

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»
РОЗРОБКА САЙТУ ДИТЯЧОГО САДОЧКА №28 «ВУЛИК»
DEVELOPMENT OF THE WEBSITE OF KINDERGARDEN
NO. 28 «VULIK»

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ІПЗ-42

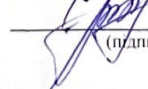
Циплінецька Дарія Сергіївна


(підпис)

Керівник:

к.т.н., доцент

Самчук Людмила Миколаївна


(підпис)

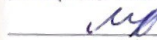
Кваліфікаційну роботу
допущено до захисту

«8» 06 2024 р.

к.т.н., доцент

Гарант освітньої програми:

Ліщина Наталія Миколаївна


(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення
Ступінь вищої освіти: бакалавр
Галузь знань: 12 Інформаційні технології
Спеціальність: 121 Інженерія програмного забезпечення
Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
М. М. Микитюк
« 2 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Циплінський Дарій Сергійович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: розробка сайту дитячого садочка №28 «Вулик»

Керівник роботи: к.т.н., доцент, Самчук Людмила Миколаївна

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

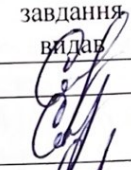
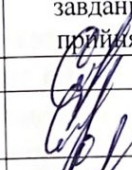
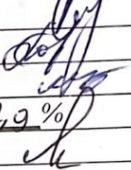
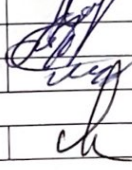
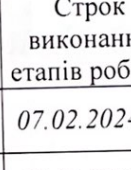
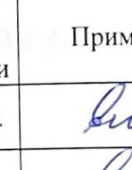
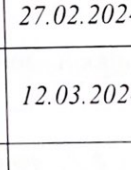
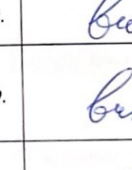
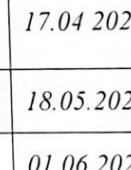
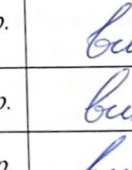
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р.

3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.

5. Перелік графічного матеріалу: 26 рисунки; 6 таблиць.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Самчук Л. М.		
Специфікації вимог до розробленої системи	Самчук Л. М.		
Розробка об'єкта проєктування	Самчук Л. М.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		2,9 %	
Академічна доброчесність	Яцук А. А.		

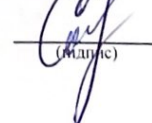
7. Дата видачі завдання «2» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	07.02.2024 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	27.02.2024 р.	
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	12.03.2024 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проєктування та проєктування бази даних	17.04 2024 р.	
5	Практична реалізація об'єкта проєктування та розробка бази даних	18.05.2024 р.	
6	Тестування та налагодження об'єкта проєктування	01.06.2024 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	05.06.2024 р.	

Здобувач вищої освіти

Керівник кваліфікаційної роботи


(підпис)

(підпис)

Циплінська Д. С.
(прізвище, ініціали)

Самчук Л. М.
(прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

Рецензія
на кваліфікаційну роботу бакалавра

Здобувач вищої освіти:
Циплінська Дарія Сергіївна

Тема: Розробка сайту дитячого садочка №28 «Вулик»

Коротка характеристика кваліфікаційної роботи:

Кваліфікаційна робота присвячена розробці сайту для дитячого садочка №28 «Вулик». Ця тема є доволі актуальною через постійний абгрейд освітніх технологій та зростаючу потребу у цифрових ресурсах. Вона надає можливість всім охочим включаючи батьків, дізнаватися про актуальні новини, поради, навчальні програми тощо без зайвих труднощів.

Самостійні розробки і пропозиції автора:

Сайт розроблено самостійно за допомогою мови програмування JavaScript, та інших інструментів, включаючи різноманітні бібліотеки. Авторкою роботи запропоновано ряд оригінальних рішень щодо поліпшення функціональності сайту, зокрема: інтерактивна карта, що дозволяє батькам швидко знайти необхідну інформацію про дитячий садочок; Зручний та інтуїтивно-зрозумілий інтерфейс з присмною анімацією; Галерея з фотографіями подій різних груп у дитячому садочку; Можливість зміни мови для більшого впливу на суспільство.

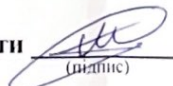
Практичне значення роботи:

Розроблений сайт має значне практичне значення, оскільки дозволяє дитячому садочку №28 «Вулик» ефективніше комунікувати з батьками, надавати актуальну інформацію про роботу закладу, та забезпечує зручний доступ до освітніх ресурсів.

Загальний висновок:

Роботу виконано у повному обсязі, у відповідності із завданням. Здобувач освіти Циплінська Дарія Сергіївна заслуговує на оцінку «відмінно».

Рецензент: Мороз Сергій Анатолійович
(прізвище, ім'я, по-батькові, посада)

Рецензент кваліфікаційної роботи 
(підпис)

«7» 06 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Здобувач вищої освіти Ципліньська Дарія Сергіївна
(прізвище, ім'я, по батькові)

група ІПЗ-42

Тема кваліфікаційної роботи Розробка сайту дитячого садочка №28 «Вулик»

Керівник к.т.н., доц. Самчук Л. М.

Актуальність теми. Тема кваліфікаційної роботи є актуальною через зростаючу необхідність інтеграції сучасних інформаційних технологій в освітню сферу. Розробка сайту для дитячого садочка забезпечує інформаційну доступність та ефективну комунікацію між закладом та батьками. Це відповідає сучасним вимогам цифрової ери та сприяє підвищенню якості освітніх послуг.

Об'єкт дослідження. Об'єктом дослідження є процес розробки веб-сайту дитячого садочка №28 «Вулик».

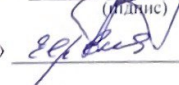
Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи. У першому розділі було проведено аналіз сучасного стану проблеми. У розділі другому розроблено специфікацію вимог, вибрано технології для розробки проекту. У третьому розділі реалізовано сайт, виконано тестування, оптимізацію та створено мобільну версію. Практична значимість роботи полягає у створенні функціонального сайту, що покращує комунікацію та доступність інформації.

Загальний висновок

Роботу виконано у повному обсязі у відповідності до поставлених завдань, а здобувач Ципліньська Д.С. заслуговує на оцінку «відмінно»

Керівник 
(підпис)

Самчук Л.М.
(прізвище, ім'я, по батькові)

«07»  2024 р.

АНОТАЦІЯ

Ципліньська Д. С. Розробка сайту для дитячого садочка №28 «Вулик».
Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків і пропозицій, списку використаних джерел та додатків.

У першому розділі здійснено аналіз сучасного стану проблеми розробки сайту і сформульовано завдання. В другому розділі визначено вимоги до розроблюваної системи, а також засоби, методи і алгоритми розробки. У третьому розділі розроблено практичну реалізацію сайту. У висновках узагальнено інформацію, відображену у попередніх частинах. Результати розробки демонструють можливості ефективного впровадження інформаційних технологій у розробку дитячих дошкільних закладів, зокрема, забезпечення зручного доступу до інформації про діяльність дитячого садочка.

Ключові слова: веб-сайт, розробка, дитячий садок, інтерактивність.

ABSTRACT

Tsyplinska D. S. Development of a Website for Kindergarten No. 28 "Vulyk". Manuscript.

Bachelor's qualifying thesis of the educational program "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualifying thesis consists of an introduction, three sections, conclusions and proposals, a list of used sources and annexes.

In the first chapter, an analysis of the current state of the problem of developing computer games was carried out and the task was formulated. The second section defines the requirements for the developed system, as well as means, methods and development algorithms. In the third chapter, the practical implementation of the website was developed and tested. The conclusions summarize the information presented in the previous parts. The development results demonstrate the possibilities of effectively implementing information technologies in the development of preschool institutions, in particular, ensuring convenient access to information about the activities of Kindergarten.

Keywords: website, development, kindergarden, interactivity.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ У СФЕРІ РОЗРОБКИ ВЕБ-САЙТУ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ.....	9
1.1 Аналіз сучасного стану проблеми.....	9
1.2 Постановка завдань для бакалаврської кваліфікаційної роботи.....	19
Висновки до розділу 1.....	19
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ВЕБ-САЙТУ ДИТЯЧОГО САДКА «ВУЛИК».....	20
2.1 Аналіз, визначення вимог до розроблювального програмного забезпечення та проектування програмного забезпечення.....	20
2.2 Вибір засобів, методів та алгоритмів вирішення поставленого завдання.....	30
Висновки до розділу 2.....	42
РОЗДІЛ 3 РОЗРОБКА САЙТУ ДЛЯ ДОШКІЛЬНОГО НАВЧАЛЬНОГО ЗАКЛАДУ.....	43
3.1 Практична реалізація сайту.....	43
3.2 Тестування та налагодження інформаційно-комп'ютерної системи.....	51
3.3 Оптимізація продуктивності веб-сайту.....	54
3.4 Розробка мобільної версії веб-сайту.....	56
Висновки до розділу 3.....	57
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61

ВСТУП

Актуальність теми. На сучасному етапі розвитку суспільства, коли комп'ютери та інтернет відіграють важливу роль у житті кожної людини, створення та просування веб-сайтів є неабиякою частиною успішної діяльності кожного закладу, зокрема дитячих садків. Веб-сайт – це не лише інструмент для просування продуктів чи послуг, але й візитна картка компанії в онлайн-світ, яка відображає її професіоналізм та доступність для клієнтів у будь-який час. Для дитячих садків веб-сайт є платформою для обміну інформацією. Тому дуже важливо, щоб веб-сайти для дитячих садків відповідали сучасним стандартам та тенденціям. Основні світові тенденції включають:

- дизайн, зручний для користувачів. На сайті не повинно бути зайвих елементів або надлишку тексту, які можуть заплутати користувачів;
- зручний інтерфейс. Сайт повинен бути легким у використанні та зрозумілим для всіх користувачів. Важлива інформація про садок повинна бути на головному екрані, така як розклад, меню харчування, контакти та інше;
- мобільна оптимізація. Сайти повинні добре працювати на різних пристроях, включаючи телефони та планшети;
- використання соціальних мереж. Веб сайти можуть посилатися на соціальні мережі, щоб поширювати важливу інформацію про важливі новини;
- безпека та конфіденційність. Кожен сайт повинен захищати дані дітей та іншу інформацію, щоб вони залишалися безпечними та приватними;
- мультимедійний контент. Використання аудіо, відео, анімації та інших мультимедійних матеріалів робить сайт цікавим та привабливішим.

Отож, ці тенденції вказують на те, що розробка сайту має, перш за все, бути орієнтованою на користувачів та забезпечувати легкодоступність для всіх бажаючих.

Дана розробка є актуальною, адже вона передбачає створення сучасного сайту який відповідає вимогам та очікуванням навчального закладу та самих батьків. Сучасні батьки все частіше використовують цифрові інструменти, щоб отримувати інформацію та комунікувати з навчальним закладом. На жаль, у переважній більшості випадків сайти мають лише приватні дитячі садки або ті, які є досить відомими у місті. Це створює нерівність у доступі до інформації для батьків, чиї діти відвідують інші дитячі садки, особливо в сільських або в менш популярних районах області. Тому потрібно вирішувати цю проблему шляхом створення сайтів для всіх садків, незалежно від їх популярності чи місця розташування. Створення сайту для дитячого садка сприятиме прозорості в організації навчального закладу та забезпечить рівні можливості для всіх дітей. Актуальністю створення сайту полягає також у тому, що якщо потрібно донести інформацію максимально швидко до великої кількості людей, то краще ніж за допомогою власного сайту зробити це не вийде ніяк.

Мета кваліфікаційної роботи бакалавра є створення сучасного веб-сайту для садочка «Вулик» Луцького району Волинської області.

Завдання на дану роботу:

- провести аналіз літературних джерел;
- розробити інтуїтивно зрозумілий та привабливий дизайн;
- забезпечити інформаційну доступність;
- оптимізувати сайт для різних пристроїв;
- інтегрувати карти та геолокацію;
- розробити візуальні ефекти та анімацію.

Об'єкт кваліфікаційної роботи бакалавра є сайт дитячого садка «Вулик».

Предметом кваліфікаційної роботи бакалавра є процес розробки сайту для дошкільного навчального закладу №28 «Вулик» Луцького району Волинської області.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ У СФЕРІ РОЗРОБКИ

ВЕБ-САЙТУ

1.1 Аналіз сучасного стану проблеми

Дитячі садки є важливими закладами освіти, які відіграють ключову роль у навчанні та розвитку дітей на ранніх етапах життя. Однак сучасні дитячі садки все дедалі частіше стикаються з різними проблемами, зокрема в управлінні та організації освітнього процесу, взаємодії з батьками, забезпечення безпеки та зручності для дітей. У нинішній цифровій епосі, веб-сайт для дитячого садка стає чимось більше ніж просто зручністю – це важливий інструмент для зв'язку та інформації. Проте, також дуже часто зустрічаються проблеми з неефективними або складними у використанні та сприйнятті сайтами. Батьки прагнуть простого та зрозумілого інтерфейсу та легкого переміщення по сайту, але часто зіштовхуються зі старим дизайном і нестачею необхідної інформації та функцій. Такі проблеми можуть ускладнити взаємодію між батьками та дитячим садком, що врешті може посприяти негативному ставленні до закладу. Ускладнений доступ до актуальної інформації про події чи оновлення розкладу може спричинити труднощі для батьків. Крім того, дитячі садки можуть зіштовхнутися з технічними обмеженнями, такими як відсутність достатнього фінансування для розробки та підтримки сучасних веб-сайтів, або нестача персоналу для обслуговування таких ресурсів. Отож, враховуючи ці виклики, важливо провести глибокий аналіз ситуації, щоб з'ясувати, як покращити якість веб-сайтів для садків. В результаті таких аналізів можна запропонувати кращі рішення для покращення веб-сайтів, щоб вони відповідали сучасним вимогам та задовольняли всіх користувачів включаючи батьків.

