

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри
інтелектуальних інформаційних
систем, д-р техн. наук, проф.
_____Ю. П. Кондратенко
«_____» _____ 2023 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК СИСТЕМИ
СОЦІАЛЬНОГО ЗАХИСТУ НАСЕЛЕННЯ
МИКОЛАЇВСЬКОЇ ТЕРИТОРІАЛЬНОЇ ГРОМАДИ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21910121

Виконав студент 4-го курсу, групи 401

І. В. Скопенко

«22» червня 2023 р.

Керівник: д-р техн. наук, професор

І. М. Журавська

«22» червня 2023 р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 2022 р.

З А В Д А Н Н Я
на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Скопенку Івану Вікторовичу.

1. Тема кваліфікаційної роботи «Інформаційний вебзастосунок соціального захисту населення Миколаївської територіальної громади».

Керівник роботи Журавська Ірина Миколаївна, д-р техн наук, професор.

Затв. наказом Ректора ЧНУ ім. Петра Могили від « ____ » _____ 20__ р. № ____

2. Строк представлення кваліфікаційної роботи студентом « ____ » _____ 20__ р.

3. Вхідні (початкові) дані до роботи: результати тестування деякої групи людей.

Очікуваний результат: інформаційний вебзастосунок соціального захисту населення Миколаївської територіальної громади з використанням заданих результатів тестування.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- актуальність теми та аналіз аналогів;
- аналіз інструментів розробки сучасних вебзастосунків;

- структура та проектування власного вебзастосунок;
- тестування програмної реалізації проєкту.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Правила охорони праці при роботі з комп'ютером».

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Алексєєва Анна Олександрівна, канд. техн. наук, доцент кафедри екології Медичного інституту ЧНУ імені Петра Могили	

Керівник роботи д-р техн. наук, професор Журавська І. М.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Завдання прийнято до виконання Скопенко І. В.
(прізвище та ініціали)

_____ (підпис)

Дата видачі завдання « 23 » листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН
виконання бакалаврської кваліфікаційної роботи

Тема: Інформаційний вебзастосунок соціального захисту населення
Миколаївської територіальної громади

№ з/п	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників БКР	26.10.2022	26.10.2022	Виконано
2	Отримання завдання на виконання БКР	25.11.2022	25.11.2022	Виконано
3	Складання календарного плану роботи на весь період виконання БКР	07.12.2022	09.12.2022	Виконано
4	Отримання завдання на переддипломну практику	01.05.2023	01.05.2023	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	01.05.2023	14.05.2023	Виконано
6	Розробка звіту з переддипломної практики	15.05.2023	17.05.2023	Виконано
7	Виконання БКР: аналіз сучасного стану задачі відслідковування очей, огляд існуючих технологій, розробка ПЗ	15.05.2023	17.05.2023	Виконано
8	Попередній захист БКР на засіданні комісії кафедри	29.05.2023	30.05.2023	Виконано
9	Доробка та остаточне оформлення БКР	02.06.2023	19.06.2023	Виконано
10	Подання БКР рецензенту	15.06.2023	17.06.2023	Виконано
11	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	19.06.2023	22.06.2023	Виконано
12	Захист БКР перед екзаменаційною комісією (ЕК)	26.06.2023	29.06.2023	

Розробив студент Скопенко І. В.
(прізвище, ім'я, по батькові студента) _____ (підпис)

Керівник роботи д-р техн. наук, професор Журавська І. М.
_____ (посада, прізвище, ім'я, по батькові) _____ (підпис)

«_09_» _____ 12 _____ 2022 р.

АНОТАЦІЯ

**бакалаврської кваліфікаційної роботи студента групи 401 ЧНУ ім. Петра
Могили**

Скопенка Івана Вікторовича

**Тема: «Інформаційний вебзастосунок системи соціального захисту населення
Миколаївської територіальної громади»**

Об'єкт роботи – процес розробки інформаційних вебзастосунків, придатних для обробки чисельних одночасних запитів через розроблені форми.

Предмет роботи – програмне забезпечення для створення вебзастосунків.

Мета роботи – покращення надання соціальних послуг та забезпечення доступу населення до соціальних послуг.

Для досягнення мети необхідно вирішити наступні задачі:

- провести аналітичний огляд останніх публікацій та існуючих аналогів;
- розробити програмне забезпечення для одночасного онлайн-доступу чисельних користувачів до соціальних послуг;
- виконати тестування розробленого вебзастосунку на працездатність та навантаженість.

Розроблений інформаційний вебзастосунок у складі бази даних, вебінтерфейсу та програмного коду відповідно до отриманого завдання.

Для створення серверної частини обробки даних були застосовані такі технології, як PHP та MySQL. Клієнтський інтерфейс було створено за допомогою HTML, CSS, JS.

Кількість сторінок – 70 (без додатків), таблиць – 5, рисунків – 62, додатків – 2 та джерел посилання – 28.

Ключові слова: *вебзастосунок, PHP, HTML, JS, база даних, MySQL*

ABSTRACT

of bachelor's thesis student of group 401 of Petro Mohyla Black Sea National University

Skopenko Ivan Viktorovich

Topic: «Information web application for social protection of the population of Mykolaiv Territorial Council»

The object of work is the process of developing information web applications suitable for processing numerous simultaneous requests through the developed forms.

Subject of work - is software for creating web applications.

The purpose of the work is to develop an information web application for the Department of Labour and Social Protection, designed to improve the provision of social services and ensure public access to social services. To achieve this goal, the following tasks need to be solved:

- conduct an analytical review of recent publications and existing analogues;
- develop software for simultaneous online access to social services for numerous users;
- to test the developed web application for performance and load.

Developed informational web application consisting of a database, a web interface and program code in accordance with the assignment.

Technologies such as PHP and MySQL were used to create the server side of data processing. The client interface was created using HTML, CSS, JS.

Number of pages – 70 (without appendices), tables – 5, figures – 62, appendices – 2 and sources – 28.

Keywords: *web application, PHP, HTML, JS, database, MySQL.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	4
ВСТУП	5
1 ОГЛЯД ІНФОРМАЦІЙНОГО ВЕБЗАСТОСУНКУ СОЦІАЛЬНОГО ЗАХИСТУ НАСЕЛЕННЯ. ПОСТАНОВКА ЗАДАЧІ	7
1.1 Актуальність обраної теми	7
1.2 Огляд останніх публікацій та існуючих аналогів	9
1.3 Огляд технологій розробки інформаційних вебзастосунків	14
Висновки до розділу 1	16
2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	17
2.1 Мова програмування JS	17
2.2 HTML – основа вебсторінок	19
2.3 Таблиця стилів CSS	24
2.4 PHP	30
2.5 Бази даних	36
2.5.1 MySQL	36
2.5.2 PhpMyAdmin	38
Висновки до розділу 2	39
3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ВЕБЗАСТОСУНКУ ДОСТУПУ ДО СОЦІАЛЬНИХ ПОСЛУГ	40
3.1 Сценарії використання	40
3.2 Макет користувацького інтерфейсу	44
Висновки до розділу 3	49
4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ВЕБЗАСТОСУНКУ	50
4.1 Опис інтерфейсу програми	50

Кафедра інтелектуальних інформаційних систем	
Інформаційний вебзастосунок системи соціального захисту населення Миколаївської територіальної громади	3
4.2 Тестування вебзастосунку.....	63
Висновки до розділу 4.....	66
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
ДОДАТОК А Код програми.....	71
ДОДАТОК Б Матеріали апробації роботи.....	78

ПЕРЕЛІК СКОРОЧЕНЬ

БД	– база даних
ВПО	– внутрішньо переміщені особи
СКБД	– система керування базами даних
УСВІК	– Управління соціальних виплат і компенсації
CMS	– Content Management System
CSS	– Cascading Style Sheets
DOM	– Document Object Model
HTML	– HyperText Markup Language
MySQL	– My Structured Query Language
PHP	– Personal Home Page (Tools), Hypertext Preprocessor

ВСТУП

Соціальний захист населення є однією з найважливіших складових суспільного розвитку. Забезпечення належного рівня соціальної підтримки та доступу до соціальних послуг для громадян є однією з основних функцій держави. Однак, ефективна організація соціального захисту потребує використання сучасних інформаційних технологій та інструментів.

