.

Згідно з ГОСТ 12.1.005-88 СН 4088-86, у виробничих приміщеннях на робочих місцях з ВДТ мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря (Таблиця 1).

Пора року	Категорія робіт	Температура повітря, град.С	Відносна вологість повітря	Швидкість руху повітря м/с
		оптимальна	оптимальна	оптимальна
Холодна	Легка-1 а	22-24	40-60	0.1

					КГ05.05.004.00 ДП ПЗ	Арк.
						55
Ізм.	Лист	№ докум.	Підпис	Дата		

	Легка-1 б	21-23	40-60	0.1
Тепла	Легка-1 а	23-25	40-60	0.1
	Легка-1 б	22-24	40-60	0.2

Таблиця 1.

Іонізація повітря

Для підтримки оптимальної кількості іонів в повітрі, у приміщенні використовується іонізатор повітря.

Рівні позитивних і негативних іонів у повітрі приміщень з ВДТ мають відповідати санітарно-гігієнічним нормам N 2152-80 (Таблиця 2)

Рівні	Кількість іонів в 1 см куб. повітря	
	N+	N-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

Таблиця 2.

4.2 Пожежна безпека

Пожежна безпека – стан об’єкта, при якому з регламентованою ймовірністю виключається можливість виникнення та розвиток пожежі і впливу на людей її небезпечних факторів, а також забезпечується захист матеріальних цінностей. Причинами пожеж та вибухів на підприємстві є порушення правил і норм пожежної безпеки, невиконання Закону “Про пожежну безпеку”. Можливими причинами виникнення пожежі в приміщенні є:

- коротке замикання проводки;

- неправильне користування побутовими електроприладами;
- не дотримання умов протипожежної безпеки;

У зв'язку з цим відповідно до ПУЕ необхідно передбачити наступні заходи щодо пожежної безпеки:

- ретельна ізоляція всіх струмоведучих провідників до робочих місць;
- періодичний огляд і перевірка ізоляції; суворе дотримання норм протипожежної безпеки на робочому місці.

У випадку виникнення пожежі необхідно відключити електроживлення, викликати по телефоні пожежну команду, евакуювати людей із приміщення відповідно до плану евакуації і приступити до ліквідації пожежі вогнегасниками. При наявності невеликого вогнища полум'я, можна скористатися підручними засобами з метою припинення доступу повітря до об'єкта загоряння.

Для гасіння пожеж на робочому місці використовують вуглекислотні та порошкові вогнегасники. Наявність первинних засобів пожежогасіння і вогнегасників, їхня кількість і зміст відповідає вимогам ГОСТ 12.4.009-75 і ISO3941-77.

У приміщенні виконуються усі вимоги по пожежній безпеці відповідно до вимог НАПБ А.0.001-95 “Правила пожежної безпеки в Україні”. У приміщенні також мається план евакуації на випадок виникнення пожежі. Час евакуації відповідає вимогам СНиП 2.01.02-85, а максимальне видалення робочих місць від евакуаційних виходів відповідає СНиП 2.09.02-85.

4.3 Висновки

Вивчення й вирішення проблем, пов'язаних із забезпеченням здорових і безпечних умов, за яких відбувається праця людини - одне з найважливіших завдань у системах виробництва. Дослідження й виявлення можливих причин виробничих нещасних випадків, професійних захворювань, аварій, пожеж і розробка заходів, спрямованих на усунення цих причин дозволяють створити безпечні умови для праці людини. Комфортні й безпечні умови праці - один з основних факторів, що впливає на здоров'я працівників та продуктивність їхньої праці.

					<i>КГ05.05.004.00 ДП ПЗ</i>	Арк.
						57
<i>Ізм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВИСНОВОК

В даному дипломному проекті розглянуто основні етапи розробки веб-сайту курсів іноземних мов. У проекті розроблено наступне:

- 1) обґрунтування вибору технологій для розробки;
- 2) алгоритм створення усього сайту, кожної сторінки, кожного блоку.
- 3) розглянуто основні питання розробки та верстки веб-сторінки із використанням технології HTML5.0+CSS+JS(ES6).