У 1990-х роках появилися перші веб-сайти для дитячих садків, які характеризувалися статичними дизайном та функціями, що означало, що вони

не мали жодних динамічних функціональностей. Вони склалися переважно з текстового та графічного контенту, без можливостей взаємодіяти чи оновлювати сайт. Перші веб-сайти для дитячих садків, як і більшість веб-сайтів того часу, були побудовані за допомогою мови розмітки HTML для структурування контенту та таблиць стилів CSS для стилізації вмісту. Такий підхід забезпечував базову організацію всього потрібного вмісту, проте обмежував можливості сайту в плані функціональності, адаптивності та інтерактивності. Рисунок 1.1 демонструє приклад використання сайтів, розроблених у 90-х роках. Він демонструє особливості тогочасних можливостей та технологій в плані розробки, включаючи простоту структурузації сторінок, використання основних або давно застарілих HTML-тегів та CSS-властивостей.

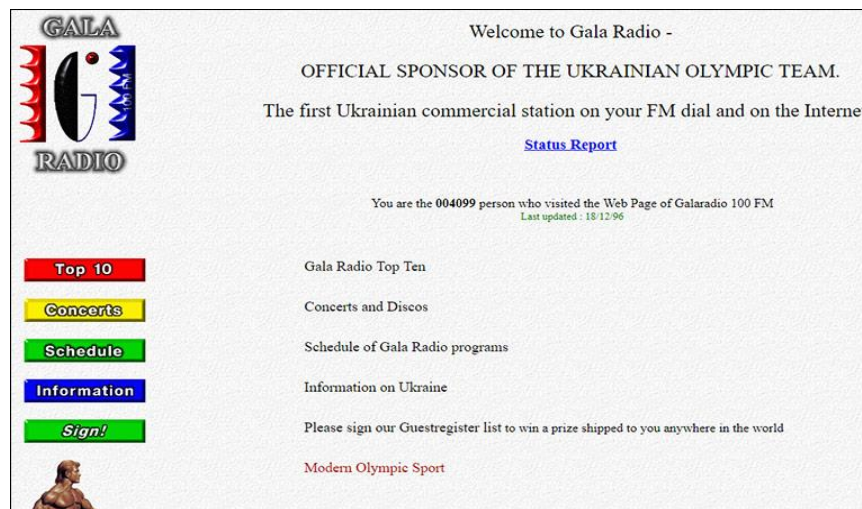


Рисунок 1.1 – Приклад сайтів 90-х років [1]

Розробка на ранніх етапах дала поштовх для подальшого розвитку більш складних і динамічних веб-ресурсів, що поступово вдосконалювали цей процес та ставали дедалі популярними серед розробників і замовників. Вже з 2010 року функціональні можливості веб-сайтів тільки зростали, надаючи можливості онлайн-записам на консультації, через спеціально створені форми. З розвитком нових технологій, процес створення сайтів ставав дедалі

затребуваним серед підприємств. З'явилися нові технології, такі як JavaScript, яка залишається однією з основних мов програмування для створення веб-сайтів і до сьогодні. Також розробники все частіше почали використовувати різнофункціональні сучасні фреймворки, такі як: React, Angular та Vue.js, які значно спростили створення динамічних інтерфейсів та інтерактивного контенту. Почали широко проявляти себе у використанні і системи управління контентом (CMS), як от WordPress чи Drupal, які дозволили адміністраторам дошкільних чи інших різноманітних установ самостійно керувати контентом без залучення різного роду програмістів. Всі ці технології сприяли створенню більш зручних і безпечних систем для взаємодії з батьками та персоналом закладу. Загалом, початково в Україні веб-сайти для дитячих навчальних закладів були рідкістю, а їх використання було більш поширеним в США та в інших деяких країнах. З розповсюдженням Інтернету та зростанням інформаційної культури в Україні дитячі навчальні заклади все частіше почали розуміти важливість наявності свого власного ресурсу.

Для досягнення оптимального рівня ефективності та зручності веб сайту для дитячих установ, було проведено ретельний аналіз наявної літератури та джерел, присвячених даній темі. Перегляд джерел дозволив отримати уявлення про поточний стан дослідження у цій галузі, включаючи нові тенденції та інновації. Завдяки аналізу було виявлено актуальні методи, які можуть сприяти розробці ефективного та зручного сайту. Такий підхід забезпечує більш глибоке розуміння вимог і потреб цільової аудиторії, а також дозволяє застосувати сучасні технології та інструменти для реалізації завдання. Результати перегляду літератури допомогли мені у створенні комплексного плану розробки сайту, враховуючи аспекти функціональності, інтерактивності та дизайну.

Розглянемо аналіз літературних джерел з захоплюючого обговорення про важливості наявності веб-сайту для дошкільних закладів, яка представлена в статті «The importance of having a website for your childcare

business» від Parenta [2]. Ця стаття розглядає, чому веб-сайти стають дедалі важливими інструментами для дитячих установ у сучасному світі. Стаття починається з інтерв'ю з керівником цифрової команди Parenta Семом Дейві, який підкреслює цінність веб-сайтів у нашому цифровому світі. Зокрема, він зазначає, що більшість клієнтів, особливо батьки, активно використовують пошукові системи для пошуку інформації про освіту своїх дітей. Відсутність веб-сайту в цьому контексті може призвести до втрати значної кількості потенційних клієнтів. Далі в статті розкриваються способи, за допомогою яких веб-сайт може зміцнити відносини з батьками, надаючи можливості для спілкування та інформування про події в дитячому садку, а також демонструє увагу та турботу про дітей та їхнє оточення. Крім того, стаття акцентує увагу на перевагах наявності веб-сайту перед конкурентами, які його не мають. Зокрема, присутність в Інтернеті дозволяє закладам бути доступним для клієнтів цілодобово, що сприяє кращому залученню потенційних клієнтів.

Наступним, досить цікавим джерелом для огляду є стаття «Why is a Website needed for Preschool» від Sonali Bindal [3]. У цій статті Біндал наводить різні причини створення веб-сайту для дошкільних закладів. Автор підкреслює, що веб-сайт сприяє підвищенню впізнаваності та репутації дитячого садка, а також приверненню уваги батьків та інших зацікавлених сторін. Демонстрація досягнень, інфраструктури та умов для дітей, на думку автора допомагає батькам розвинути довіру до закладу навчання. Біндал також акцентує на важливості створення блогу для залучення батьків та підвищення їх інтересу до життя в садку. Це дозволяє забезпечувати батькам постійний доступ до актуальних новин та порад, а також надати інформацію про події та активності в навчальному закладі. Більше того, автор зазначає, що веб-сайт може залучити запити на вступ та зміцнити довіру серед батьків. Це надає можливість показати унікальні можливості садочка та виділитися серед інших закладів. Таким чином, стаття Біндал робить висновок про важливість веб-

сайтів для дошкільних закладів з точки зору покращення їх репутації, взаємодії та підвищенню довіри.

Ще одним важливим та цікавим джерелом для ознайомлення є стаття «Designing WebSites For Kids: Trends And Best Practices» від Smashing Magazine [4]. У цій статті розглядається питання дизайну веб-сайтів, призначених тематиці дошкільних навчальних закладів, та обговорюються найкращі методи для забезпечення того, щоб сайти були привабливими, безпечними та корисними для дітей. Основні пункти статті включають:

- 1) розуміння аудиторії. Важливо розуміти вікову категорію для якої розробляється сайт, щоб підібрати відповідний контент та візуальні елементи;
- 2) візуальна привабливість. Веб-сайти, які розробляються для дитячих садків, повинні бути яскравими та легкими у використанні;
- 3) активність. Стаття акцентує увагу на важливості створення секції «активностей» для залучення дітей, наприклад, розвиваючі відео, ігри тощо.

Отож, на основі цих джерел стає зрозуміло, що добре спроектований сайт дитячого садка може значно покращити надання інформації для батьків, підвищити репутацію дитячого садка та залучити нових клієнтів.

На ринку існує низка аналогів систем для дошкільних установ, зокрема веб-сайти, мобільні додатки та програмне забезпечення для управління. Для точного розуміння сучасного стану ринку дитячих садків і визначення найкращих рішень для розвитку нового веб-сайту, я провела порівняльний аналіз вже існуючих сайтів. Такий огляд дає можливість виявити переваги, недоліки та потенційні області покращення проєктованого сайту, враховуючи досвід конкурентів. Оцінюючи які функції та можливості користувачі вважають корисними та зручними, тоді я зможу підібрати оптимальний набір функціоналу для сайту. Крім того, аналіз неабияк допоможе визначити недоліки і проблеми існуючих веб-сайтів. Уникнення цих проблем у розробці нового сайту дозволить створити більш ефективний та корисний інструмент для всіх охочих. Для глибокого аналізу обрано кілька сайтів, які є

популярними серед цільової аудиторії та надають подібні функціональні можливості. Нижче наведено опис кожного з них. На рисунку 1.2 зображено головну сторінку веб-ресурсу «Світлинка» [5].



Рисунок 1.2 – Головна сторінка веб-сайту дитячого садка «Світлинка» [5]

Аналізуючи перший сайт, можна сказати, що він створений з акцентом на сучасний, ефективний та привабливий дизайн. Завдяки гармонійному поєднанню кольорів створюється естетично приємний вигляд, який підвищує задоволення користувачів від взаємодії з ресурсом. Щодо функціональності сайту, то він надає широкий спектр можливостей: користувачі можуть ознайомлюватися з інформацією про різноманітні групи, події, розклад занять, а також отримати доступ до фото та відео галереї. Історія про садок що представлена на сайті, створює додатковий зв'язок з цільовою аудиторією. Також додатковою перевагою цього сайту є хороша адаптивність, яка забезпечує користувачеві можливість відвідувати сайт з різних пристроїв, різних екранів. Для удосконалення цього сайту, на мою думку, можна додати більше цікавих розвиваючих відео для дітей а також порад з виховання дітей для батьків. Це залучатиме все більш широкому спектру

користувачів та підвищенню цінності сайту як ресурсної платформи для батьків та дітей.

На рисунку 1.3 відображено головну сторінку сайту «Лелітка» [6].



Рисунок 1.3 – Головна сторінка веб-сайту дитячого садка «Лелітка» [6]

Аналізуючи наступний сайт, що надає доступ до інформації про дитячий садок «Лелітка» у місті Луцьку, можна зазначити, що функціональний аспект сайту охоплює важливі елементи: інформацію про педагогічний колектив, освітні програми, новини, галерею, а також контакти для швидкого зв'язку. Проте, на мою думку, дизайн вищезазначеного сайту є не зовсім вдалим та зручним у використанні. Надмірна кількість контенту, розміщеного на одній сторінці, може спричинити плутанину та ускладнити навігацію для користувачів. Додатковим недоліком є те, що цей сайт не адаптований до використання різними пристроями, що обмежує зручність доступу для користувачів з різних пристроїв. Це може зменшити залучення користувачів та ефективність веб-ресурсу. Для вдосконалення цього сайту рекомендується приділити більше уваги у покращенні дизайну, у тому числі оптимізації структури сайту для розподілу контенту на окремі тематичні сторінки. Також важливо створити адаптацію сайту під різні пристрої, що підвищує доступність та зручність використання ресурсу. Крім того, слід вибрати

гармонійну палітру кольорів, яка покращить візуальну привабливість сайту і посприє приємною сприйняття інформації.

Останнім оглядом та аналізом сайту дитячого садка є «Twinkling stars» [7]. На рисунку 1.4 зображено головну сторінку іноземного сайту.



Рисунок 1.4 – Головна сторінка сайту дитячого садка «Twinkling Stars» [7]

Аналіз сайту «Мерехтливі зірочки» демонструє його стильний та сучасний дизайн, що забезпечує легкий доступ до основних розділів. Гармонійне поєднання кольорів та розташування елементів створюють привабливий вигляд і підвищують комфорт користувачів під час взаємодії з ресурсом. Сайт використовує мінімалістичний стиль, який сприяє зосередженню користувачів на пошуку подібної інформації. Це підвищує ефективність роботи з веб-ресурсом і покращує користувацький досвід. Функціональні можливості сайту включають детальну інформацію про освітні програми, навчальні підходи, та командний склад. Крім того, сайт пропонує можливість онлайн запису на консультацію через зручну онлайн-форму, що полегшує комунікацію з адміністрацією. На головній сторінці сайту розміщена інтуїтивно зрозуміла навігація, яка дозволяє швидко та легко знайти необхідні розділи. Сайт також адаптований для використання на різних пристроях, що забезпечує його зручність для широкого кола користувачів. Для подальшого вдосконалення веб-сайту варто розширити наявну інформацію про садочок та

виховання дітей, надаючи батькам більше конкретних даних про програми та підходи, що використовуються в навчальному процесі. Це неабияк підвищить цінність ресурсу для батьків і може сприяти кращому розумінню його діяльності.

На основі аналізу технологічного стеку веб-сайтів «Світлинка», «Лелітка» та «Twinkling Stars» можна зробити висновки про використані технології та їх вплив на ефективність роботи сайтів. Аналіз включає дослідження платформ, мов програмування, фреймворків, баз даних, інструментів кешування, бібліотек JavaScript, інструментів аналітики та інших технологій:

1) технології фронтенду:

- HTML та CSS. Дані технології використовуються для створення структури та стилізації сторінок на всіх трьох сайтах. Вони забезпечують стандартну розмітку та оформлення контенту, створюючи зручний користувацький інтерфейс;

2) JavaScript бібліотеки:

- «Світлинка» та «Лелітка» використовують бібліотеку JQuery для додавання динамічності та забезпечення взаємодії з користувачем. Бібліотека JQuery дозволила цим сайтам спростити роботу з DOM, обробляти події, анімацію та інші динамічні функції;

- «Twinkling Stars» використовує бібліотеку Modernizr, яка служить для перевірки можливостей браузера та надання різних функцій;

3) технології бекенду:

- PHP. Веб-сайти «Світлинка» та «Лелітка» використовують мову PHP для обробки динамічного контенту та взаємодії користувача зі сайтом;

- MySQL. Сайт «Лелітка» використовує MySQL як базу даних для збереження та обробку даних;

- Laravel. Веб-сайт «Світлинка» використовує Laravel, який надає ефективні інструменти для роботи з бекендом;

4) платформи CMS:

- CMS WordPress. Веб-сайти «Світлинка» та «Лелітка» побудовані за допомогою WordPress. Він допомагає легко створювати та керувати контентом сайту;
- SquareSpace. «Twinkling Stars» працює на платформі SquareSpace, яка надає широкий спектр шаблонів для розробки сайтів.

В таблиці 1.1 надано порівняльний аналіз трьох вищезазначених сайтів дитячих садків.