У сучасному цифровому світі інформаційні вебзастосунки здобувають все більшу популярність та значущість. Вони відіграють важливу роль у забезпеченні ефективної комунікації, зручного доступу до інформації та послуг, а також полегшують взаємодію між різними сторонами. Вебзастосунки у сфері соціального захисту населення можуть сприяти покращенню організації та наданню соціальних послуг, а також підвищенню якості життя громадян.

Інформаційний вебзастосунок департаменту праці та соціального захисту є складною системою, яка повинна забезпечувати надання соціального захисту громадянам.

Об'єктом роботи є процес розробки інформаційних вебзастосунків, придатних для обробки чисельних одночасних запитів через розроблені форми.

Предметом роботи є програмне забезпечення для створення вебзастосунків.

Метою роботи є розробка інформаційного вебзастосунку департаменту праці та соціального захисту, призначеного для поліпшення надання соціальних послуг та забезпечення доступу населення до соціальних послуг.

Для досягнення цієї мети необхідно виконати такі завдання:

- провести аналітичний огляд останніх публікацій та існуючих аналогів;
- розробити програмне забезпечення для одночасного онлайн-доступу чисельних користувачів до соціальних послуг;
- виконати тестування розробленого вебзастосунку на працездатність та навантаженість.

Практичне значення розробленого інформаційного вебзастосунку полягає

в забезпеченні різних функцій, такі як проведення опитування населення міста, наближення населення до надання послуг, інформування населення про можливості отримання допомог.

Основні положення, запропоновані для підвищення рівня надання населенню онлайн-послуг соціальної спрямованості, пройшли апробацію під час Всеукраїнської науково-практичної конференції «Інформаційні технології та інженерія» (м. Миколаїв, 7–10 лютого 2023 р.). За результатами опубліковано тези доповіді [1].

1 ОГЛЯД ІНФОРМАЦІЙНОГО ВЕБЗАСТОСУНКУ СОЦІАЛЬНОГО ЗАХИСТУ НАСЕЛЕННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність обраної теми

Швидкий доступ до інформації про соціальний захист та доступні послуги має велике значення для покращення якості життя людей, які перебувають у складних життєвих обставинах. Наприклад допомога внутрішньо переміщеним особам (ВПО), отримання соціальних послуг, отримання гуманітарної допомоги, тощо.

Однак, не завжди громадяни можуть знайти необхідну інформацію про соціальний захист та послуги міської ради Миколаєва, оскільки інформація розповсюджується по різних каналах та джерелах. Наприклад, вона може бути розміщена на офіційному сайті міської ради, в соціальних мережах, на окремих сайтах соціальних проєктів та програм. Такий розпорошений підхід до розміщення інформації не забезпечує ефективної допомоги та підтримки людям, які потребують соціального захисту та підтримки.

Тому, для розв'язання даної соціальної проблеми необхідно створити інформаційний вебзастосунок департаменту праці та соціального захисту населення Миколаївської міської ради, який забезпечуватиме централізований та зручний доступ до всієї необхідної інформації про соціальний захист та послуги міської ради Миколаєва.

Цей вебзастосунок дозволить громадянам швидко та легко знайти необхідну інформацію, надавати заявки на отримання соціальної допомоги та проходити анкетування через онлайн-форми, а також знаходити нову інформацію про соціальні проєкти та програми.

Також, вебзастосунок забезпечить зручний доступ до контактної інформації відповідальних осіб за соціальний захист та послуги в міській раді Миколаєва. Громадяни зможуть швидко зв'язатися з фахівцями та отримати необхідну допомогу та консультації.

Створення інформаційного вебзастосунку департаменту праці та

соціального захисту населення Миколаївської міської ради буде важливим кроком у забезпеченні ефективного та зручного доступу до необхідної соціальної інформації та послуг для всіх громадян, зокрема для тих, хто перебуває у складних життєвих обставинах через війну в Україні.

Зважаючи на те, що Україна перебуває в стані війни, соціальний захист населення є одним з найважливіших завдань держави. Багато громадян, зокрема ті, хто були вимушено покинуту зону бойових дій, мають потребу у відповідній соціальній допомозі та підтримці, однак нерідко зіштовхуються зі складнощами у пошуку необхідної інформації. Зараз виникла необхідність створення в департаменті праці та соціального захисту населення Миколаївської міської ради сервісу, якій допоможе користувачам отримувати інформацію, замовляти соціальні послуги, листуватися з владою та максимально скорочують час отримання послуг.

Одним з головних завдань вебзастосунку є доступність населення до надання соціальних послуг, інформування населення щодо можливостей соціального захисту. Відповідно до завдань вебзастосунку, він може мати різні модулі, такі як модуль обліку соціальних послуг, модуль розрахунку соціальних послуг, модуль моніторингу надання соціальних послуг, тощо. Кожен модуль повинен мати відповідну функціональність та можливість взаємодії з іншими модулями.

Нарешті, важливим елементом вебзастосунку є його інтерфейс. Він повинен бути зрозумілим та легким у використанні для різних користувачів, зокрема працівників соціальних служб, керівників та клієнтів. Інтерфейс повинен забезпечувати можливість вводу та виводу даних, зміну налаштувань та належну інформаційну підтримку користувачів.

Отже, вебзастосунок соціального захисту є важливим інструментом для поліпшення надання соціальних послуг та забезпечення доступу до них для населення. Він повинен забезпечувати ведення електронної бази даних соціальних послуг та клієнтів, моніторинг надання соціальних послуг та автоматизацію деяких процесів, пов'язаних з наданням соціальних послуг. Крім

того, вебзастосунок повинен мати різні модулі, які забезпечують різноманітну функціональність, та інтерфейс, який є зрозумілим та легким у використанні для різних користувачів. Це допоможе забезпечити більш ефективно та ефективно надання соціальних послуг для населення та поліпшити їх якість.

1.2 Огляд останніх публікацій та існуючих аналогів

Огляд останніх публікацій та існуючих аналогів інформаційних вебзастосунків соціального захисту показує, що ця тема є досить новою та досі не дослідженою належним чином. Спостерігається певна активність у створенні та впровадженні схожих застосунків у різних містах України, але науковий аналіз їх ефективності, зручності використання та задоволення потреб користувачів ще не проводився.

Одним з недавніх досліджень у цій області була стаття [2]. Авторами описується процес розробки та впровадження інформаційного застосунку для соціального захисту населення у місті Чернівці. Застосунок дозволяє користувачам отримувати інформацію про соціальні послуги, а також звертатися за допомогою онлайн (рис. 1.1).

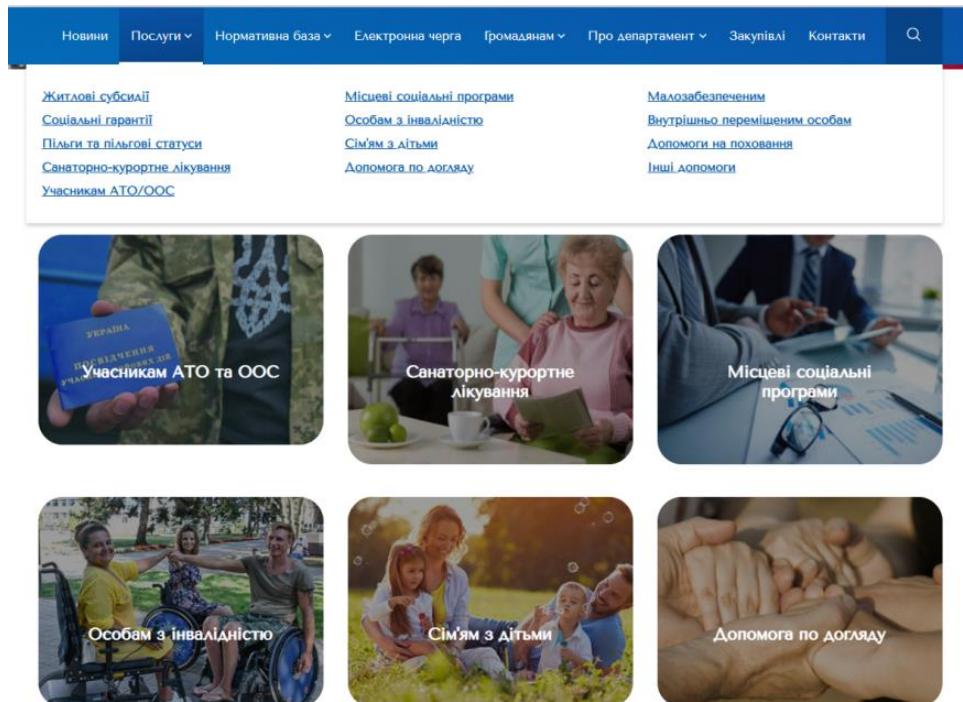


Рисунок 1.1 – Інтерфейс інформаційного застосунку соціального захисту населення у м. Чернівці

Аналізуючи інформаційний вебзастосунок можна зазначити, що в даному сайті визначені категорії громадян, яким необхідна допомога, як приклад розділ ВПО, є дуже актуальним в умовах сьогодення, але знаходиться на стадії заповнення (рис. 1.2).