В дипломному проекті також проведено необхідні економічні розрахунки та розглянуто заходи з охорони праці.

Запропонований проект може бути розміщений як самостійна сторінка у мережі Інтернет.

					КГ05.05.000.00 ДП ПЗ	Арк.
						58
Ізм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. DOM Living Standart [Електронний ресурс] Режим доступу до ресурсу: <https://dom.spec.whatwg.org/> (дата звернення 20.05.2022)
2. HTML Living Standart [Електронний ресурс] Режим доступу до ресурсу: <https://html.spec.whatwg.org/> (дата звернення 20.05.2022)
3. Compatibility Living Standart [Електронний ресурс] Режим доступу до ресурсу: <https://compat.spec.whatwg.org/> (дата звернення 20.05.2022)
4. MDN Web Docks [Електронний ресурс] Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/>
5. Сучасний підручник з JavaScript [Електронний ресурс] Режим доступу до ресурсу: <https://uk.javascript.info/>
6. Sass: Documentation [Електронний ресурс] Режим доступу до ресурсу: <https://sass-lang.com/documentation/>
7. Quick start / Methodology / BEM [Електронний ресурс] Режим доступу до ресурсу: <https://en.bem.info/methodology/quick-start/>
8. Git - Reference [Електронний ресурс] Режим доступу до ресурсу: <https://git-scm.com/docs>
9. Scott Chacon, Ben Straub. ProGit. apress 2021
10. Бойчик І. М. Економіка підприємства : навчальний посібник для студентів економічних спеціальностей вищих навчальних закладів I-IV рівнів акредитації. Третє видання, випр. і доп. / І. М. Бойчик, П. С. Харів., М. І. Холчан, Ю. В. Піча. – К. : Каравела, 2016. – 328 с.
11. ДСанПіН 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих приміщень».
12. ДСанПіН 2.3.6.037-99 «Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку»

					КГ05.05.000.00 ДП ПЗ	Арк.
						59
Ізм.	Лист	№ докум.	Підпис	Дата		

ДОДАТОК



ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Розробка веб-сайту курсів іноземних мов

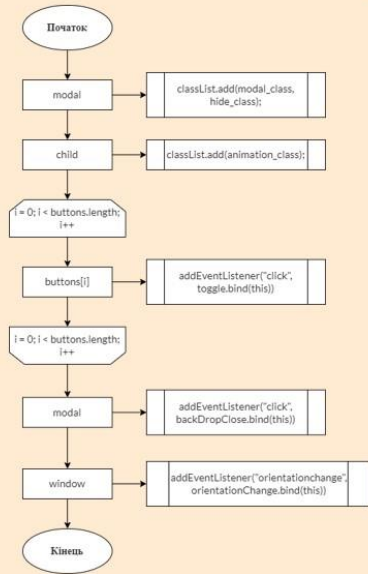
Виконав: Грабовий О.В., 4КГ-05



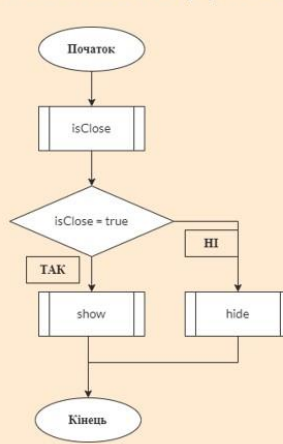
СТРУКТУРНА СХЕМА ГОЛОВНОЇ СТОРІНКИ



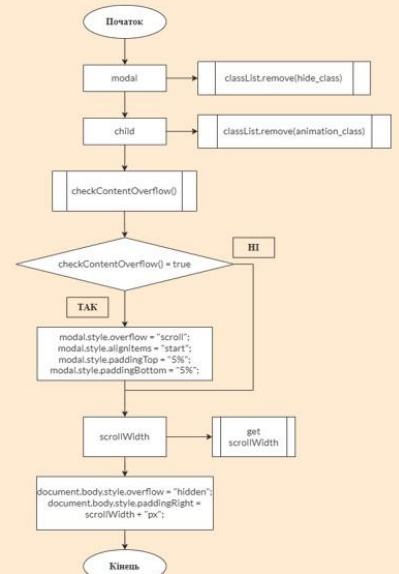
БЛОК-СХЕМА АЛГОРИТМУ ФУНКЦІЙ ФОРМИ «ЗАМОВИТИ ДЗВІНОК»