Таблиця 1.1 – Порівняння існуючих веб-сайтів дитячих садків

Параметри	«Світлинка»	«Лелітка»	«Twinkling Stars»
Технологічний стек	Laravel, PHP, CMS WordPress	PHP, CMS WordPress	CMS SquareSpace, Modernizr
Можливості	Онлайн запис, фото та відео-матеріали, актуальні події та новини	Перегляд фото, велика кількість новин та порад	Реєстрація дітей онлайн, фото та відео-матеріали
Переваги	Легкість у користуванні та пошуку потрібної інформації, адаптивність	Великий обсяг корисної інформації	Оптимізація адаптивності, сучасний дизайн
Недоліки	Потребує постійного оновлення, можливі технічні складності через велике навантаження	Незручний дизайн, відсутність адаптації	Менша кількість функціональних можливостей

Розробка нової системи для дитячого садка є критично важливим кроком для підвищення його продуктивності, комфорту та впізнаваності серед користувачів. Однією з головних причин необхідності розробки нової системи є можливість створення платформи, що забезпечить легкий доступ до такої інформації, як оперативне оновлення розкладу занять, меню харчування, запланованих заходів, цікавих порад та фотографії дітей. Щодо людей які тільки обирають навчальний заклад для своєї дитини, сайт може стати цінним ресурсом, який надасть можливість ознайомитися з послугами садка, його

програмами, особливостями, інформацією про колектив. Сайт забезпечить батькам можливість робити зважений вибір при виборі навчального закладу, враховуючи власні пріоритети та потреби дитини. Отож, в сучасних умовах дитячий садок має бути не тільки візуально привабливим, але й легкодоступним для батьків в плані отримання всієї необхідної інформації.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Мною проаналізовано ключові аспекти створення сайту для дитячого садка, які необхідно враховувати під час розробки цього проекту. На підставі цього було визначено такі завдання:

- провести аналіз літературних джерел;
- розробити інтуїтивно зрозумілий та привабливий дизайн;
- забезпечити інформаційну доступність;
- оптимізувати сайт для різних пристроїв;
- інтегрувати карти та геолокацію;
- розробити візуальні ефекти та анімацію.

Висновки до розділ 1

Дослідження в області створення веб-сайтів для дитячих садків допомогло мені зрозуміти основні вимоги та аспекти які необхідно враховувати під час розробки нового сайту. Ретельний аналіз існуючих аналогів веб-сайтів та наявних джерел дозволив мені сформулювати чіткі критерії зручності та функціональності, які повинні бути реалізовані у новому проекті. Проведений аналіз також показав, що існує потреба у створенні веб-сайту для дошкільного навчального закладу «Вулик», що сприятиме покращенню доступу до інформації, комунікації та забезпечення зручності для всіх учасників освітнього процесу.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО ВЕБ-САЙТУ ДЛЯ ДИТЯЧОГО САДКА «ВУЛИК»

2.1 Огляд та визначення вимог до програмного забезпечення та проектування системи

Модель програмного забезпечення (ПЗ) – це певний план, за допомогою якого можна організувати послідовність дій та процесів для розробників в межах певного життєвого циклу. Вона описує весь шлях від початку до кінця, забезпечуючи чіткі кроки для виконання різних завдань, наприклад, аналіз вимог, проектування, розробка, тестування та впровадження сайту.

Найбільш підходящими до моєї теми можуть бути наступні моделі ПЗ:

1) гнучка (Agile) модель – це підходи до розробки програмного забезпечення, які характеризуються гнучкістю, співпрацею та адаптацією до змін. Тобто вони дозволяють командам швидко реагувати на нові вимоги, ефективно працювати разом і змінювати плани під час роботи. Основні риси гнучких моделей:

- ітеративний процес. Робота ділиться на короткі етапи або спринти, кожен з них має свою мету;
- тестування протягом усього процесу. Постійне або регулярне тестування забезпечує виявлення проблем на ранніх етапах розробки та виправлення їх;
- гнучкість у плануванні. Можливість змінювати вимоги на різних етапах розробки, що дозволяє швидко адаптуватися до нових умов та забезпечувати максимально ефективний результат.

На рисунку 2.1 продемонстровано приклад використання гнучкої (Agile) моделі.

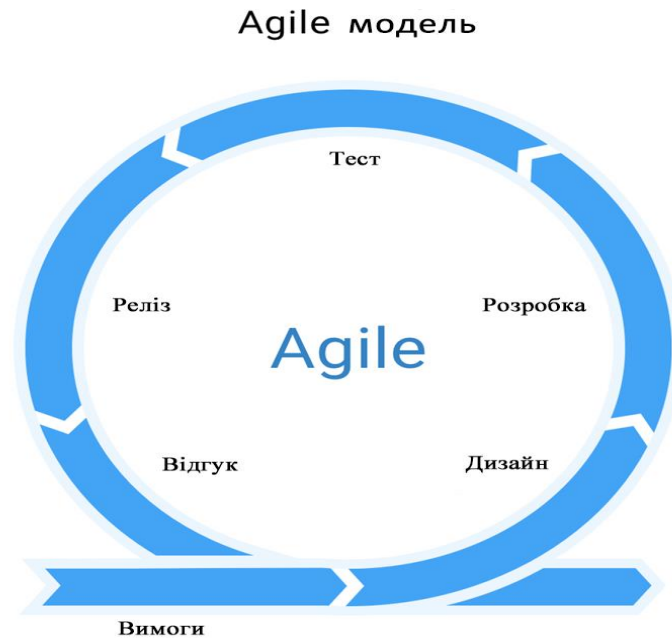


Рисунок 2.1 – Приклад використання гнучкої моделі [8]

2) каскадна модель (лінійна або водоспадна) – підхід до розробки програмного забезпечення, де кожен етап виконується послідовно, перед тим як перейти на наступний етап потрібно повністю завершити поточний етап.

Основні риси каскадних моделей:

- послідовність етапів. Кожен етап розробки програмного забезпечення проходить через ітерацію та виконується послідовно;
- чітка структура. Модель передбачає чітко продумані вимоги та низький рівень змін;
- фіксовані вимоги. Вимоги до проекту визначаються на початку проекту та не змінюються протягом його розробки;
- тестування наприкінці. Тестування виконується тільки після завершення всіх етапів розробки ПЗ.

На рисунку 2.2 зображено приклад використання каскадних моделей.

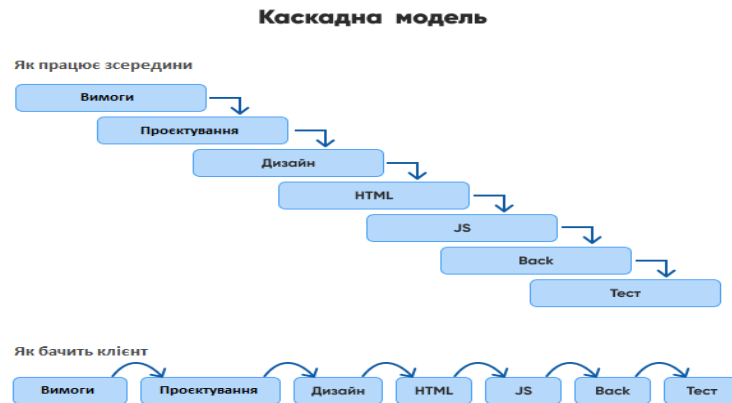


Рисунок 2.2 – Приклад використання каскадних моделей [9]

3) ітеративна модель – це метадологія, яка забезпечує розробку ПЗ через серію ітераційних (повторювальних) циклів. Кожна ітерація проходить через етапи планування, розробки, тестування і вдосконалення, отримуючи зворотній зв'язок від користувачів. Основні риси ітеративної моделі:

- тестування на кожному етапі;
- регулярне вдосконалення через ітерацію;
- гнучкість у вимогах. Вимоги можна змінювати під час розробки.

На рисунку 2.3 продемонстровано приклад використання ітеративної моделі.

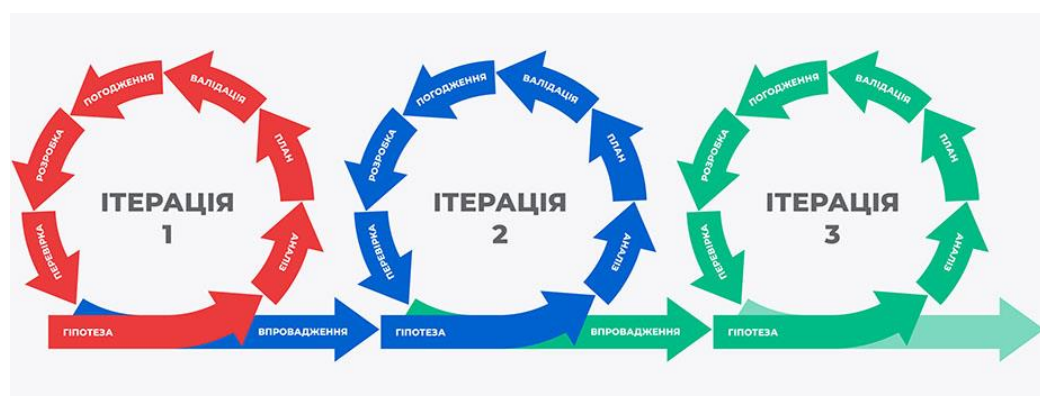


Рисунок 2.3 – Приклад використання ітеративної моделі [10]

Для порівняння вибраних моделей програмного забезпечення, нижче наведена таблиця, яка описує їхні позитивні та негативні сторони (табл. 2.1).

Таблиця 2.1. Позитивні та негативні сторони моделей ПЗ

Модель	Позитивні сторони	Негативні сторони
Каскадна (Waterfall)	Чітка структура та послідовність етапів розробки. Добре підходить для проектів з чітким визначенням вимог	Важко внести зміни після початку роботи. Не підходить для проектів з невизначеними вимогами
Гнучка (Agile)	Гнучкість та можливість швидко адаптуватися до змін. Постійна взаємодія з користувачем	Потребує активної участі клієнтів
Ітераційна	Зосередженість на оцінці та зниженні ризиків на кожному етапі	Потребує ретельного планування та управління

Основні фактори які впливають на вибір моделі:

- постійне оновлення контенту;
- гнучкість та адаптивність;
- доступність в наданні інформації;
- залучення зацікавлених сторін.

Отже, огляд моделей розробки дозволило з'ясувати, що для моєї теми найкраще підходять гнучка модель. Ця модель розробки, також відома як Agile, є одним з найпопулярніших підходів у сучасному розробленні ПЗ, і вона особливо добре підходить для розробки сайту для дошкільного закладу.

Отже, використання гнучкої моделі дозволить:

- 1) тісно взаємодіяти з користувачами;
- 2) гнучко планувати. Agile передбачає розбиття розробки на короткі ітерації (спринти), що дозволяє швидко адаптуватися до змін у вимогах;
- 3) мінімізувати ризики, виявляючи їх на початкових етапах;
- 4) відстежувати процес за допомогою «дошки завдань»;
- 5) забезпечувати якість шляхом тестування та перевірки функціональностей на ранніх етапах розробки.

Таким чином, використання гнучкої моделі забезпечить максимальну гнучкість, чіткість та ефективність під час виконання мого завдання. Цей

підхід дозволить розробити сайт, який відповідатиме потребам користувачів, адаптуватиметься до змін та забезпечить високу якість кінцевого результату.

Наступним етапом було проведено збір та аналіз вимог до системи, щоб гарантувати, що розробка сайту для дошкільного навчального закладу відповідатиме очікуванням користувачів (батьків, вихователів) та адміністрації закладу. Головною метою цього етапу було збір якомога детальнішої інформації щодо вимог та функціональності, які повинні бути інтегровані у дизайн та структуру веб-сайту. Для досягнення максимального розуміння потреб користувачів було використано кілька методів збору даних:

- зустрічі з адміністрацією закладу. На початковому етапі виявлення та аналізу вимог було проведено кілька зустрічей з адміністрацією закладу, спрямованих на обговорення основних вимог до сайту. Під час цих зустрічей адміністрація поділилася своїми очікуваннями щодо майбутнього сайту та висловила ряд побажань щодо його функціональності та дизайну. Адміністрація не раз підкреслювала на важливості зручного та інтуїтивного інтерфейсу для користувачів. Вони зазначили, що сайт повинен бути легким у навігації та доступним для всіх користувачів. Щодо функціональних можливостей, адміністрація підкреслила необхідність в наданні такої інформації як: розклад занять, матеріали та ресурси, фотогалерея, детальний опис садка та його програму навчання. За результатом цієї зустрічі була чітко розроблена концепція веб-сайту, узгоджено вимоги та визначено алгоритм послідовних завдань для подальшої розробки;

- спостереження за діяльністю садочка. Під час мого перебування у дитячому садку я спостерігала за роботою працівників, їхньою взаємодією з дітьми та адміністрацією, а також за тим, як проходить комунікація між персоналом та батьками. Тому було виявлено низку потреб і можливостей для покращення роботи через веб-сайт. Наприклад, адміністрація садочка висловила бажання мати розклад занять та харчування доступними на сайті.

Така функціональність дозволила б батькам та вихователям мати постійний доступ до актуальної інформації про плани занять та меню на кожен день;

- аналіз документів та нормативно-правової бази. За результатами аналізу документації було виявлено вимоги, які повинні бути відображені на сайті, включаючи навчальні плани, нормативно-правові акти а також рекомендації для батьків та вихователів. Зокрема, необхідно відобразити інформацію про важливі закони України, постанови Кабінету Міністрів, накази та листи Міністерства освіти і науки, які стосуються діяльності дошкільного навчального закладу. Надання доступу до цих документів на сайті є необхідним для забезпечення інформованості користувачів про діяльність закладу, його навчальні програми та політику;

- аналіз аналогів веб-сайтів. Ще одним додатковим методом збору був аналіз та огляд аналогів веб-сайтів інших дошкільних закладів. Цей огляд дозволив визначити сильні та слабкі сторони подібних сайтів, а також кращі практики, які можуть бути використані для власного проекту. Особлива увага приділялася функціональності сайтів, зручності у використанні та інструменти використання.

Отож, включення всіх цих методів збору даних дозволило сформуванати певне уявлення про потреби користувачів та обґрунтувати вимоги до розробки сайту, що дасть поштовх до розробки ефективного та зручного веб-сайту.

UML (Unified Modeling Language) або уніфікована мова моделювання – стандарт, який використовується для опису та планування програмних систем.