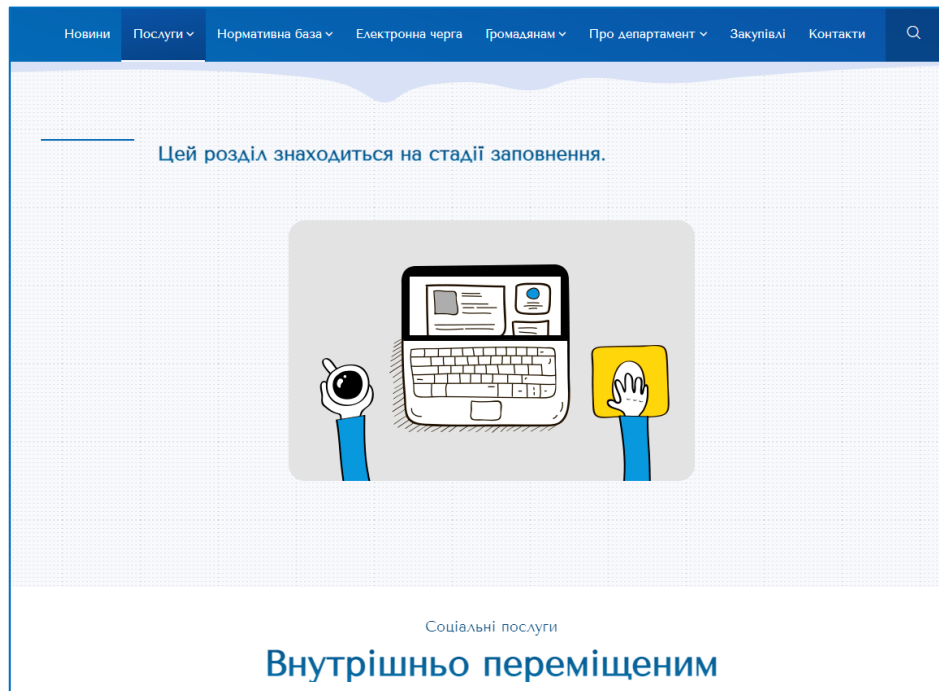


Рисунок 1.2 – Інтерфейс розділу ВПО

В розділі «Сім'ям з дітьми» перелічено всі найважливіші державні допомоги, при відкритті кожної вкладки в даному розділі, чітко визначені умови призначення допомог, але при загрузці інтерактивної вкладки завантаження заяви, завантажуються заява на призначення житлової субсидії, яка не має відношення до жодних з розділів (рис. 1.3). В цілому інтерфейс інформаційного застосунку соціального захисту населення у м. Чернівці, містить повний інформаційний ресурс для отримувачів соціальних допомог в місті Чернівці.

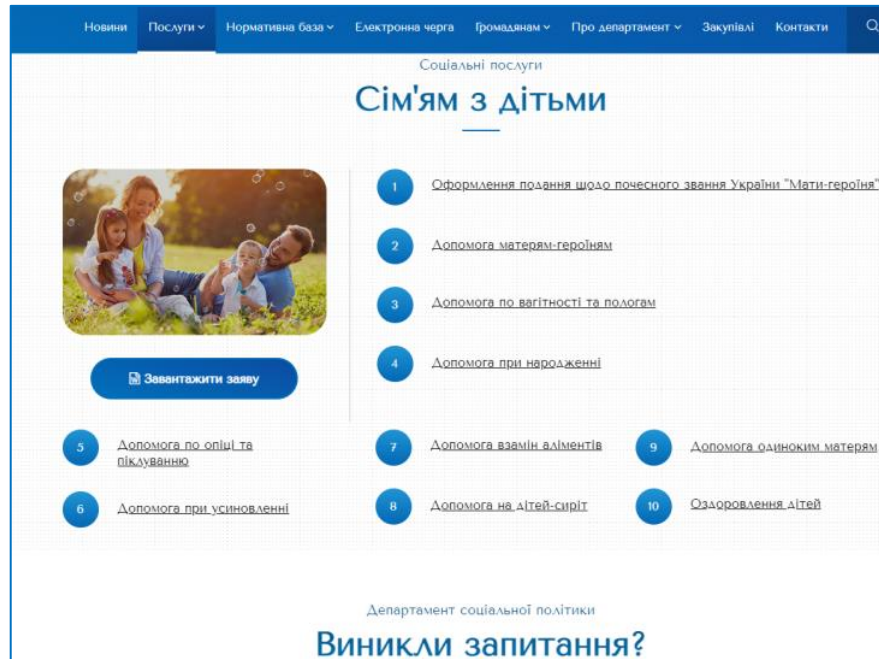


Рисунок 1.3 – Інтерфейс розділу «Сім'ям з дітьми»

Вебсайт [2] був розроблений з використанням HTML, CSS та JavaScript, щоб створити інтерактивну та корисну платформу для місцевих жителів. На сайті використовується HTML для створення основної структури та вмісту, CSS - для стилізації та розміщення вмісту на сторінках, а JavaScript – для додавання інтерактивності та динамічного вмісту, такого як анімації та переходи між сторінками. Сайт також використовує PHP та MySQL для обробки даних на сервері та зберігання даних в базі даних. Технології AJAX та jQuery використовуються для додавання додаткової функціональності та забезпечення користувачам швидкого та зручного взаємодії з сайтом. Загалом, використання цих технологій дозволяє створити простий та зрозумілий сайт з інформацією про послуги та діяльність Департаменту соціального захисту населення міста Чернівців, який забезпечує швидкий та зручний доступ до інформації для місцевих жителів та відвідувачів сайту.

Інший приклад – це інформаційний застосунок «Соціальний захист Одеси», розроблений департаментом інформації та цифрових рішень Одеської міської ради [3]. Застосунок містить інформацію про соціальні послуги, форми та порядки їх отримання, а також контакти установ, що займаються наданням

Кафедра інтелектуальних інформаційних систем
Інформаційний вебзастосунок системи соціального захисту населення Миколаївської територіальної громади 12
соціальних послуг (рис. 1.4). Застосунок також дозволяє реєструватися на отримання соціальних послуг та звертатися до фахівців з питань соціального захисту.