Функція «setup»

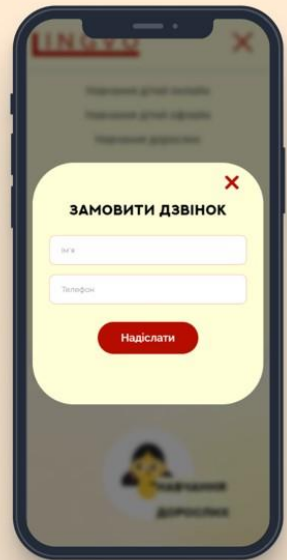
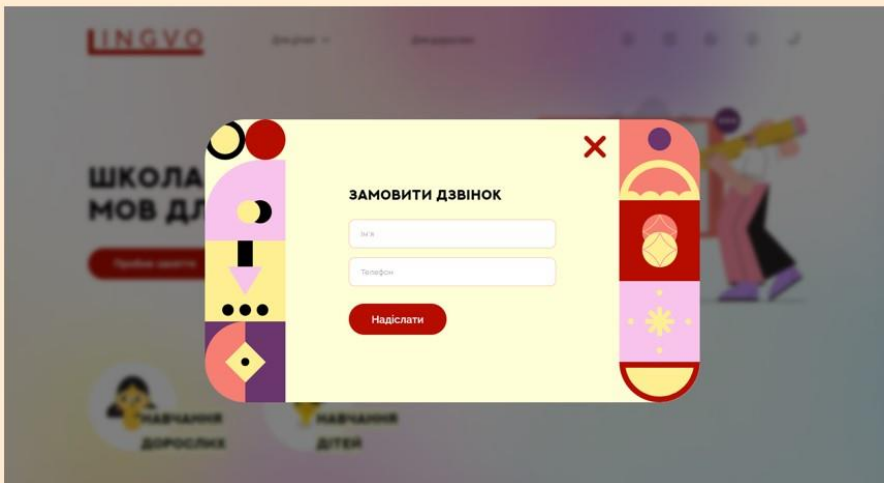


Функція «toggle»

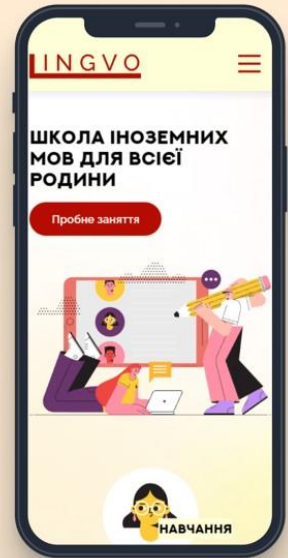
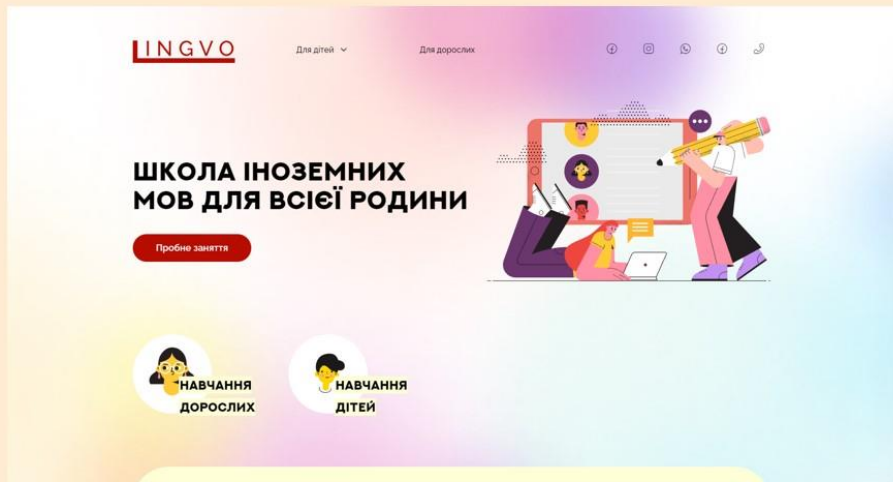