Основна мета UML полягає в створенні єдиного стандарту для візуалізації, опису, планування та документування архітектури програмних систем. У контексті веб-розробки використання UML може сприяти покращенню спілкування між учасниками проекту, зменшенню помилок і підвищенню якості створюваного веб-продукту.

В нотація UML на загальному рівні виділяють чотири основні типи елементів: фігури (форми або малюнки, що показують різні частини програми,

наприклад, класи, об'єкти або компоненти), лінії (стрілки або штрихи, які з'єднують фігури на діаграмі та показують, як вони пов'язані), значки (спеціальні малюнки, які розміщуються біля фігур і ліній, щоб показати важливі деталі або властивості) та написи (це слова або фрази, які дописуються до діаграми, щоб пояснити різні частини).

У мові моделювання UML існує кілька типів діаграм, які використовуються для візуалізації різних аспектів програмних систем:

- 1) діаграма класів. Вони показують структуру системи, її класи, атрибути та методи, а також взаємозв'язки між класами;
- 2) діаграма прецедентів. Показує взаємодію користувача з системою;
- 3) діаграма послідовностей. Допомогає відтворити порядок виконання операцій між об'єктами;
- 4) діаграма активностей. Показує потік управління в процесі виконання певної діяльності;
- 5) діаграма станів. Показує можливі стани об'єкта в системі та переходи між ними відповідно до подій.

В рамках розробки веб-сайту для дошкільного навчального закладу було застосовано формалізовані методи синтезу моделей ПЗ з використанням нотації UML (Unified Modeling Language). Я обрала методи моделювання UML для системи, структурований та наочний підхід до розробки.

Для синтезу моделей ПЗ було розроблено діаграму випадків. Діаграма випадків використання (Use Case Diagram) або діаграма прецедентів моделює взаємодію між користувачами та системою, показуючи, як користувачі взаємодіють з різними функціями системи. Ці діаграми відображають різні ролі користувачів (або акторів) та дії (випадок використання), які можуть виконуватися в системі (табл. 2.2-2.3). У випадку з моїм завданням, актори можуть бути різних типів, наприклад, батьки, вихователі, адміністрація. Кожен актор має свої функціональні спроможності та завдання, які вони виконують в системі.

Таблиця 2.2 – Демонстрація акторів та опис їх ролей

Актор	Опис ролі
Батьки	Батьки виступають як основні учасники, які взаємодіють з системою. Вони мають можливість бачити загальну інформацію про садок, його події, розклад харчування тощо
Вихователі	Роль вихователя у впливі на контент сайту полягає в активному оновленні інформації про заняття, події. Вони також можуть додавати фотографії, змінювати інформацію щодо кадрового складу та висвітлювати поради щодо виховання дітей
Адміністратор	Адміністратор відповідає за безперервну підтримку сайту закладу. Йому доручено технічне обслуговування, оновлення, безпека сайту, оптимізація швидкості та продуктивності, резервне копіювання даних і розвиток сайту
Сайт закладу	Сайт закладу виступає як основний засіб комунікації з батьками та іншими учасниками, де публікуються події, розклади, програма навчання, галерея тощо

Таблиця 2.3 – Демонстрація прецедентів різних користувачів та їх опис

Прецедент	Опис прецеденту
Вхід на сайт	Користувач входить на сайт за посиланням або текстовим вводом в пошукову рядку
Перехід по інших сторінках	Користувачі переходять на різні сторінки сайту для отримання різної інформації
Запити на інформацію про садок	Користувач може взаємодіяти з меню сайту, який надає всі сторінки з різною інформацією
Перегляд пропозицій	Користувач може переглядати пропозиції, які пропонує навчальний заклад
Перегляд розкладу занять	Користувач може переглядати актуальний розклад занять для дітей у дитячому садку
Перегляд розкладу харчування	Користувач може переглядати актуальний розклад харчування для дітей у дитячому садку

Продовження таблиці 2.3

Прецедент	Опис прецеденту
Перегляд кадрового складу	Можливість переглядати актуальну інформацію про кадровий склад дитячого садка, вихователів, адміністраторів тощо
Перегляд або завантаження документів	Користувач може переглядати чи завантажувати різні документи, які пов'язані з діяльністю дитячого садка, наприклад, закони України, постанови КМУ, Накази МОН, Листи МОН тощо
Перегляд програми	Користувач може переглянути програми дитячого садка
Перегляд галереї	Можливість переглядати, завантажувати фото та відео матеріали, які були опубліковані на сайті
Перегляд подій	Перегляд актуальних подій або заходів у дитячому садку
Поради щодо виховання дітей	Секція надає корисні ресурси для батьків, щодо виховання їхніх дітей поза навальним закладом
Зворотній зв'язок	Включає секцію, яка забезпечує батькам всі номери, чи соціальні мережі різних вихователів чи адміністраторів садка
Звернення до вихователів	Функція, що дозволяє батькам або опікунам звертатись безпосередньо до вихователів з питань, пов'язаних з навчанням та доглядом за дітьми у садку
Звернення до адміністрації	Функція, що надає користувачам звертатися до адміністрації або керівництва садка, з питань, які не можуть бути вирішені вихователями
Перегляд та редагування інформації про садок	Доступний для спеціалізованих користувачів, таких як адміністратори чи вихователі, які мають права на редагування та оновлення контенту
Отримання звернень від батьків	Функція, що дозволяє адміністраторам чи вихователям садка, отримувати повідомлення, запити та запитання від батьків
Підтримка сайту	Забезпечує надання технічної підтримки для вирішення будь-яких проблем, які можуть виникнути під час використання
Оновлення інформації про садок	Включає процес регулярного оновлення інформації на сайті, щоб забезпечити актуальність та достовірність наданої інформації
Безперервна робота у мережі	Забезпечує надійну доступність веб-сайту для користувачів у будь-який час, без відключень або перерв у роботі

На рисунку 2.4 зображено діаграму випадків використання, яка ілюструє різноманітні сценарії взаємодії користувачів з системою, дозволяючи краще зрозуміти їх потреби та очікування.

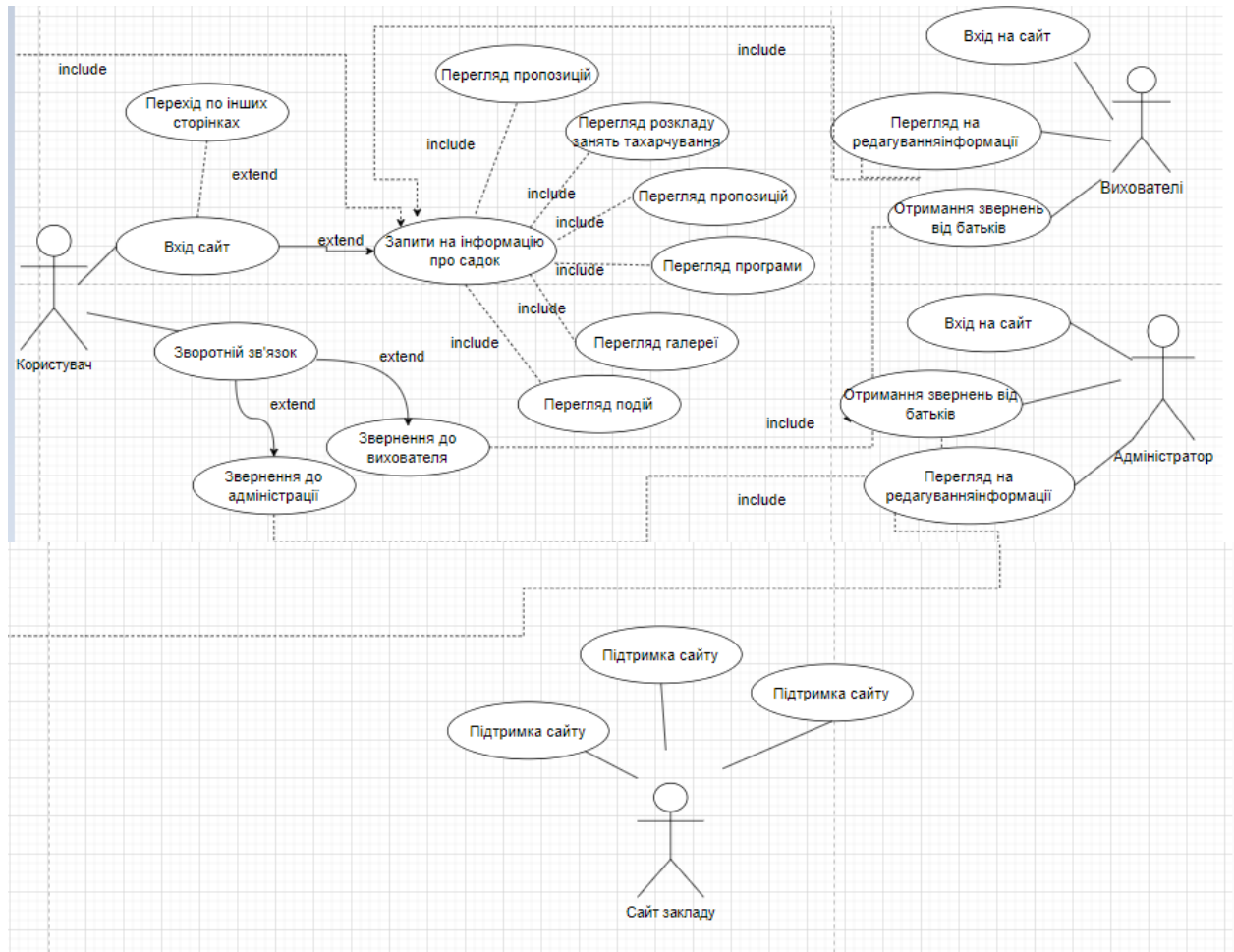


Рисунок 2.4 – Діаграма випадків (прецедентів) використання

Отже, після завершення моделювання програмної системи для дитячого садка, можна зробити висновок про ефективність та придатність моделі для відображення потреб та вимог користувачів. У даному випадку модель випадків використання чітко окреслює ролі різних акторів та їх взаємодію з системою. Кожен прецедент відображає конкретні можливості, доступні для користувачів. Це дозволяє зрозуміти, як система буде використовуватися в реальних умовах та які функції є найважливішими для користувачів.

2.2 Вибір засобів розробки для вирішення поставленого завдання

У галузі програмування існує велика кількість інструментів, які можуть бути використанні для розробки веб-сайту. Сам веб-сайт являє собою колекцію взаємопов'язаних між собою сторінок, яких об'єднує спільна тема, мета або концепція. Кожен веб-сайт будується на певному технологічному стеку, який складається з кількох ключових компонентів, що працюють разом для забезпечення функціональності та естетичного оформлення веб-сайту.

Зазвичай веб-сайти будуються на таких рівнях технологічного стеку:

1) рівень клієнтської сторони (Front-End) – це частина технологічного стеку, яка відповідає за взаємодію користувача з веб-сайтом та його візуальне відображення;

2) рівень серверної частини (Back-End) у технологічному стеку відповідає за обробку, зберігання даних, забезпечення безперервної роботи.

Ці рівні взаємодіють між собою, забезпечуючи злагоджену роботу веб-сайтів. Також важливо зазначити, що файли, які відповідають за фронтенд веб-застосунку також зберігаються на бекенді, та відправляються користувачу по його запиту [11].

Для розробки на рівні клієнтської та серверної сторін можна використовувати різні інтегровані середовища розробки (IDE), одні з найбільш популярніших це Visual Studio Code (VS Code) – найпопулярніший редактор коду, відомий своїм потужним функціоналом та легкістю у використанні; WebStorm – інтегроване середовище розробки, яке спеціалізується на веб-розробці; Sublime Text – менш популярний редактор, проте все ще є значущим та зручним інструментом для розробників.

Нижче наведено більш докладний опис кожного з рівнів технологічного стеку:

1) Front-End містить три обов'язкові складові при розробці:

– HTML – скорочення від «Hyper Text Markup Language» – що в перекладі означає «Мова розмітки гіпертексту» вона надає структуру веб-сторінки, тобто певну основу для розташування контенту на веб-сторінках за допомогою тегів, посилань, зображень, абзаців тощо.

Одним із найбільших переваг HTML є створення семантичної структури веб-сторінки, що розуміють браузер, пошукові системи та інші інструменти аналізу даних.

Використання цих тегів допомагає точно визначити різні елементи веб-сторінки, такі як заголовки, статті, розділи, навігація та інші частини контенту.

– CSS (Cascading Style Sheets) або каскадні таблиці є мовою стилів, яка використовується для оформлення та дизайну веб-сторінок.

CSS надає можливість визначати зовнішній вигляд веб-сторінок, кольори, шрифти, розміри, відступи, розміщення та інші аспекти візуального оформлення.

Перевагами CSS є повторне використання стилів, легкість у сприйнятті, каскадування – стилі можуть спадати (каскадувати) від батьківських елементів до дочірніх. Щодо недоліків CSS, так це кросбраузерні проблеми, що призводить до некоректного відображення контенту у різних браузерах.

CSS має також свої препроцесори SASS/SCSS (Syntactically Awesome Style Sheets) та LESS (Leaner Style Sheets), які все частіше появляються у розробках різних сайтів. Основна їхня перевага, це те що вони значно розширюють можливості звичайного CSS, надаючи розробникам більш зручні інструменти для написання стилів.

Основні можливості SASS та інших препроцесорів це міксини (шаблони, які можна використовувати повторно у коді), вкладання стилів, імпорт, змінні, умовні оператори та цикли, функції, наслідування тощо.

На рисунку 2.5 зображено різницю у використанні SASS (SCSS) з CSS.



Рисунок 2.5 – Різниця у використанні SASS (SCSS) з CSS [12]

– Java Script – високорівнева, інтерпретована мова програмування, що використовується для розробки динамічних, інтерактивних веб-сторінок. Вона відповідає за додавання функціональності, динамічності до сайтів та багато інших спроможностей, наприклад, додавати анімацію, обробляти події, взаємодіяти з користувачами та підвищувати зручність використання сайтів.

Однією з важливих особливостей JS є підтримка об'єктно-орієнтованого програмування. Цей підхід до програмування дозволяє організовувати код навколо об'єктів. Об'єкти містять дані (властивості) та функції (методи), які працюють з цими даними.