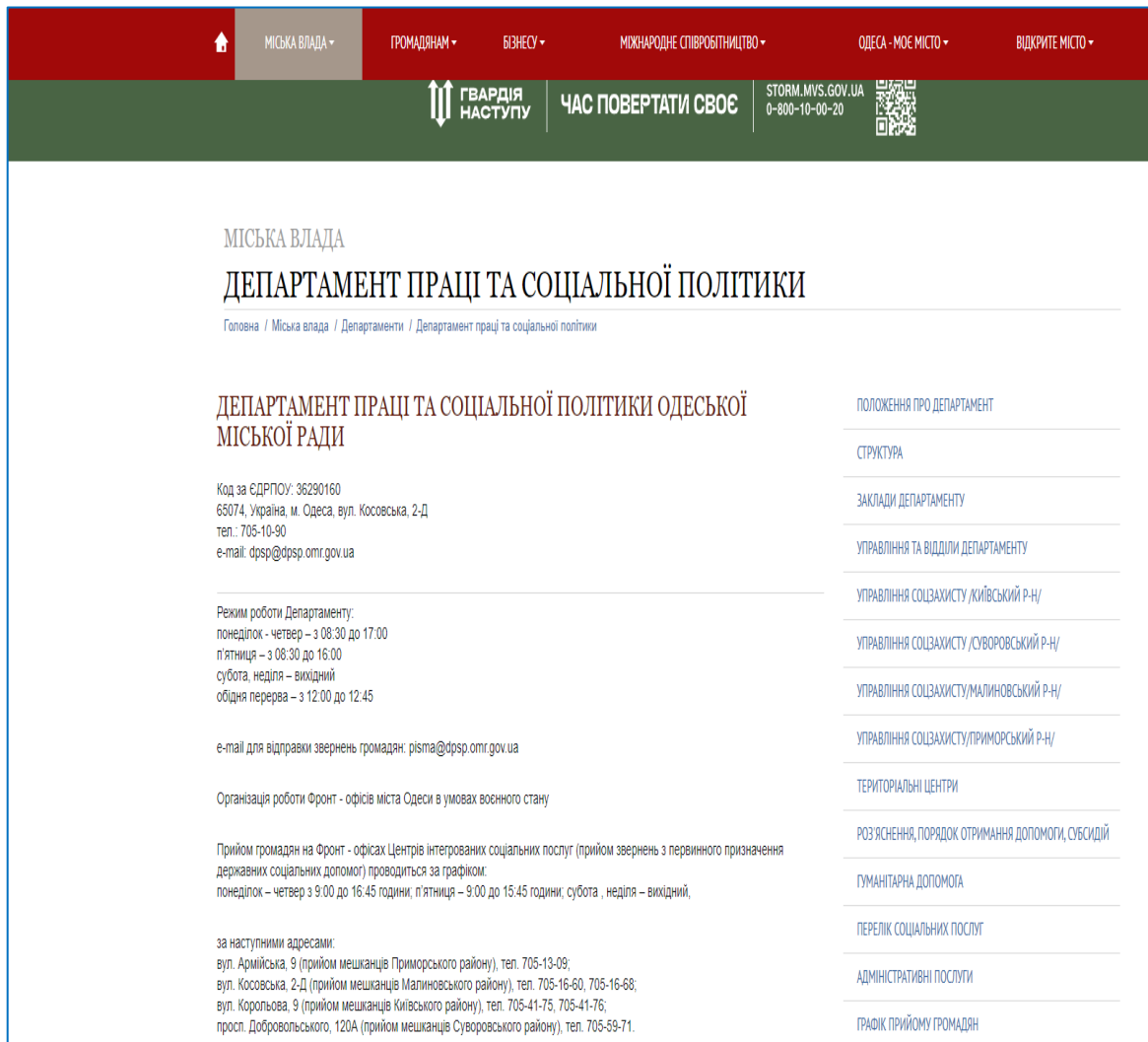


Рисунок 1.4 – Інтерфейс інформаційного застосунку

«Соціальний захист Одеси» [3]

У даному сайті відсутня інтерактивність, зазначена лише контактна інформація, структура, положення про департамент і т. д. Відсутні інформаційні матеріали щодо можливостей оформлення соціальних гарантій.

Основні технології, використані на вебсайті [3], включають HTML, CSS, JavaScript та PHP. Також використовується CMS Drupal для створення та редагування вмісту, а Bootstrap використовується для створення респонсивного дизайну та забезпечення сумісності з різними пристроями. Крім того, сайт використовує jQuery для спрощення складних задач JavaScript, таких як

маніпулювання DOM та робота з AJAX запитами. Загалом, використання цих технологій дозволяє створити простий та інформативний вебсайт з інтерактивним вмістом та респонсивним дизайном.

Надання соціальних послуг населенню можуть забезпечувати через вебсайти не тільки державні, але й комунальні установи. Таким прикладом може слугувати вебсайт комунальної установи «Сумський міський територіальний центр соціального обслуговування (надання соціальних послуг) «Берегиня» (рис. 1.5) [4]. На сторінці вебсайту викладено в повному обсязі інформація про комунальні установи соціального захисту населення м. Суми. Громадяни, які звертаються до даного вебзастосунку, мають можливість вивчити всю інформацію про надавачів соціальних послуг, але на даній сторінці відсутні інтерактивні можливості.

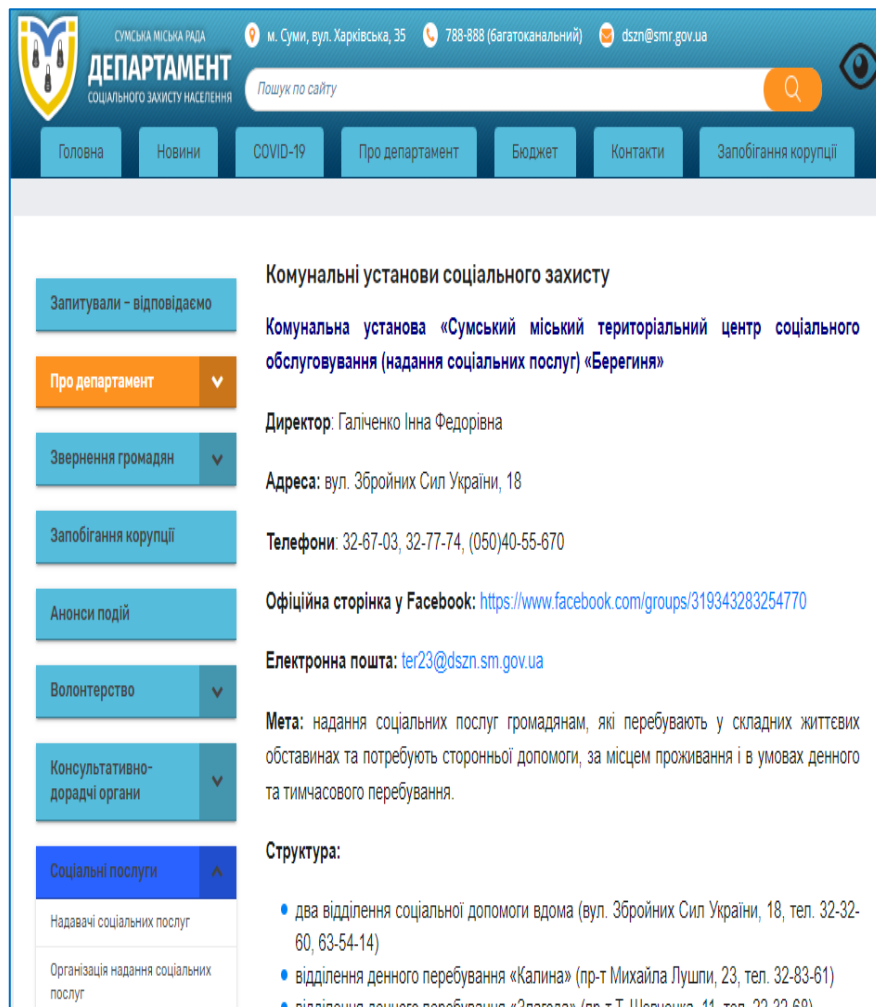


Рисунок 1.5 – Інтерфейс вебсайту Сумського міського територіального центру соціального обслуговування «Берегиня» [4]

Отже, існують деякі інформаційні вебзастосунки соціального захисту, які вже були розроблені та впроваджені у деяких містах України. Проте, дослідження їх ефективності та зручності користування ще не були проведені на достатньому рівні. Більшість існуючих статей та публікацій в основному описують процес розробки та впровадження таких застосунків, але не надають достатньої інформації про їх результативність та задоволення потреб користувачів.

Потенційні переваги інформаційних вебзастосунків установ соціального захисту, які можуть бути досліджені у майбутньому, включають поліпшення доступності та ефективності соціальних послуг, зменшення очікування на отримання допомоги, поліпшення якості надання послуг та зниження витрат на їх надання. Також можливе забезпечення більшої прозорості та відкритості у роботі установ соціального захисту та покращення взаємодії між ними та користувачами. Вебсайт [4] використовує різноманітні технології розробки для створення та редагування свого вмісту. До цих технологій належать HTML, CSS та JavaScript. Також для створення інтерактивного вмісту. PHP використовується для обробки даних на сервері та взаємодії з базою даних. Крім того, для спрощення роботи з JavaScript використовується бібліотека jQuery. Окрім цього, використовується CMS WordPress для забезпечення легкості управління вмістом сайту. Загалом, застосування цих технологій допомагає створювати простий та інформативний вебсайт з інтерактивним вмістом та легким управлінням вмістом.