Функція «show»

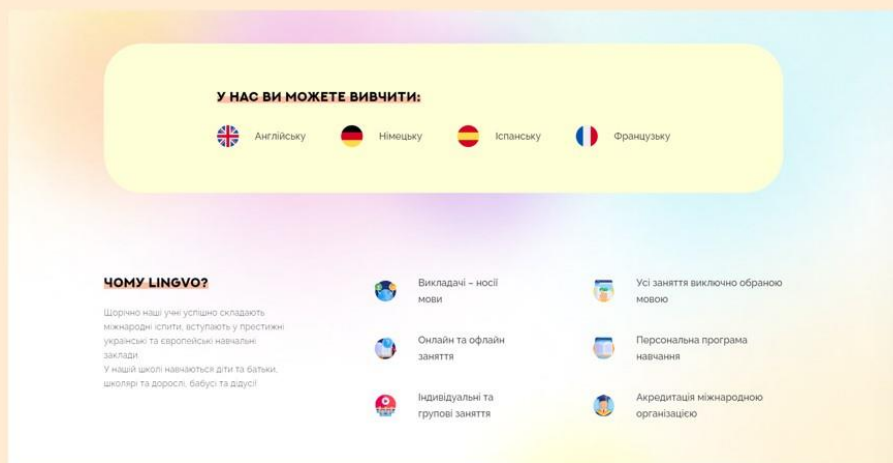
СКРИНШОТИ ФОРМИ «ЗАМОВИТИ ДЗВІНОК»



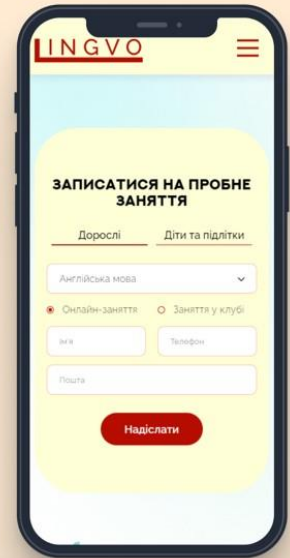
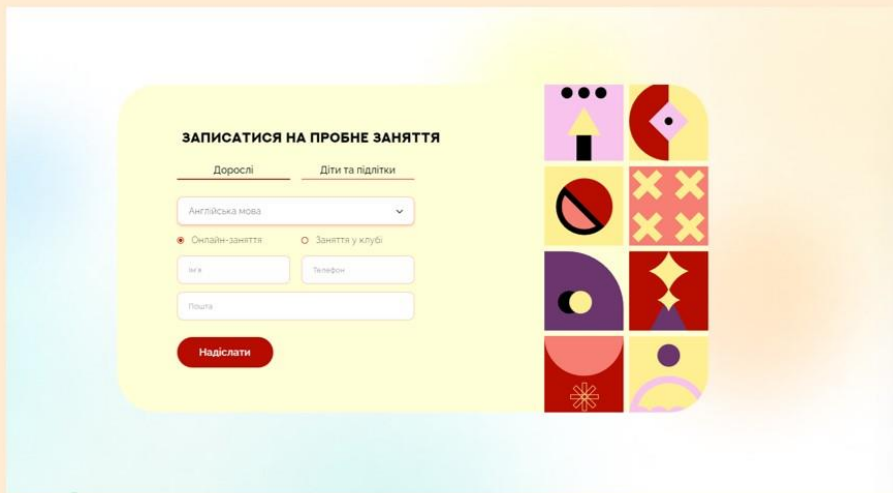
СКРИНШОТИ ГОЛОВНОЇ СТОРІНКИ