Основними особливостями мови програмування JavaScript є:

- 1) інтеграція з HTML та CSS;
- 2) велика кількість бібліотек та фреймворків;
- 3) підтримка асинхронного програмування;
- 4) модульність;
- 5) динамічна типізація;
- 6) використовується як на клієнтській так і на серверній частині (з Node JS).

Щодо недоліків мови, так це невеликий спектр можливостей, у багатьох випадках слід підключати бібліотеки, безпекові ризики, відсутність строгих правил, проблеми продуктивності тощо.

Java Script містить також і свої фреймворки, серед них React – фреймворк, який використовується для створення інтерактивних компонентів на веб-сторінках; Angular – фреймворк, який підходить для великих та складних веб-застосунків; Vue.js – легкий та гнучкий фреймворк, який може створювати динамічні веб-застосунки з мінімальними зусиллями.

Усі три фреймворки популярні в розробці веб-сайтів завдяки своїй ефективності, можливості роботи з компонентами та підтримці технологій, таких як асинхронність та віртуальний DOM.

Webpack та Gulp є одними з найпопулярніших інструментів автоматизації та управління процесами під час розробки веб-сайтів. Їхня мета – полегшення та оптимізація процесу розробки веб-сайтів. Webpack дозволяє ефективно управляти модулями та пакетами JavaScript, об'єднуючи їх у єдиний файл для зменшення часу завантаження. Gulp автоматизує рутинні задачі, такі як компіляція CSS та мінімізація файлів, прискорюючи розробку та тестування. Разом вони забезпечують швидкість та ефективність у процесах розробки сучасних веб-додатків (табл. 2.4).

Таблиця 2.4 – Порівняння популярних інструментів для управління процесами під час розробки

Критерій	Gulp	Webpack
Призначення	Інструмент для автоматизації обробки файлів	Модульний збірник для управління залежностями та організації коду
Тип завдань	Мініфікація завдань, стискання, компілювання, об'єднання	Бандлінг (збірка в один файл), оптимізація, компіляція, спліттинг
База	gulpfile.js	webpack.config.js
Переваги	Можливість паралельно виконувати кілька завдань, легка інтеграція	Управління залежностями, підтримує розділення коду для асинхронного завантаження
Недоліки	Менш ефективний для більш складних проектів	Складність у сприйнятті
Використання	Прості проекти, менша кількість залежностей	Складні проекти, велика кількість модулів

2) Back-End – відповідає за серверну частину ПЗ, включаючи обробку запитів, роботу з базами даних та реалізацію бізнес-логіки (тобто, обробку запитів, обслуговування користувачів та інших дій необхідних для функціональності ПЗ). Back-End відіграє надважливу роль у розробці веб-сайтів, він є одним із ключових етапів для кожної розробки проекту.

Тут використовується велика кількість мов програмування серед них Python – інтерпретована об’єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією [13]; PHP – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера [14]; Node JS – серверне середовище виконання, яке використовує мову програмування Java Script (табл. 2.5).

Таблиця 2.5 – Порівняння популярних інструментів для управління процесами під час розробки

Критерій	PHP	Node JS	Python
Переваги	Висока сумісність з різними базами даних. Велика кількість фреймворків та CMS для розробки	Використовує одну мову (JS) для фронтенду та бекенду. Асинхронна модель для обробки запитів одночасно	Багата бібліотека стандартних функцій та пакетів. Підтримує різні парадигми
Недоліки	Повільніший у виконанні	Менше фреймворків порівняно з іншими мовами	Повільніший порівняно з Node JS
Тип завдань	Створення динамічних веб-сайтів та додатків	Додатки реального часу (чати, стрімінги)	Широкий спектр застосувань
Призначення	Бекенд-розробка для веб-додатків	Виконання серверних запитів	Розробка багатьох типів програм

3) Система керування контентом (CMS) – це система, за допомогою якої можна без необхідності володіння знаннями програмування змінювати

контент. Вона базується на принципі розділення контенту від структури, що означає, що вміст зберігається окремо від дизайну та функціоналу сайту.

Найпопулярнішими CMS на сьогодні є:

1) WordPress – одна з широко використовуваних систем керування контентом. Основною її мовою є PHP. PHP забезпечує WordPress можливість обробляти дані, створювати динамічний вміст, взаємодіяти з базами даних.

Перевагами у використанні WordPress є простота у використанні, плагіни, теми та гнучкість. Щодо недоліків то це потреба у вивченні PHP;

2) Drupal – є більш складнішою CMS порівняно з WordPress. Основною мовою є теж PHP. PHP дозволяє Drupal обробляти дані, взаємодіяти з базами даних та створювати динамічний вміст. Drupal також використовує MySQL та MariaDB як базу даних для зберігання контенту.

Перевагами Drupal є гнучкість, безпека та потужна можливість, щодо недоліків, то це певна складність, високі вимоги, менша кількість плагінів.

Для розробки сайту дитячого садка, я уважно підійшла до вибору кожних засобів, методів та алгоритмів, що забезпечують зручність у розробці, функціональну спроможність та ефективність.

Середовищем розробки (IDE) для написання коду було обрано Visual Studio Code. VS Code – є потужним редактором коду, який поєднує в собі простоту, з могутніми інструментами розробника, такими як IntelliSense для автозаповнення коду та вбудований дебагер [15].

VS Code буде використовуватися як основне середовище не тільки для написання коду, а й для тестування та відлагодження його.

Функціональними можливостями цього редактора є легка налаштованість, інтеграція з Git (вбудована підтримка Git дозволяє відстежувати зміни та керувати кодом), інтерактивне середовище (забезпечує швидкий доступ до терміналу, дебагера, без необхідності лишати редактор).

На рисунку 2.6 зображено головну сторінку середовища VSCode.

Під час написання коду в HTML було вибрано метадологію BEM (Block Element Modifier). Це сучасний підхід до написання коду, який покращує його структурування та організацію, що робить його більш зрозумілим. Основними концепціями метадології BEM є блоки – основні, самостійні частини коду, які представляють окремий компонент сторінки; елементи – частина блоку, які виконують певну підфункцію; модифікатори – використовуються для зміни зовнішнього вигляду, він записується після блоку або елемента.

На рисунку 2.7 зображено приклад BEM метадології.

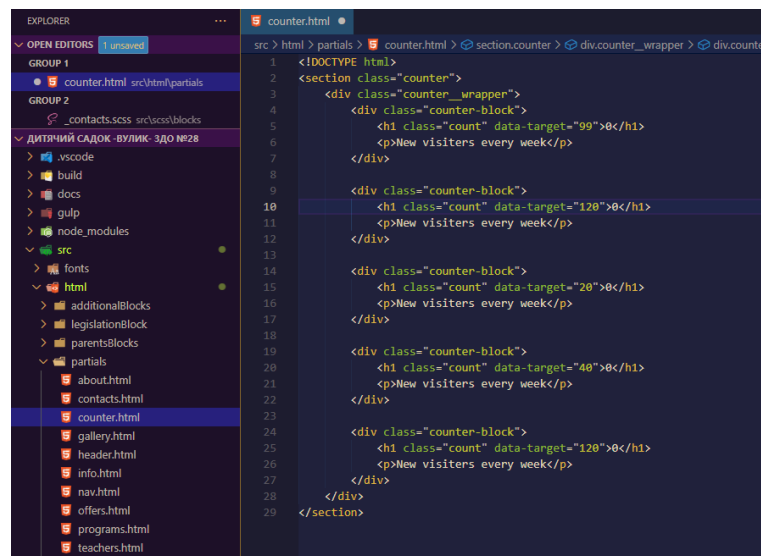


Рисунок 2.7 – Демонстрація коду з використанням BEM метадології

Для написання стилів для сайту було вибрано препроцесор SASS а саме SCSS. Препроцесор SCSS в сутності є покращенням і розширенням синтаксису CSS. Він так само як і CSS складається з селекторів (елементи, класи, ідентифікатори), властивостей (наприклад, color, font-size тощо) та значення (#fff, 23px).

Конвертувати SCSS в CSS можна багатьма способами, але найпопулярнішим та найшвидшим використання консолі: `sass example.scss example.css`.

SCSS покращить процес написання коду, а саме:

- вкладанням стилів один в одного;

- створення змінних;
- створення міксинів. Міксини – це функції, які можна використовувати повторно. Створення міксину позначається ключовим словом «@mixin», за яким слідує назва міксина та його параметри. Його можна викликати в будь-якому місці коду за допомогою «@include»;
- дозволить використовувати різні класифікації та імпортувати файли, щоб розділити стилі на логічні частини.

Так як CSS є мовою стилів, використовувати я буду її постійно при написанні проекту. CSS а точніше його препроцесор SCSS, забезпечить більшу ефективність у роботі зі стилями, дозволить швидше створювати та підтримувати структуровані та стильні веб-сайти завдяки зручним функціям.

Мовою програмування на якій буде базуватися розробка сайту є Java Script. Його вміння виконувати операції на клієнтській стороні, дозволяючи створювати динамічні та інтерактивні інтерфейси, є головною причиною, чому я обрала саме цю мову для розробки.

Java Script є динамічною, високорівневою та інтерпретованою мовою програмування, яка є невід’ємною частиною веб-розробки. Вона працює тільки на клієнтській стороні, що означає, що код використовується у браузері користувача, забезпечуючи швидку відповідь на дії користувача.

Підключення JS до сторінки HTML можна за допомогою наступного коду: '<script src = «index.js»></script>'.
'

Детальний опис інструментів JS:

- DOM (Document Object Model) – програмний інтерфейс (API), який дає змогу розробникам отримувати доступ до структури та вмісту HTML-або XML-документів. DOM являє собою документ у вигляді ієрархічного дерева елементів, де кожен елемент є об’єктів [16]. На рисунку 2.8 зображено структуру DOM.

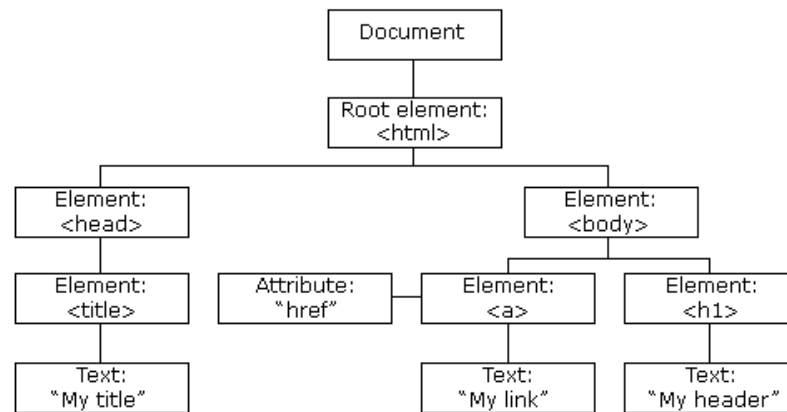


Рисунок 2.8 – Структура DOM [17]

Доступ до елементів DOM та їхньої маніпуляції:

- `document.getElementById()` – метод шукає елемент за ідентифікатором;
- `document.querySelector()` – шукає перший елемент, який відповідає заданому селектору CSS;
- `element.innerHTML` – властивість, яка дозволяє отримувати або змінювати HTML-вміст елементу;
- `element.classList` – надає метод для додавання, видалення та перевірки класів елементу;
- `element.addEventListener()` – метод, який дозволяє додати обробник подій для елементу.

Використання DOM у розробці мого проекту забезпечило доступ до маніпуляції читання, зміни, додавання, видалення, реагування на події тощо;

- інтеграція з HTML ТА CSS. Java Script тісно взаємодіє з цими стандартними інструментами, дозволяючи тісно маніпулювати контентом і стилем веб-сторінок;
- асинхронність. Java Script ефективно працює з асинхронними завданнями, використовуючи проміси (об'єкти для керування асинхронним операціями), `async/await` (спрощений синтаксис для роботи з промісами) та інші механізми. Це дозволить створювати плавні ефективні користувацькі

інтерфейси. У свою чергу асинхронність – це здатність виконувати завдання незалежно одне від одного, тобто якщо веб-сайт запитує дані із сервера, він може продовжувати працювати не чекаючи на відповідь. Проміси дозволяють керувати цими асинхронними операціями, розділяючи на різні стани: очікування (операція ще не завершена), виконання (операція успішно завершена і є результат) та відхилення (виявлена помилка).

Асинхронність використовуватиметься у моєму проекті для покращення оптимізації веб-сайту. У моєму випадку, я використовуватиму асинхронність у багатьох секціях, одна з них – це секція «Лічильник». У цій секції відбудуватиметься асинхронне перерахування кількості дітей різних груп. Секція починатиме перераховувати тільки тоді, коли прокручування сторінки досягне певного моменту.

Отож, Java Script за допомогою якого буде писатися сайт, дозволить розробити як стандартні активні клік так і більш складні асинхронні функції.

У проекті для створення сучасного веб-сайту дитячого садка «Вулик» було вибрано два популярних інструменти для автоматизації завдань та збірки проекту – Gulp та Webpack. Кожен з цих інструментів був обраний для виконання конкретних завдань, що полегшують розробку та підтримку сайту.

Для автоматизації завдань для розробки сайту я обрала Gulp. Цей інструмент багатий на плагіни, які допомагають автоматизувати практично всі аспекти процесу розробки від компіляції вихідних файлів до оптимізації ресурсів. Він базується на JavaScript і побудований на NodeJS.

Основними компоненти Gulp є `gulpfile.js` – це конфігураційний файл, де визначаються всі задачі і плагіни для їх виконання та `plugins` – додаткові модулі, які розширюють функціональність Gulp і дозволяють виконувати різноманітні завдання під час розробки веб-сайтів.