У цілому, дослідження інформаційних вебзастосунків установ соціального захисту є важливим напрямком, який може допомогти у поліпшенні якості та доступності соціальних послуг для населення. Необхідно проводити детальні дослідження щодо ефективності та зручності користування такими застосунками, а також розробляти їх з урахуванням потреб користувачів та специфіки роботи установ соціального захисту.

1.3 Огляд технологій розробки інформаційних вебзастосунків

Для вирішення поставленої задачі можна використовувати різні технології,

методи та підходи, включаючи HTML, CSS, JavaScript, PHP, MySQL та фреймворки, такі як Symfony.

HTML та CSS є основними мовами розмітки та стилізації вебсторінок. Вони дозволяють створювати структуру вебзастосунку та його дизайн. HTML використовується для створення різних елементів вебсторінки, таких як заголовки, текст, таблиці та інші елементи. CSS використовується для стилізації цих елементів та надання їм вигляду та розміру.

JavaScript є мовою програмування, яка використовується для створення динамічних вебсторінок та інтерактивних елементів. JavaScript дозволяє додавати різні функції та взаємодію з користувачем, такі як валідація форм, відображення попереджень та повідомлень.

PHP є мовою програмування, яка дозволяє створювати бекенд вебзастосунку та взаємодію з базою даних (БД). PHP зазвичай використовується для створення динамічних вебсторінок, роботи з формами та БД. PHP дозволяє розробникам створювати бекенд-код вебзастосунку та забезпечувати взаємодію між фронтендом та БД.

MySQL є однією з найпопулярніших реляційних баз даних, яка використовується для збереження даних вебзастосунку. MySQL дозволяє зберігати дані в табличному форматі та забезпечувати ефективний доступ до цих даних з бекенду вебзастосунку. Використання MySQL дозволяє забезпечити стійкість та безпеку зберігання даних, а також зручний доступ до них.

Symfony є одним з найпопулярніших PHP-фреймворків та використовується для створення складних вебзастосунків. Symfony дозволяє розробникам швидко та ефективно створювати бекенд-код вебзастосунку, забезпечити взаємодію з БД та створювати різні функції, такі як авторизація, аутентифікація та інші. Symfony також забезпечує високий рівень безпеки та стійкості вебзастосунку, що є важливим для інформаційних систем соціального захисту.

Висновки до розділу 1

У розділі 1 було проведено аналіз предметної області, обгрунтовано актуальність обраної теми.

Також був проведений аналіз наявних аналогів. На основі проведеного аналізу було виявлено, що інформаційні сторінки широко використовуються. Кожен аналог має свої переваги та недоліки. Розробка власного вебзастосунку дозволить викоренити більшість недоліків, виявлених в існуючих аналогах.

2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Мова програмування JS

Неможливо уявити сучасну вебсторінку без застосування такої мови програмування, як JavaScript (надалі – JS). JS є однією з найпопулярніших та найважливіших мов програмування для веброзробки. За допомогою JS можна створювати динамічні та інтерактивні вебсторінки, розширювати функціональність вебзастосунків та взаємодіяти з користувачами.

JavaScript – це легка об'єктно-орієнтована мова програмування, яка використовується багатьма вебсайтами для написання сценаріїв вебсторінок. Це інтерпретована, повноцінна мова програмування, яка забезпечує динамічну інтерактивність на вебсайтах при застосуванні до HTML-документа. Вона була введена в 1995 році для додавання програм на вебсторінки в браузері Netscape Navigator. З тих пір вона була прийнята всіма іншими графічними веббраузерами. За допомогою JavaScript користувачі можуть створювати сучасні вебзастосунки для безпосередньої взаємодії без необхідності щоразу перезавантажувати сторінку. Традиційний вебсайт використовує js для забезпечення декількох форм інтерактивності та простоти.

Хоча JavaScript не має жодного зв'язку з мовою програмування Java. Назва була запропонована і надана в часи, коли Java набирала популярність на ринку. Окрім веббраузерів, баз даних, таких як CouchDB і MongoDB, використовують JavaScript як мову сценаріїв і запитів [12].

JavaScript має широкий спектр функцій, які роблять її потужною та універсальною мовою програмування. Основні функції включають:

- маніпуляцію DOM (Document Object Model): JavaScript дозволяє змінювати вміст, стиль та структуру вебсторінок, що дає змогу створювати динамічні та інтерактивні ефекти;

- обробка подій: JavaScript дозволяє встановлювати обробники подій, такі як натискання кнопок, наведення курсору, відправлення форми. Це дозволяє реагувати на дії користувача та забезпечувати взаємодію;

- взаємодія з сервером: JavaScript може виконувати запити до сервера без перезавантаження сторінки за допомогою технологій, таких як AJAX та Fetch API. Це дозволяє створювати асинхронні застосунки та отримувати дані з сервера в реальному часі;

- робота з «кукі» та локальним сховищем: JavaScript дозволяє зберігати дані на стороні клієнта за допомогою кукі (англ. cookies) та локального сховища (localStorage та sessionStorage). Це дозволяє зберігати стан застосунку та виконувати розрахунки без звернення до сервера.

JavaScript надає широкі можливості для розробників. Ось декілька з них:

- розробка мобільних застосунків: За допомогою фреймворків, таких як React Native та Ionic, JavaScript може бути використана для розробки мобільних застосунків для платформ Android та iOS. Це дозволяє розробникам використовувати свої знання JavaScript для побудови кросплатформних застосунків;

- розширення браузерів: JavaScript може бути використана для створення розширень та застосунків для різних браузерів. Це дозволяє розширити функціональність браузера та надати користувачам нові можливості.

- розробка серверних застосунків: За допомогою платформи Node.js, яка базується на JavaScript, розробники можуть створювати серверні застосунки та API. Це дозволяє використовувати JavaScript як мову програмування як на стороні клієнта, так і на стороні сервера, що дозволяє створювати повноцінні повностекові застосунки;

- розробка ігор та графічних програм: JavaScript використовується для створення вебігор, включаючи HTML5-ігри, а також для реалізації графічних програм та анімації.

Незважаючи на свою популярність і широкі можливості, JS також має деякі недоліки, які варто враховувати. Ось декілька прикладів:

– вразливість до помилок: JavaScript допускає легкі помилки, такі як опечатки, неправильна типізація та інші синтаксичні помилки, які можуть викликати проблеми у виконанні коду. Це може бути особливо проблематичним у великих проєктах, де важко виявити й виправити помилки;

– проблеми з безпекою: JavaScript запускається на клієнтській стороні, що робить його вразливим до атак та зловмисного використання. Недоброзичливі користувачі можуть впливати на код JavaScript, використовуючи техніки, такі як впровадження шкідливого коду або крадіжка даних;

– різна підтримка браузерами: Різні браузери можуть мати різну реалізацію JavaScript і підтримку його функцій та функціональності. Це може вимагати додаткових зусиль для тестування та забезпечення сумісності з різними браузерами;

– проблеми з продуктивністю: JavaScript є інтерпретованою мовою, що може впливати на продуктивність застосунків, особливо у випадку обробки великих обсягів даних. Велика кількість обчислень та складні анімації можуть сповільнювати роботу вебсторінок;

– обмеження безпеки між доменами: З міркувань безпеки, JavaScript обмежує доступ до інформації, що знаходиться на інших доменах. Це накладає обмеження на спілкування та обмін даними між різними доменами.

2.2 HTML – основа вебсторінок

HTML – мова, якою написано більшість вебсайтів. Вона використовується для створення сторінок та надання їм функціональності.