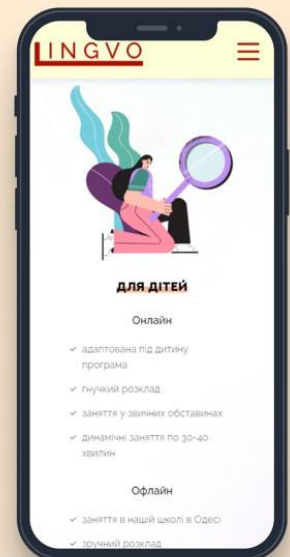
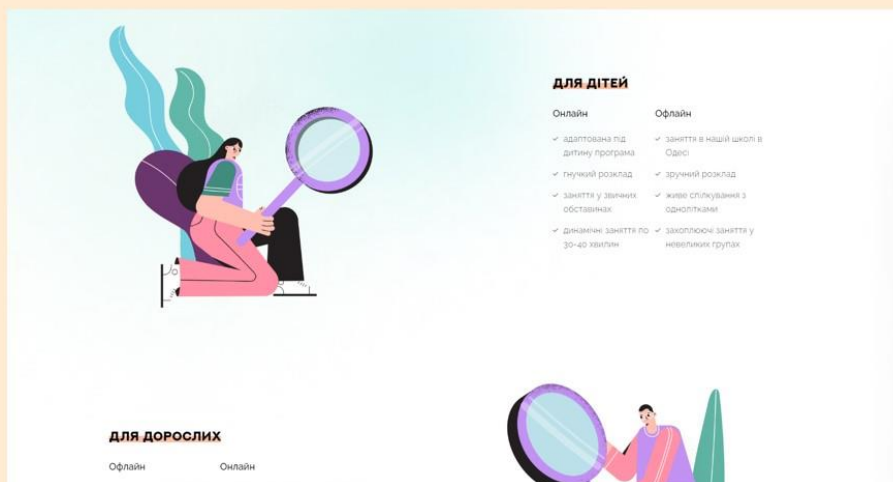
СКРИНШОТИ ГОЛОВНОЇ СТОРІНКИ



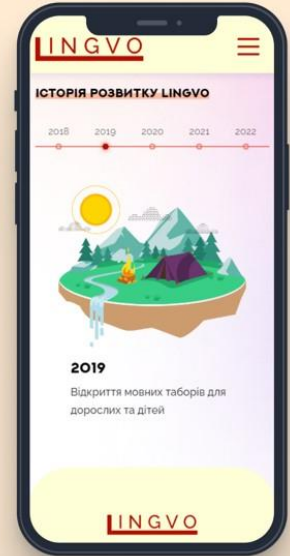
СКРИНШОТИ ГОЛОВНОЇ СТОРІНКИ



СКРИНШОТИ ГОЛОВНОЇ СТОРІНКИ

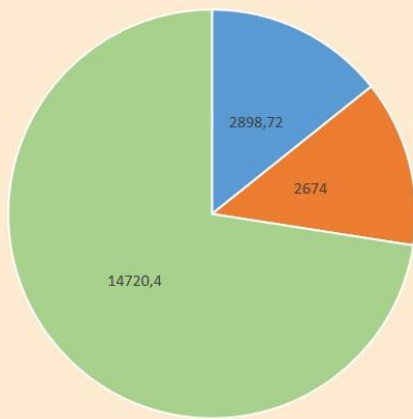


СКРИНШОТИ ГОЛОВНОЇ СТОРІНКИ



ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ

Витрати на розробку та впровадження



- Витрати на розробку (грн)
- Витрати на впровадження (грн)
- Витрати на експлуатацію (за рік, грн)

Економічна ефективність



- Загальні витрати на розробку (грн)
- Вивільнені кошти (грн)