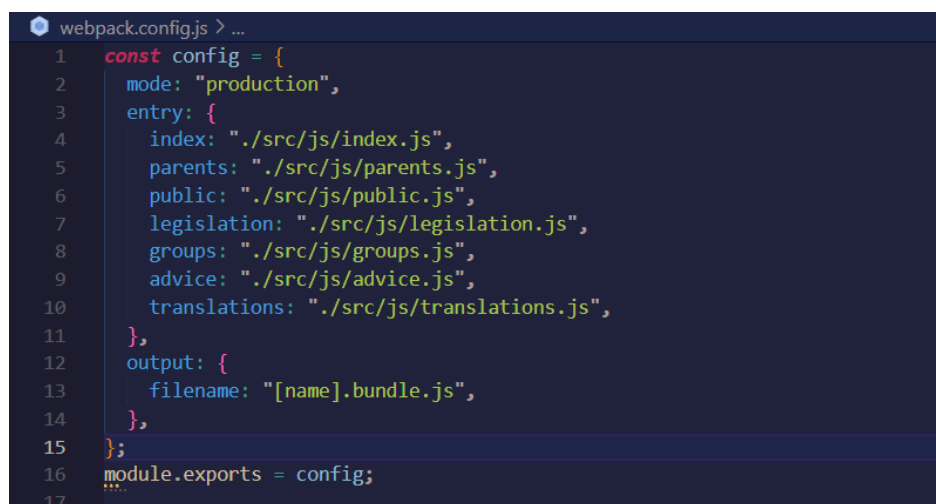
Gulp побудований на потоках, яка використовує концепцію стрімів (streams) для обробки даних. Це ніби потік води, яка протікає через трубу. У Gulp дані обробляються так само, як вода у трубі – вони постійно рухаються з

однієї точки до іншої, проходячи через різні обробники (функції), що змінюють або аналізують наступні дані: читання файлів (зчитування) – це процес, коли в return вписуються всі файли потрібні для цього таску. Далі потік подається у трубу (.pipe()), яка в свою чергу переходить з одного обробника до іншого. Для виводу оброблених файлів потрібно прописати в dest і папку.

Для розділення коду буде використано Webpack. Використання Webpack у розробці сайту допоможе розбити JavaScript код на окремі частини або бандли. Кожен бандл включатиме код, який відповідатиме за певну частину веб-сайту, наприклад, окремий бандл для головної сторінки, інший для сторінки батьків, ще один для галереї тощо.

Webpack також базується на JavaScript і побудований на NodeJS.

Основні компоненти Webpack включають конфігураційний файл webpack.config.js, де визначаються всі налаштування для збірки проекту, та ладери (loaders) і плагіни (plugins), які розширюють функціональність Webpack і дозволяють обробляти різноманітні типи файлів. На рисунку 2.9 відображено розділення коду JavaScript.



```
webpack.config.js > ...
1  const config = {
2    mode: "production",
3    entry: {
4      index: "./src/js/index.js",
5      parents: "./src/js/parents.js",
6      public: "./src/js/public.js",
7      legislation: "./src/js/legislation.js",
8      groups: "./src/js/groups.js",
9      advice: "./src/js/advice.js",
10     translations: "./src/js/translations.js",
11   },
12   output: {
13     filename: "[name].bundle.js",
14   },
15 };
16 module.exports = config;
```

Рисунок 2.9 – Демонстрація конфігураційного файлу Webpack

Висновки до розділу 2

У другому розділі було детально розглянуто специфікацію вимог до розроблюваної системи, а також вибір засобів, методів і алгоритмів до вирішення поставленого завдання.

У першому пункті було проаналізовано та вибрано підходящу модель програмного забезпечення – Agile, яка допоможе мені у розробці сайту самостійно за рахунок гнучкості та коротких циклів роботи. Крім того у цьому пункті було детально описано збір та аналіз вимог до сайту, серед них аналіз аналогів веб-сайтів, зустрічі з адміністрацією закладу тощо. Також я застосувала діаграму випадків використання, щоб чітко окреслити основні функціональні можливості системи з точки зору різних категорій користувачів.

У другому пункті детально розглянуто всі можливі інструменти для реалізації та впровадження сайту, їх переваги та недоліки, а також вибрано інструменти на яких буде розроблятися сайт «Вулик».

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ САЙТУ ДЛЯ ДОШКІЛЬНОГО НАВЧАЛЬНОГО ЗАКЛАДУ

3.1 Практична реалізація сайту

Сайт «Вулик» був створений з використанням HTML для структури, SCSS для стилізації та JavaScript для функціональності. Для автоматизації розробки і оптимізації коду використовувалися інструменти, такі як Gulp та WebPack. Робота над проектом проводилася в середовищі розробки Visual Studio Code (VSCode), що забезпечило зручний та продуктивний робочий процес завдяки багатофункціональному інтерфейсу та набору корисних інструментів.

Мова програмування JavaScript, на якій базується мій проект, використовується для додавання інтерактивності та реалізації логіки сайту. Нижче наведено декілька кодових прикладів, щоб продемонструвати застосування JavaScript у проекті. У лістингу 3.1 продемонстровано скрипти геолокації садка.

Лістинг 3.1 – Скрипти геолокації садка

```
export function initializeSocial() {
  const icons = document.querySelectorAll(".social__map");
  const blocksHide =
document.querySelectorAll(".social__hideBlock");
  icons.forEach((icon, index) => {
    icon.addEventListener("click", (event) => {
      event.stopPropagation();
      icons.forEach((otherIcon) => {
        otherIcon.classList.remove("active");
      });
      icon.classList.add("active");
      blocksHide.forEach((blockHide, i) => {
        if (i === index) {
          blockHide.classList.add("active");
        } else {
          blockHide.classList.remove("active");
        }
      });
    });
  });
}
```

```

    });
  });
});}

```

Кінець лістингу 3.1

Перший приклад коду створює функцію `initializeSocial` для ініціалізації геолокації на сайті. Вона знаходить усі елементи з класами `.social__map` та `.social__hideBlock` для подальшої роботи. Далі, при кліку на іконку, вона стає активною (додається клас `active`), а відповідний блок інформації відображається. Ця функція експортується за допомогою ключового слова `export`, що зробить її доступною для інших модулів JavaScript. У лістингу 3.2 показано асинхронне завантаження геолокації садка.

Лістинг 3.2 – Скрипти асинхронного завантаження геолокації садка

```

    async function loadSocialCode() {
      const { initializeSocial } = await
import("../js/blocks/social.js");
      initializeSocial();
      const mapButton = document.querySelector(".social__map");
      if (mapButton) {
        mapButton.addEventListener("click", () => {
          var map = L.map("map").setView(
            [50.728319090473846, 25.298594209031943],
            13);
          L.tileLayer("https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png" {
            attribution: "© OpenStreetMap contributors",
          })
            .addTo(map);
          L.marker([50.728319090473846,
            25.298594209031943]).addTo(map);
        })
      }
    }
loadSocialCode();

```

Кінець лістингу 3.2

Другий приклад коду також пов'язаний із геолокацією і використовує асинхронне завантаження скрипту. Функція `loadSocialCode` завантажує скрипт з функцією `initializeSocial` і запускає її. Перевіряється наявність кнопки з класом `.social__map`, якщо вона є, додається обробник подій для кліку на цю кнопку. Під час кліку створюється карта за допомогою бібліотеки `Leaflet`, яка відображається на веб-сторінці. На рисунку 3.1 зображено візуальне представлення геолокації.

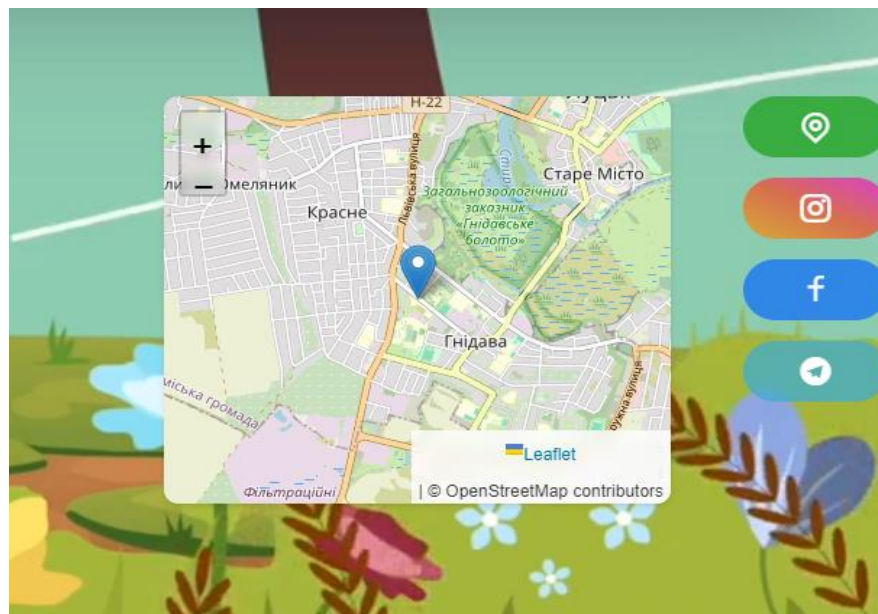


Рисунок 3.1 – Демонстрація геолокації

На головній секції сайту є динамічний слайдер з повною назвою садка. Над слайдером розташоване навігаційне меню з активними посиланнями на відповідні секції сайту. Меню підсвічує поточну сторінку, щоб користувачі знали, де вони знаходяться. Ліворуч розміщено блок для зміни мови сайту. Праворуч розміщено блоки з соціальними мережами садка для швидкого доступу до профілів. На рисунку 3.2 показано головну секцію сайту дитячого садка.

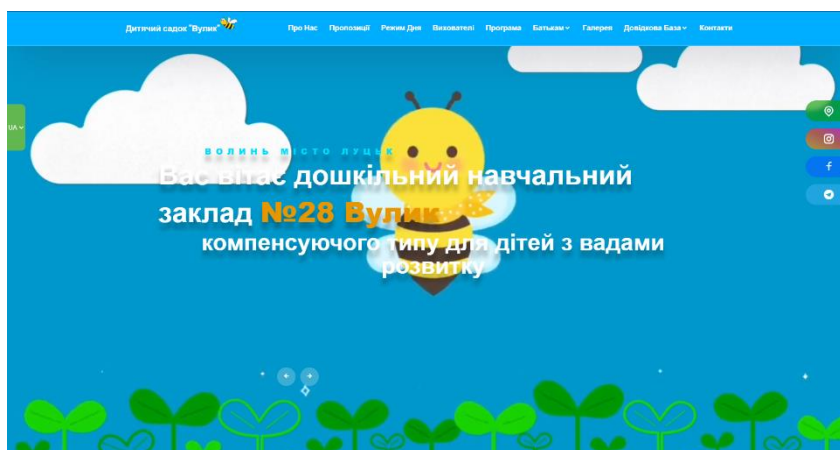


Рисунок 3.2 – Головна сторінка сайту дитячого садка

Наступна секція, яку користувач перегляне після першого враження – «Про нас». Один із потенційних елементів цієї секції є анімаційний слайдер, що включає фотографії та текстовий опис про садок. Також тут присутній анімаційний background. Впровадження всього цього функціоналу відбувалося за допомогою мови програмування JavaScript та бібліотеки Animate.css. На рисунку 3.3 зображено секцію «Про нас».

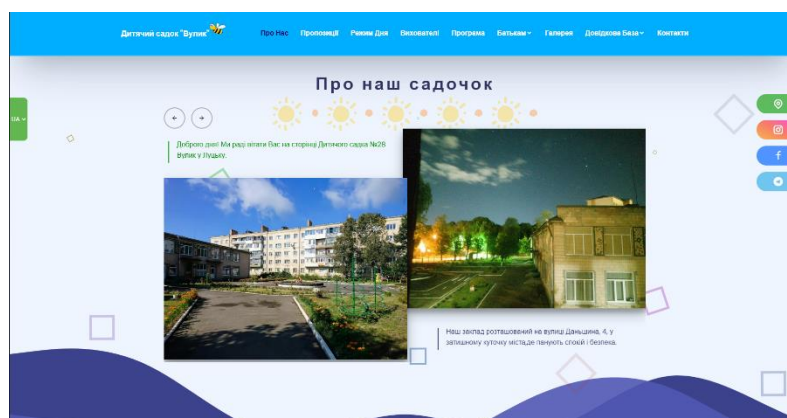


Рисунок 3.3 – Секція «Про нас»

Впровадження слайдера у цій секції відбувалося наступним чином. Спочатку створено функцію toggleActiveClass для додавання або видалення класу active для блоків. Вона робить активним блок з певним index та знімає клас active з інших блоків. Функція changeContent змінює currentIndex в

залежності від напрямку (next або prev) і оновлює клас active для всіх блоків контенту та зображень.

У лістингу 3.3 зображено функціонал перемикання контенту та зображень на веб-сторінці, що працює за допомогою кнопок.

Лістинг 3.3 – Функціонал перемикання контенту та зображень

```
function toggleActiveClass(blocks, index) {
  blocks.forEach((block, i) => {
    block.classList.toggle("active", i === index);
  });
}
function changeContent(direction) {
  if (direction === "next") {
    currentIndex = (currentIndex + 1) % contentBlocks.length;
  } else if (direction === "prev") {
    currentIndex =
      (currentIndex - 1 + contentBlocks.length) %
      contentBlocks.length;
  }
}
linkWrapper.addEventListener("click", (event) => {
  if (event.target.classList.contains("ri-arrow-left-line")) {
    changeContent("prev");
  } else if (event.target.classList.contains("ri-arrow-right-line")) {
    changeContent("next");
  }
});
```

Кінець лістингу 3.3

Секція «Пропозиції» поділена на два відділи. Перший це слайдер з пропозицій садка, побудований на бібліотеці Swiper.js. Слайдер використовує додаткову пагінацію для відображення кількості колонок можливих при прокрутці. Другий відділ відображає події у садку з переходом на повну сторінку з відображенням всіх актуальних подій чи інших свят у садку. На рисунку 3.4 зображено секцію «Пропозиції».

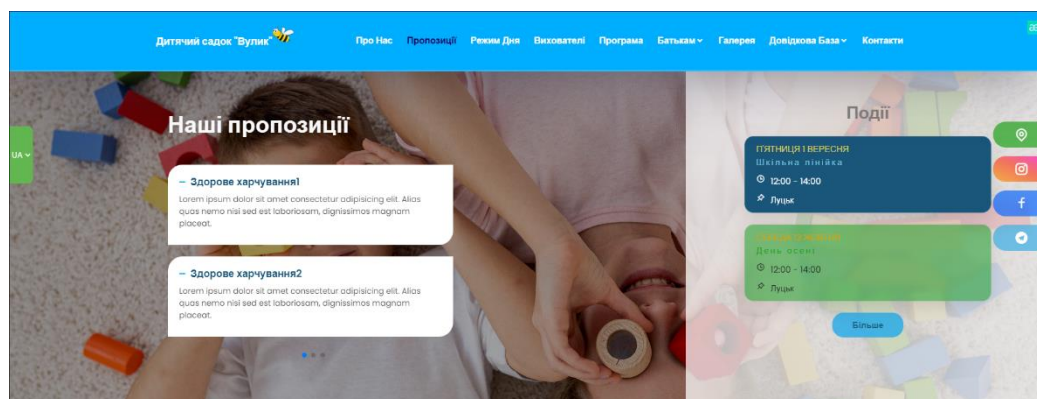


Рисунок 3.4 – Демонстрація секції «Пропозиції»

У секції «Режим дня» розміщені таблички з розкладом дня. Нижче розташовані три кнопки: «Розклад дня», «Харчування» та «Оплата». Кожна з цих кнопок відкриває модальне вікно при натисканні. У вікні «Розклад дня» та «Харчування» відображено тижневий графік у вигляді табів. На рисунку 3.5 зображено секцію «Режим дня».