HTML вперше створили Тім Бернерс-Лі, Роберт Кайо та інші, починаючи з 1989 року. Це означає Hyper Text Markup Language [15]. «Гіпертекст» означає, що документ містить посилання, які дозволяють читачеві перейти до інших місць документа або взагалі до іншого документа. Остання версія відома як HTML5. Мова розмітки – це спосіб, за допомогою якого комп'ютери спілкуються один з одним, щоб контролювати, як текст обробляється та представляється. Для цього

HTML використовує дві речі: теги та атрибути. Вони є основою HTML. Вони працюють разом, але виконують різні функції. Теги – основні одиниці, з яких складається HTML-документ. Вони вкладаються в кутові дужки, наприклад «<p>». Теги бувають ті, які треба закривати (парні), і ті, які не потрібно закривати (непарні). Парний тег має вигляд «<p></p>». Між відкриваючим і закриваючим тегом може бути текст. Непарний тег має вигляд «». Також теги поділяють на блокові та рядкові. Усю інформацію про елементи та їх характеристику можна знайти у офіційній документації. Тут наведено лише декілька прикладів блокових та рядкових тегів. Блокові елементи: <html>, <div>, <body>, <p>, <h1>-<h6>, , . Рядкові елементи: <a>, ,
, <code>, <input>. Кожна вебсторінка повинна мати обов'язкову структуру, яка зображена на рис. 2.1.



```
1.html > html > head > meta
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10 |
11 </body>
12 </html>
```

Рисунок 2.1 – Базова структура HTML

Кожна сторінка повинна містити базову структуру, яка зображена на рис. 2.3. Елемент «<!DOCTYPE html>» визначає тип документа. Тег «<html>» є корневим, так як містить в собі інші елементи. «<head>» визначає конфігурацію документа: заголовок, кодування, масштабування тощо. Елемент «<body>» містить у собі структуру документа: тексти, картинки, відео, аудіо тощо.

Існує безліч редакторів коду, які представляють свої інструменти та методи для створення вебсторінок: WebStorm, Visual Studio Code, Sublime Text тощо.

Блокові формують видимий блок на сторінці. Вони будуть на новому рядку

після будь-якого контенту, і будь-який контент після них теж буде на новому рядку. Блокові елементи не можуть бути вкладені в рядкові елементи, але іноді можуть вкладатися в інші елементи блокового рівня. Вони займають весь доступний простір по горизонталі.

Рядкові елементи – це ті, які знаходяться в блокових елементах та оточують лише малі частини документа. Вони не призводять до появи нового рядка у документі. Блокові елементи мають таку структуру, яка зображена на рис. 2.2.

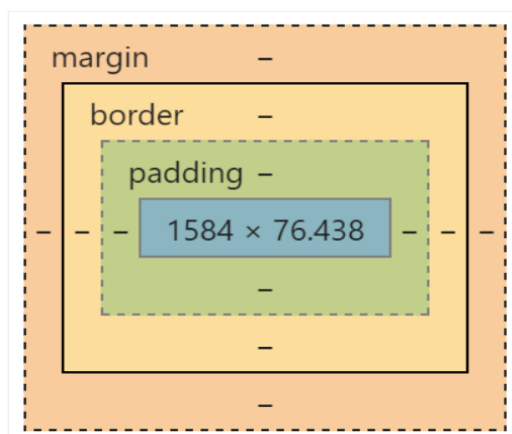


Рисунок 2.2 – Структура блокового елемента HTML:

- 1) margin – зовнішній відступ;
- 2) border – рамка елемента;
- 3) padding – внутрішній відступ

Синім на рис. 2.2 позначено контент елемента. Кожна частина налаштовується окремо. Margin, border, padding мають верхню, нижню, праву та ліву границі.

Атрибути містяться всередині тегу, та визначають додаткову інформацію про тег.

HTML5 – це 5-та версія HTML. Завдяки новим можливостям HTML5 можливо створювати динамічні вебсайти. З випуском HTML5 було введено безліч нових можливостей та функціональностей, які значно полегшили розробку та покращили користувацький досвід.

Теги Audio та Video – це два основних доповнення до HTML5. Вони

дозволяють розробникам вбудовувати відео або аудіо на свій сайт. Відео в HTML5 може використовувати CSS і CSS3 для стилізації тегу відео. У ньому можна змінювати межі, непрозорість, відображення, градієнти, переходи, трансформації і навіть анімацію. HTML5 робить додавання відео надшвидким і не вимагає створення відеоплеєра. Це економить час для розробника і пропонує клієнту більш якісне і доступне рішення. На рис. 2.3 представлено приклад тегу Audio та Video.

```
<!DOCTYPE html>
<html>
<body>
<h2>Example of video and audio tag</h2>

<video width = "300" height = "200" controls autoplay>
  <source src = "/html5/foo.ogg" type = "video/ogg" />
  <source src = "/html5/foo.mp4" type = "video/mp4" />
  Your browser does not support the video element.
</video>

<audio controls autoplay>
  <source src = "/html5/audio.ogg" type = "audio/ogg" />
  <source src = "/html5/audio.wav" type = "audio/wav" />
  Your browser does not support the audio element.
</audio>
</body>
</html>
```

Рисунок 2.3 – Тег Audio та Video

Також HTML5 дозволяє використовувати елемент `<figure>` для розмітки фотографії в документі та елемент `<figcaption>` для визначення підпису до фотографії. Тег `<figcaption>` визначає підпис для елемента `<figure>`. Цей тег надає контейнер для вмісту, еквівалентного рисунку. Його можна використовувати для групування підпису з одним або декількома зображеннями, блоком коду або іншим вмістом. На рис. 2.4 представлено відповідний приклад.

```
<figure>
  
  <figcaption>Fig.1 - Geeksforgeeks.</figcaption>
</figure>
```

Рисунок 2.4 – Елементи `<figure>`, `<figcaption>`, `<figure>`

Тег `<nav>` визначає набір навігаційних посилань. Він використовується для частини інтернет-сайту, яка посилається на різні сторінки сайту. Гіперпосилання можуть бути організовані за допомогою декількох підходів. Поширеними прикладами навігаційних елементів є меню, таблиці, зміст і покажчики. Цей елемент значно полегшує створення навігаційного меню, створює акуратне горизонтальне меню текстових посилань і допомагає програмам для зчитування з екрану правильно визначати основні навігаційні області в документі. На рис. 2.5 представлено приклад тегу `<nav>`.

```
<h1> HTML Nav tag</h1>
<nav>
  <a href="/html/">HTML</a>
  <a href="/css/">CSS</a>
  <a href="/js/">JavaScript</a>
  <a href="/jquery/">jQuery</a>
</nav>
```

Рисунок 2.5 – Тег `<nav>`

HTML має широкі можливості для розробників. Ось декілька з них:

- мультимедійні можливості: HTML5 дозволяє безпосередньо вбудовувати вебвідео та вебаудіо без необхідності використання сторонніх плагінів. Він також підтримує векторну графіку та анімацію;
- локальне сховище: HTML5 включає можливості локального сховища, такі як `localStorage` та `sessionStorage`, що дозволяють зберігати дані на боці клієнта для підтримки офлайн-застосунків та збереження стану застосунків;
- геолокація: HTML5 надає можливість отримувати географічні координати користувача, що дозволяє розробляти застосунки, засновані на визначенні місцезнаходження.

Незважаючи на свою популярність та широке застосування, HTML має деякі недоліки:

- обмежені можливості стилізації: HTML забезпечує базові функції стилізації, але для більш складних та затребуваних дизайнів часто потрібно використовувати CSS (Cascading Style Sheets);

– сумісність між браузерами: Різні браузери можуть трактувати HTML-код по-різному, що може призвести до відображення вебсторінок по-різному на різних браузерах;

– вразливість до атак: Як і будь-який вебзасіб, HTML також може бути вразливим до атак, таких як впровадження шкідливого коду або крадіжка даних. Розробники повинні приділяти увагу безпеці та застосовувати відповідні заходи безпеки.

HTML продовжує бути найважливішою мовою розмітки для веброзробки, забезпечуючи основу для створення вебсторінок та взаємодії з користувачами.

2.3 Таблиця стилів CSS

Cascading Style Sheets (CSS) – каскадна таблиця стилів, яка використовується для представлення зовнішнього вигляду документа, написаного на HTML або XML. Якщо HTML призначався для опису вмісту вебсторінки, то CSS – для стильового форматування цієї сторінки. Якби HTML-документ містив будову вебсторінки та ще й колір, розмір шрифтів, відступів тощо, то це був би величезний документ, який дуже важко читати та підтримувати. Тому Консорціум World Wide Web Consortium (W3C) створив CSS. Стилї зазвичай зберігають у зовнішніх файлах з розширенням «.css». Завдяки такому зовнішньому файлу можна легко змінювати форматування сайту, змінивши лише один або декілька файлів. Це значно спрощує та прискорює створення вебсайтів.

CSS складається з правил. Правило складається з селектора та блоку оголошення. Синтаксис наведено на рис. 2.6.

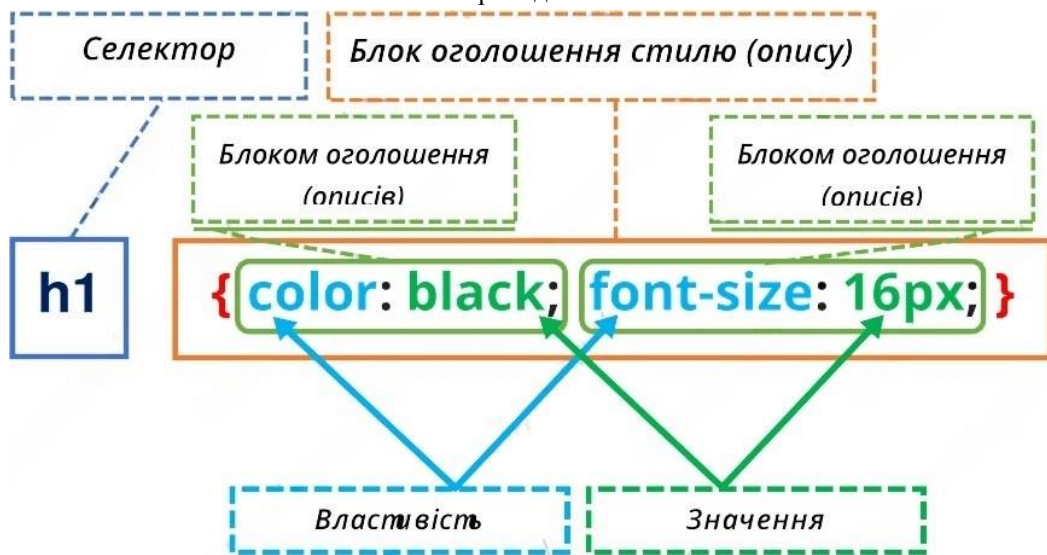


Рисунок 2.6 – Синтаксис CSS

Селектор вказує на елемент HTML, який потрібно створити. Блок декларації містить одну або кілька декларацій, розділених крапкою з комою. Кожна декларація містить назву властивості CSS і значення, розділені двокрапкою. Кілька декларацій CSS розділяються крапкою з комою, а блоки декларацій оточені фігурними дужками.

На рис. 2.7 наведено приклад використання CSS-правил.

```

p{
  color: green;
  text-align: center;
}
    
```

Рисунок 2.7 – CSS-правило:

- «p» – це селектор, що вказує на елемент, який треба стилізувати;
- «color» – властивість, а «green» – її значення;
- «text-align» – властивість, а «center» – її значення

Селектори бувають декількох типів:

- прості селектори;
- селектори комбінаторів;
- селектори псевдокласів;
- селектори псевдоелементів;
- селектори атрибутів.

Прості селектори обирають елементи на основі назви, ідентифікатора або класу. На рис. 2.8–2.10 наведено приклади таких селекторів.

```
p{
  text-align: center;
  color: green;
}
```

Рисунок 2.8 – Селектор, який обирає елемент на основі назви

```
#para1 {
  text-align: center;
  color: green;
}
```

Рисунок 2.9 – Селектор, який обирає елемент зі значенням id para1

```
.centre {
  text-align: center;
  color: green;
}
```

Рисунок 2.10 – Селектор класу

Комбінатор – це те, що пояснює зв'язок між селекторами. Селектор CSS може містити більше одного простого селектора. Між простими селекторами ми можемо включити комбінатор. Розрізняють такі комбінатори: нащадків (пробіл), дочірній («>»), сусідній («+»), братів і сестер («~»).

Комбінатор нащадків відповідає всім елементам, які є нащадками зазначеного елемента. Дочірній комбінатор обирає усі елементи, які є дочірніми для зазначеного елемента. Сусідній комбінатор використовується для вибору елемента, який знаходиться безпосередньо після іншого певного елемента. Ці два елемента повинні знаходитися всередині одного батьківського елемента. Загальний комбінатор братів і сестер обирає усі елементи, які є наступними братами і сестрами зазначеного елемента. На рис.2.11–2.14 наведено приклади використання комбінаторів.

```
div p {  
  background-color: blue;  
}
```

Рисунок 2.11 – Селектор нащадків

```
div > p {  
  background-color: blue;  
}
```

Рисунок 2.12 – Селектор дочірній

```
div + p {  
  background-color: blue;  
}
```

Рисунок 2.13 – Родинний селектор

```
1 div ~ p {  
2   background-color: blue;  
3 }
```

Рисунок 2.14 – Загальний батьківський селектор

Псевдоклас – це ключове слово, яке додається до селектора та встановлює його особливий стан. Наприклад, «:hover» може бути використан для зміни кольору кнопки при наведенні вказівника на неї. На рис. 2.15 наведено приклад такого псевдокласу.

```
div:hover {  
  background-color: blue;  
}
```

Рисунок 2.15 – Псевдоклас «:hover»

Псевдоелемент CSS використовується для стилізації певних частин елемента. З їх допомогою можна стилізувати першу літеру або рядок елемента, вставити вміст до або після вмісту елемента тощо. На рис. 2.16 наведено приклад псевдоелементу.


```
p::first-line {  
  color: blue;  
  font-variant: small-caps;  
}
```

Рисунок 2.16 – Псевдоелемент «::first-line»

На рис. 2.16 зображено псевдоелемент, який використовується для додавання спеціального стилю до першого рядка тексту. Зверніть увагу на позначення подвійної двокрапки. Подвійна двокрапка замінила позначення з однією двокрапкою для псевдоелементів у CSS3. Це була спроба W3C розрізнити псевдокласи та псевдоелементи.

Селектор атрибуту використовується для виділення елементів із заданим атрибутом. На наступному рис. 2.17 наведено приклад такого селектора.

```
a[target] {  
  background-color: blue;  
}
```

Рисунок 2.17 – Селектор, який обирає всі «<a>» з атрибутом target

Також існує універсальний селектор CSS, який впливає на всі елементи документу без винятку. Він показаний на рис. 2.18.

```
* {  
  text-align: center;  
  color: blue;  
}
```

Рисунок 2.18 – Універсальний селектор

Іноколи буває так, що необхідно застосувати одні і ті ж стилі для декількох елементів. Для уникнення дублювання коду використовується групування селекторів. На рис. 2.19 наведено приклад такого групування.

```
h1,h2,p {  
  text-align: center;  
  color: blue;  
}
```

Рисунок 2.19 – Групування селекторів

Коли браузер прочитає таблицю стилів, він відформатує HTML- документ відповідно до інформації в таблиці стилів. Існує три способи вставки таблиці стилів: зовнішній, внутрішній, вбудований.

Зовнішній CSS зберігається в окремому файлі, внутрішній – в тегу «<style>», який знаходиться всередині «<head>». Приклад внутрішнього стилю зображено на рис. 2.20.

```
tml > ...  
<head>  
<style>  
  body {  
    background-color: linen;  
  }  
  
  h1 {  
    color: blueviolet;  
    margin-left: 40px;  
  }  
</style>  
</head>
```

Рисунок 2.20 – Приклад застосування внутрішнього стилю

Вбудований стиль може використовуватися для застосування унікального стилю для окремого елемента. Щоб його застосувати, необхідно додати атрибут style до відповідного елемента. Приклад застосування вбудованого стилю зображено на рис. 2.21.

```
1 <h1 style="color: brown;text-align:center;">This is heading</h1>  
2 <p style="color: aqua;">This is paragraph.</p>
```

Рисунок 2.21 – Приклад застосування вбудованого стилю

Вбудований стиль необхідно застосувати обережно, адже він руйнує ідею

розділення на стилі та вміст сторінки, змішуючи їх в одному файлі. У випадку, якщо є декілька варіантів стилів, CSS-стилі мають таку ієрархію пріоритетів:

- 1) вбудований стиль;
- 2) зовнішні та внутрішні таблиці стилів;
- 3) стилі браузера за замовчуванням.