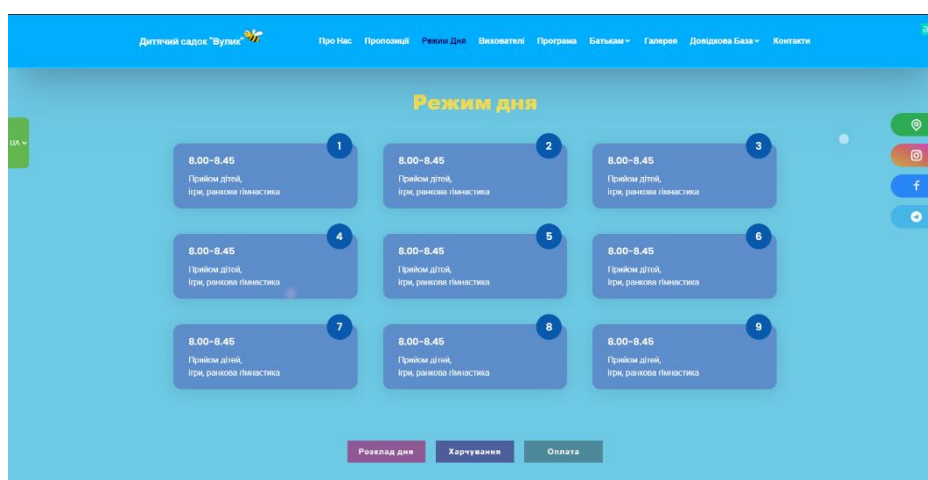


Рисунок 3.5 – Демонстрація секції «Режим дня»

У відділі «Наші вихователі» розміщена інформація про педагогів та інших освітян через модальне вікно, що показує особисті та професійні характеристики. Кожен модальний блок містить ім'я та фотографію педагога, а також його спеціалізацію. Кнопка «Більше» дозволяє отримати додаткову інформацію. На рисунку 3.6 представлена секція «Наші вихователі».

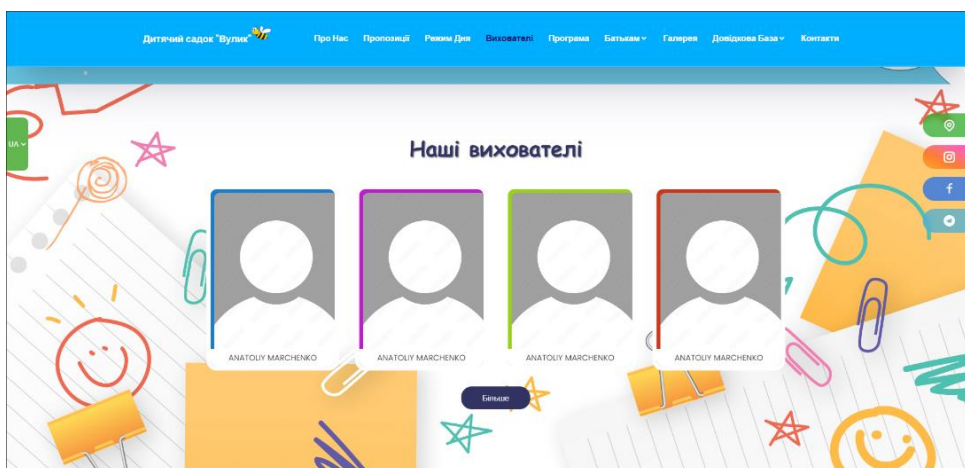


Рисунок 3.6 – Демонстрація секції «Наші вихователі»

Секція «Counter» є динамічним елементом сайту, який відображає кількість осіб у різних категоріях, таких як вихователі, учні, завідувачі, відвідувачі тощо. Цей елемент працює автоматично і починає відлік при підході користувача до його секції до певного числа. На рисунку 3.7 зображено динамічний лічильник.

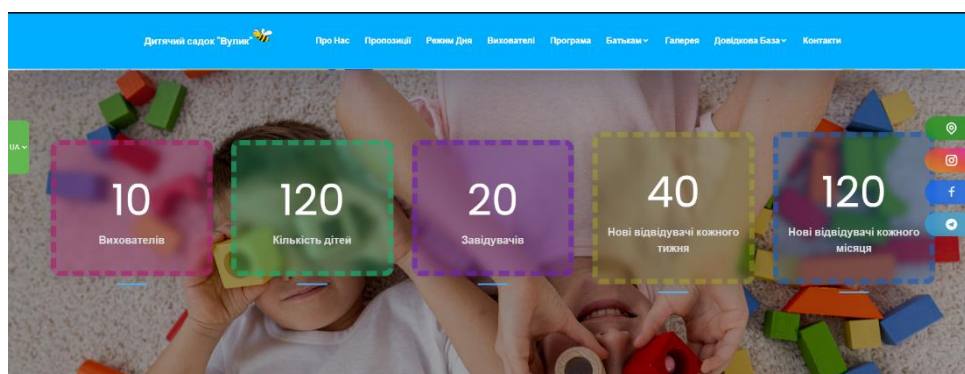


Рисунок 3.7 – Демонстрація секції «Counter»

Секція «Наша програма» пропонує детальний огляд освітньої програми для дітей дошкільного віку. Щоб зробити її більш привабливою та динамічною, було впроваджено паралакс ефект на фоні, який реагує на рух миші. Основна частина секції – це слайдер блоків з програмою садка, який дозволяє прокручувати і переглядати програму. Кожен блок містить короткий опис програми. На рисунку 3.8 зображено секцію «Наша програма».

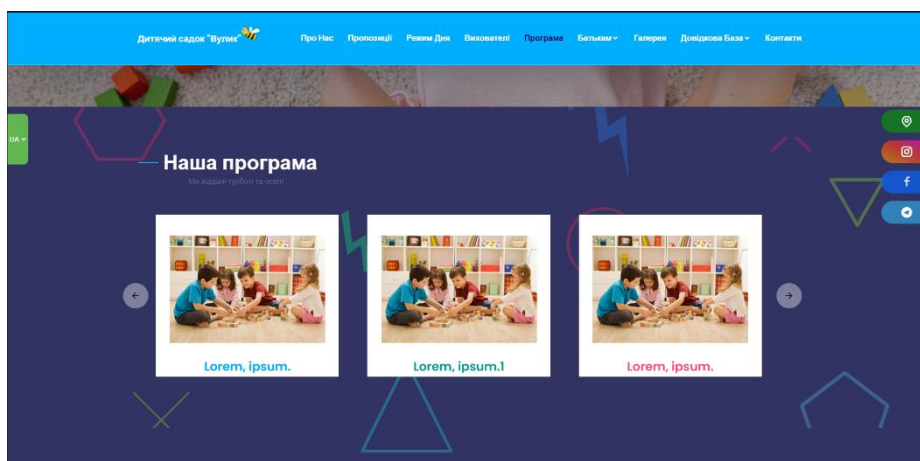


Рисунок 3.8 – Демонстрація секції «Наша програма»

Секція «Галерея» сайту пропонує користувачам можливість перегляду зображень, які відображають життя та події в садку. Ця секція розділена на різні групи, щоб дати можливість користувачам швидко знайти та переглянути ті зображення, які їх цікавлять.

У верхній частині секції розташований блок вибору груп. Користувачі можуть обирати групу, переглядаючи відповідні зображення.

Секція «Галерея» сайту використовує бібліотеку lightgallery, що надає ефективну можливість перегляду зображень. На рисунку 3.9 представлена секція з галереєю дитячого садка.

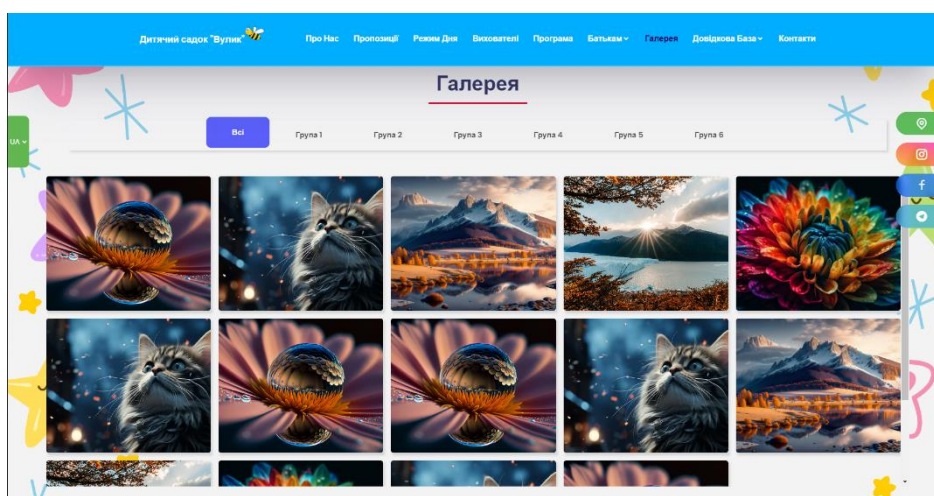


Рисунок 3.9 – Демонстрація секції «Галерея»

Остання секція головної сторінки сайту дошкільного закладу «Вулик» містить повноцінну інформацію про графік роботи закладу, його соціальні мережі, контактні дані та місцезнаходження. Також додано можливість переглядати місцезнаходження закладу на мапі, вже з доданою позначкою його розташування. На рисунку 3.10 продемонстровано секцію «Контакти».

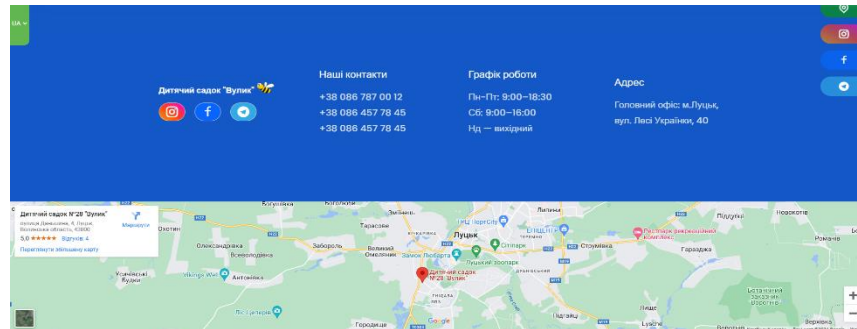


Рисунок 3.10 – Демонстрація секції «Контакти»

Окрім головної сторінки сайту, ресурс містить кілька інших важливих сторінок, серед них сторінки «Батькам», «Публічна інформація», «Мережа груп», «Нормативно-правова база», «Поради фахівців».

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

У процесі створення веб-сайту для дитячого садка «Вулик» були застосовані різні методи тестування та налагодження, щоб забезпечити надійність, продуктивність і зручність використання сайту.

Методологія тестування включала наступні частини:

- 1) ручне тестування:
 - перевірка функціональності сайту, включаючи аналіз кожної його функції щодо відповідності очікуванням та здатності виконувати призначені завдання. Наприклад, конкретні дії виконані під час тестування, охоплювали

тестування навігаційного меню, мультимедійного контенту, перевірка інтерактивних елементів, таких як слайдери, каруселі, модальні вікна, таби;

- кросбраузерне тестування. Було проведено перевірку сайту в різних популярних браузерах, таких як Google Chrome, Mozilla Firefox, Microsoft Edge та Safari, з метою забезпечити однакову зручність користування для всіх користувачів незалежно від обраного ними браузера;

- кросплатформне тестування. Перевірка відображення сайту та його інтерактивності на різних пристроях, різних екранів.

2) GitHub тестування. Створення та використання віддаленого репозиторію на GitHub для перевірки наявності помилок та багів. Це включало перевірку коректності завантаження всіх файлів, виявлення та виправлення помилок, перевірку працездатності сайту та візуалізації сайту;

3) тестування прийнятності. Під час тестування прийнятності замовник перевіряв кілька ключових аспектів продукту. Це включало оцінку функціональності, яка передбачала перевірку коректності роботи основних функцій. Далі, замовник оцінював зручність та інтуїтивність інтерфейсу, переконуючись, що користувачі з легкістю можуть здійснювати необхідні дії та знаходити інформацію. Також відбувалась перевірка відповідності дизайну бренду та цілям проекту, а також його привабливості та збереження уваги користувачів. Додатково, замовник переконувався, що продукт працює на різних пристроях та в різних браузерах без проблем, та відповідає всім вимогам, які були встановлені на етапі розробки;

4) PageSpeed Insights. Останнім інструментом для тестування сайту був онлайн-інструмент. Він аналізує різні аспекти продуктивності веб-сторінки, такі як розмір, кешування, компресія зображень, оптимізація CSS та JavaScript тощо. На рисунку 3.11 зображено результати тестування на онлайн платформі PageSpeed.

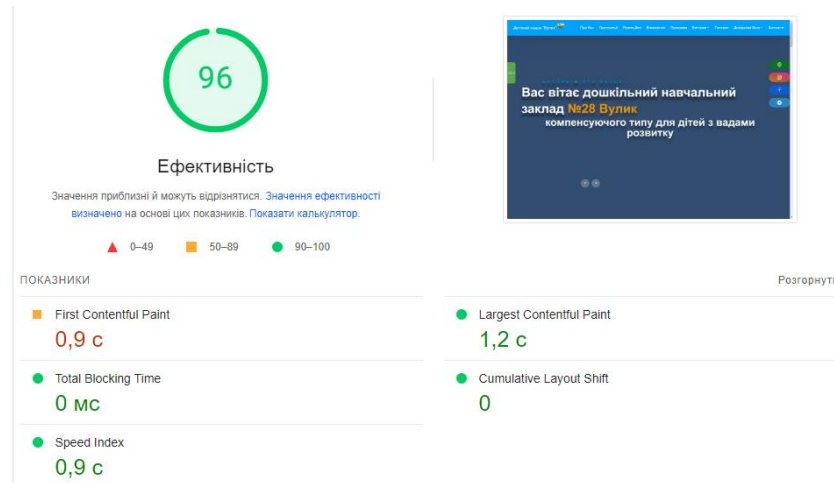


Рисунок 3.11 – Результат тестування

Висновки тестування показали, що веб-сайт дитячого садка «Вулик» є надійним, зручним у використанні та продуктивним на різних пристроях та браузерах. Рекомендації з PageSpeed Insights були враховані для подальшої оптимізації.

У процесі тестування проекту були встановлені наступні системні вимоги. Визначення системних вимог базувалися на детальному аналізі продуктивності та сумісності програмного забезпечення (ПЗ).

Мінімальні системні вимоги:

- операційна система: Windows 10, macOS, Linux;
- браузери: Google Chrome, Microsoft Edge, Safari, Opera, Firefox;
- процесор: двоядерний процесор з тактовою частотою 1.5 ГГц;
- оперативна пам'ять: мінімум 2 ГБ;
- дозвіл екрану: мінімум 1024x768px;
- швидкість інтернету: мінімум 1 Мбіт/с.

Рекомендовані системні вимоги:

- операційна система: останні версії Windows, macOS або Linux;
- браузери: Google Chrome, Microsoft Edge, Safari, Opera, Firefox;
- процесор: чотириядерний процесор з тактовою частотою 2.0 ГГц;
- оперативна пам'ять: 4 ГБ або вище;
- дозвіл екрану: Full HD (1920x1080px) або більше;

- швидкість інтернету: мінімум 10 Мбіт/с.

Під час розробки сайту було виявлено і усунуто велику кількість недоліків та проблем, що виникали у процесі створення проекту. Нижче наведено декілька із них.

1) навігаційне меню:

- проблема: неактивне розкривання навігаційного меню при кліку;
- рішення: оновлення JavaScript для правильної обробки подій та структури DOM, для забезпечення коректної роботи меню у всіх сценаріях.

2) крос-браузерна сумісність:

- проблема: некоректне відображення об'єктів у різних браузерах через різні підходи до CSS та JS;
- рішення: використання поліфілів та інструментів автоматичного додавання вендорних префіксів до CSS та трансляції JavaScript для сумісності з різними браузерами.

3.3 Оптимізація продуктивності веб-сайту

Однією з головних проблем для кожного сайту є його продуктивність. Оптимізація продуктивності включала багато аспектів, таких як мінімалізація коду, зменшення навантаження на сайт, зменшення часу очікування від користувачів, плавність інтерфейсу, оптимізація коду та ресурсів тощо.

Для досягнення вищезазначених цілей було використано декілька ефективних методів оптимізації веб-сайту.

Lazy Load. Однією з ключових технік, що було застосовано для оптимізації завантаження ресурсів сайту, є технологія Lazy Load (ліниве завантаження). Ця технологія дозволяє відкладати завантаження зображень та інших медіа-ресурсів до моменту, коли вони стають видимими для користувачів (коли користувач прокручує до певної секції). У лістингу 3.4 відображено застосування LazyLoad.

Лістинг 3.4 – Скрипти Lazy Load

```
let lazyImages = document.querySelectorAll("img[data-src]");
let parameters = {
  rootMargin: "0px",
  threshold: 0.1,
};
function handleLazyImages(entries, observer) {
  entries.forEach(function (entry) {
    if (entry.isIntersecting) {
      let lazyImage = entry.target;
      lazyImage.src = lazyImage.dataset.src;
      lazyImage.removeAttribute("data-src");
      observer.unobserve(lazyImage);
    }
  })
};
let observer = new IntersectionObserver(handleLazyImages, parameters);
lazyImages.forEach(function (lazyImage) {
  observer.observe(lazyImage);
});
```

Кінець лістингу 3.4

Асинхронне завантаження сторінки. Ще одним важливим аспектом, який був використаним під час оптимізації сайту є асинхронне завантаження скриптів. Наприклад, у секції counter, реалізовано функціонал лічильника, який оновлюється асинхронно коли користувач прогортує до певної секції. До цього моменту JS код не є видимим, чим не перезавантажує сайт під час його оновлення.

Gulp. Для автоматизації процесу оптимізації було використано інструмент gulp. За допомогою його плагінів було пришвидшено процес завантаження сторінки. Вплив на оптимізацію використаних плагінів gulp: gulp-sass. Забезпечив ефективне управління стилями та організацією коду CSS; gulp-autoprefixer. Забезпечив сумісність CSS з різними браузерами, зменшуючи необхідність ручного додавання префіксів; gulp-cssso. Зменшив розмір CSS файлів; gulp-htmlclean. Зменшив дублювання коду та спростив управління шаблонами; gulp-imagemin. Зменшив розмір зображень.

Впровадження зазначених методів призвело до значного покращення продуктивності веб-сайту. Зокрема:

- 1) зменшення часу завантаження сторінки;
- 2) зменшення обсягу даних під час завантаження сторінки;
- 3) підвищення якості користувацького досвіду;
- 4) підвищення швидкості взаємодії користувачів з сайтом;
- 5) підвищення плавності роботи інтерфейсу.

3.4 Розробка мобільної версії веб-сайту

Однією з основних вимог до розробки веб-сайту є створення його адаптивної версії для мобільних пристроїв. З огляду на потреби мого замовника – дитячого садка, було вирішено зробити сайт зручним для використання на різних мобільних пристроях, таких як смартфони, планшети тощо. Для забезпечення адаптивності веб-сайту було використано технологію CSS медіа-запитів (media queries). У лістингу 3.5 відображено демонстрацію медіа-запиту.

Лістинг 3.5 – Демонстрація медіа-запиту до меню

```
@media (max-width: 1300px) {  
  .nav {  
    position: fixed;  
    &-open-menu {  
      width: 44px;  
      position: relative;  
      left: 0px;  
      z-index: 99;  
    }  
  }  
  a:not(.sub-menu)::before {  
    opacity: 0;  
    visibility: hidden;  
  }  
}
```

Кінець лістингу 3.5

На рисунку 3.12 зображено випадаюче меню на пристроях з шириною екрана менше за 1300 пікселів.

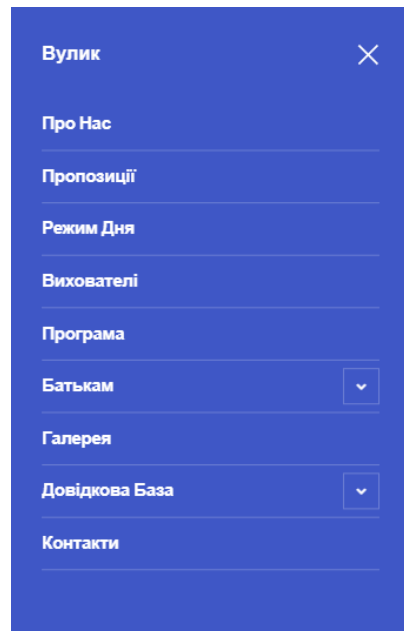


Рисунок 3.12 – Демонстрація меню «Гамбургера»

Для забезпечення плавного та приємного використання, додано анімацію відкриття та закриття меню. При натисканні на іконку «гамбургера», меню розгортається з ефектом плавного ковзання.

Висновки до розділу 3

У даному розділі були охоплені та описані всі етапи створення веб-сайту: від його програмування до його тестування та оптимізації.

На етапі практичної реалізації сайту було здійснено безпосереднє програмування сайту, включаючи розробку його основних функціональних модулів таких як інформаційні секції, динамічні елементи та інтерактивні інструменти.

Під час тестування усіх компонентів веб-сайту було виявлено та виправлено помилки, що негативно впливали на його функціональність та

зручність у користуванні. Використовувалися різні методи тестування, включаючи автоматичні та ручні тести, для забезпечення високої якості та стабільності роботи сайту.

Оптимізація продуктивності веб-сайту включала впровадження рішень для зменшення часу завантаження сторінок, підвищення швидкості обробки запитів. Застосування сучасних технологій та методів, таких як мінімізація коду, оптимізація зображень та асинхронне завантаження, дозволило досягти високих показників продуктивності.

Отже, розробка, тестування та оптимізація пройшла всі етапи та показала приємний кінцевий результат, який був продемонстрований адміністрації закладу, що викликав позитивні емоції та задоволення.

ВИСНОВКИ

Основною метою даної роботи було створення функціонального та зручного веб-сайту для дитячого садка «Вулик», який відповідатиме вимогам всіх користувачів. Реалізація цієї мети передбачала проходження всіх етапів, від аналізу предметної області до практичної реалізації та тестування сайту.

Було детально проведено аналіз сучасного стану проблеми пов'язаної з розробкою веб-сайтів даної теми. Це включало аналіз аналогів веб-сайтів садків та аналіз літературних джерел. Цей аналіз вплинув на написання сайту тим, що допоміг визначити необхідні функції, дизайн та характеристики які були реалізовані у сайтах, серед них яскравий та привабливий дизайн, геолокація, анімаційні ефекти включаючи паралакс тощо.

До розробки дизайну сайту було прикладено максимум зусиль, тому і результатом цього завдання є яскравий, привабливий дизайн, що поєднується з функціональністю та максимальною користувацькою зручністю. Для відображення такого вдалого ефекту було використано як звичайний SCSS в поєднанні з Java Script, так і додаткові бібліотеки, включаючи Animate.css.

Розроблений мною сайт забезпечує повну інформаційну доступність, надаючи батькам та відвідувачам легкий доступ до всіх необхідних даних. На сайті представлена ключова інформація про заклад дошкільної освіти, його пропозиції, актуальні події, розклад занять та харчування. Крім того, сайт містить опис кадрового складу, навчальну програму, контакти та нормативно-правову базу.

Веб-сайт був оптимізований з увагою до деталей, щоб забезпечити його коректне відображення на різних пристроях. Застосування такої технології, як медіа-запити, дозволило підлаштовувати розміри та вигляд елементів веб-сторінки залежно від характеристик пристрою. Як результат, сайт надійно працює на будь-яких пристроях, забезпечуючи зручний доступ до інформації для користувачів незалежно від обраного пристрою.

Для інтеграції карти та функціоналу геолокації на сайті була використана бібліотека Leaflet. За її допомогою було створено мапу з відміткою, для кращого розуміння місця розташування садка.

На сайті було використано різноманітні візуальні ефекти для збільшення його привабливості. Ці ефекти включали анімацію, паралакс ефект, hover ефекти, плавні переходи та світлові спливи та тіні. Використання цих візуальних прийомів допомогло створити більш привабливий та динамічний вигляд сторінок сайту, що забезпечило привабливіші враження для користувачів та сприяло залученню їх уваги до ключових елементів контенту.

Результатом роботи стало повне виконання завдання на кваліфікаційну роботу бакалавра. Було досягнуто високих кількісних та якісних показників, включаючи створення зручного інтерфейсу, швидке завантаження сторінок, функціональність, адаптивність та оптимізацію сайту. Сайт успішно інтегрує такі функції, як розклад занять, геолокацію, фотогалерею, контактну інформацію, пропозиції та програми садка а також зміну мови.

У процесі роботи я отримала нові знання, які вже використанні в наукових статтях. Ці знання допомогли мені зробити сайт кращим, додаючи нові функціональні спроможності, покращуючи його продуктивність та адаптивну здатність. Ці знання також важливі для майбутніх розробок у створенні веб-сайтів для шкіл і садків. Досвід, який я здобула, також допомагає мені у порадах щодо того, як можна покращити сайт у майбутньому, додавши нові функції, такі як реєстрація на сайті, покращення взаємодії з користувачем шляхом створення чату.

Отже, під час написання кваліфікаційної роботи було успішно виконано поставлені завдання. Результатом є функціональний, швидкий та зручний веб-сайт, який відповідає сучасним вимогам та потребам користувачів, забезпечуючи їм доступ до важливої інформації та інтерактивних можливостей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Камбек у 90-ті: як виглядали перші українські сайти. URL: <https://komanchi.com.ua/uk/kambek-u-90/> (дата звернення 10.03.2024).
2. The importance of having a website for your childcare business. URL: <https://www.parenta.com/2016/05/12/> (дата звернення: 13.03.2024).
3. Why is a website needed for preschool? URL: <https://bit.ly/3VjaOeo> (дата звернення: 15.03.2024).
4. Designing WebSites For Kids: Trends And Best Practices. URL: <https://bit.ly/4aPh8A0> (дата звернення: 16.03.2024).
5. Сайт «Світлінка». URL: <https://svitlynka.lutsk.ua> (дата звернення 20.03.2024).
6. Сайт «Лелітка». URL: <http://dnz7.lutsk.ua/> (дата звернення 20.03.2024).
7. Сайт «Twinkling Stars Preschool». URL: <https://www.twinklingstarschicago.com/> (дата звернення: 22.03.2024).
8. Життєвий цикл програмного забезпечення. URL: <https://theuppercode.com/en/blogs/1> (дата звернення: 12.04.2024).
9. Моделі життєвого циклу, принципи і метадологія розробки ПЗ. URL: <https://bit.ly/3xhQXnF> (дата звернення: 25.04.2024).
10. Ітеративна модель розробки. URL: <https://bit.ly/4cdpRx6/> (дата звернення: 30.04.2024).
11. Технологічний стек: HTML, CSS, JavaScript. URL: <http://surl.li/uheho> (дата звернення: 01.05.2024).
12. Sass vs Scss: Exploring the Benefits and Difference. URL: <https://bit.ly/4eiw38D> (дата звернення: 10.05.2024).
13. Вікіпедія Python. URL: <https://uk.wikipedia.org/wiki/Python> (дата звернення: 20.05.2024).

14. Вікіпедія PHP. URL: <https://uk.wikipedia.org/wiki/PHP> (дата звернення: 21.05.2024).
15. Why did we build Visual Studio Code? URL: <http://surl.li/ujwjb> (дата звернення: 22.05.2024).
16. Що таке DOM або об'єктна модель документа? URL: <https://foxminded.ua/dom-tse/> (дата звернення: 18.04.2024).
17. JavaScript HTML DOM (Document Object Model). URL: https://www.w3schools.com/js/js_htmlDOM.asp (дата звернення: 22.05.2024).