Тобто, стилі з найбільшим пріоритетом будуть застосовані до визначеного елемента. Таким чином, вбудований стиль має найвищий пріоритет і замінить зовнішні та внутрішні стилі та параметри браузера за замовчуванням.

Специфіка – ранг/оцінка, що визначає, яке оголошення стилю в кінцевому підсумку буде застосовано до елемента. Якщо є два або більше правил CSS, які вказують на один і той самий елемент, селектор з найвищим значенням специфічності буде мати пріоритет, а його стилі будуть застосовані до цього елемента HTML. Кожен селектор CSS займає своє місце в ієрархії специфіки. Існує чотири категорії, які визначають рівень специфічності селектора:

- 1) вбудовані стилі (значення специфічності – 1 000);
- 2) ідентифікатори (значення специфічності – 100);
- 3) класи, псевдокласи, селектори атрибутів (значення специфічності – 10);
- 4) елементи та псевдоелементи (значення специфічності – 10).

Для розрахунку специфічності конкретного правила, потрібно знайти суму специфічності усіх селекторів цього правила.

2.4 PHP

PHP (англ. PHP: Hypertext Preprocessor – «PHP: препроцесор гіпертексту») – скриптова мова програмування, застосовується для розробки застосунків та сайтів. На даний час він підтримується майже всіма хостинг-компаніями і є одним із лідерів серед мов програмування, які застосовуються для написання динамічних вебсторінок. Ця мова також підтримується нашими серверами хостингу [17].

PHP – це мова програмування з відкритим вихідним кодом, розроблена для вебу, яка може бути впроваджена в HTML-код. PHP простий для навчання, тому вже через кілька годин на ознайомлення з функціоналом PHP можливо писати прості скрипти мовою PHP.

Історія PHP починається з 1995 року, коли Рasmus Лердорф (англ. Rasmus Lerdorf) створив простий застосунок мовою Perl, що аналізував відвідування користувачами його резюме на вебсайті. Потім, коли цим застосунком вже користувалися кілька чоловік, а число охочих одержати його постійно збільшувалося, Лердорф дав ім'я своєму творінню «Personal Home Page Tools» (версія 1) і виставив для вільного завантаження. З цієї миті почався небувалий зліт популярності PHP [18].

Основним призначенням PHP є обробка даних на вебсервері. Виконуючись на сервері, PHP генерує HTML, що надсилається до клієнтського браузера, що дозволяє створювати динамічний контент. PHP може взаємодіяти з базами даних, обробляти форми, керувати файлами на сервері та виконувати багато інших завдань на серверному рівні.

Основні риси PHP включають:

1) синтаксис: PHP базується на синтаксисі C, що робить його легким для вивчення і розуміння для програмістів, які вже володіють знаннями C, C++ або Java;

2) вбудована підтримка баз даних: PHP має вбудовану підтримку для багатьох типів баз даних, включаючи MySQL, PostgreSQL, SQLite та інші. Це дозволяє легко взаємодіяти з базами даних і виконувати операції, такі як зчитування, запис і оновлення даних;

3) розширені можливості: PHP має велику кількість вбудованих функцій і розширень, що спрощують роботу з вебсервером, файлами, текстом, зображеннями і багато іншого. Це дозволяє розробникам швидко вирішувати різноманітні завдання і скорочує час розробки;

4) переносимість: PHP підтримується багатьма операційними системами, такими як Windows, Linux, macOS і інші. Код PHP може працювати на будь-якому вебсервері, який підтримує PHP;

5) велике співтовариство: PHP має широке співтовариство розробників, що означає, що завжди є багато документації, підручників, форумів і розширень, які можна використовувати для отримання допомоги та підтримки.

Недоліки PHP:

1) швидкодія: В порівнянні з деякими іншими мовами програмування PHP може бути менш ефективним щодо швидкодії. Він виконується на рівні сервера, що може призводити до затримок при обробці великих навантажень;

2) безпека: Невірні конфігурації або недбалість можуть призвести до потенційних проблем з безпекою. PHP має історію деяких вразливостей, тому необхідно використовувати належні практики безпеки та оновлювати мову та її компоненти;

3) якість коду: Завдяки простоті вивчення PHP, існує широке коло якісних рішень. Це може призводити до погано написаного і підтримуваного коду, якщо розробники не дотримуються найкращих практик програмування.

PHP-скрипти можна розміщувати в будь-якому місці документа, вони завжди починаються з `<?php` і закінчуються `?>`. Крім того, оператори PHP закінчуються крапкою з комою (;). На рис. 2.23 наведено простий скрипт, який використовує вбудовану функцію *echo* для виведення на сторінку тексту «The Best PHP Examples».

```
<!DOCTYPE html>
<html>
<body>
<h1>Social Protection</h1>
<?php echo "Ласкаво просимо до соціального захисту"; ?>
</body>
</html>
```

Рисунок 2.23 – Базовий синтаксис

Використання умовного оператора та змінних на рис. 2.24 використовується умовний оператор *if* для перевірки значення змінної *\$age*. Якщо значення *\$age* більше або дорівнює 18, виводиться повідомлення «Ви повнолітній». В протилежному випадку виводиться повідомлення «Ви неповнолітній»

```
<?php
    $age = 25;

    if ($age >= 18) {
        echo "Ви повнолітній.";
    } else {
        echo "Ви неповнолітній.";
    }
?>
```

Рисунок 2.24 – Використання умовного оператора та змінних

Робота з масивами: на рис. 2.25 наведено створення масиву користувачів зі збереженням інформації про кожного користувача. У цьому прикладі створюється масив *\$users*, який містить інформацію про різних користувачів соціального захисту. Кожен користувач представлений окремим масивом з ключами, такими як *'name'*, *'age'*, *'occupation'* та *'income'*. Можливо додавати більше користувачів до масиву, додаючи їх як окремі масиви всередині *\$users*.

```
$users = array(  
    array(  
        'name' => 'John',  
        'age' => 30,  
        'occupation' => 'Engineer',  
        'income' => 5000  
    ),  
    array(  
        'name' => 'Mary',  
        'age' => 25,  
        'occupation' => 'Teacher',  
        'income' => 4000  
    ),  
    array(  
        'name' => 'Peter',  
        'age' => 35,  
        'occupation' => 'Doctor',  
        'income' => 7000  
    )  
);
```

Рисунок 2.25 – Робота з масивами

Робота з формою та обробка даних наведена на рис. 2.26. Спочатку перевіряється, чи була надіслана форма за допомогою методу `$_SERVER["REQUEST_METHOD"]`. Якщо форма була надіслана методом POST, отримується значення полів форми за допомогою `$_POST`, а потім виводиться ці значення на екран (рис.2.26).

```
<?php
// Перевірка, чи була надіслана форма
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Отримання значень полів форми
    $name = $_POST["name"];
    $email = $_POST["email"];

    // Виведення отриманих даних
    echo "Ім'я: " . $name . "<br>";
    echo "Email: " . $email;
}
?>
```

Рисунок 2.26 – Робота з формою та обробка даних

На рис. 2.27 показується робота з базою даних. На рис. 2.27 видно, як підключитися до бази даних MySQL за допомогою функції *mysqli_connect*. Потім виконується запит до бази даних за допомогою *mysqli_query* і отримується результати запиту. За допомогою *mysqli_fetch_assoc* буде отримано кожний рядок результатів і виведено його на екран. На кінці необхідно закрити з'єднання з базою даних за допомогою *mysqli_close*.

```
<?php
// Підключення до бази даних
$conn = mysqli_connect("localhost", "username", "password", "database");

// Виконання запиту
$result = mysqli_query($conn, "SELECT * FROM users");

// Виведення результатів запиту
while ($row = mysqli_fetch_assoc($result)) {
    echo "Ім'я: " . $row["name"] . "<br>";
    echo "Email: " . $row["email"] . "<br>";
}

// Закриття з'єднання з базою даних
mysqli_close($conn);
?>
```

Рисунок 2.27 – Робота з базою даних

2.5 Бази даних

2.5.1 MySQL

MySQL – це безкоштовна система управління реляційними базами даних, розроблена TCX для пришвидшення обробки великих баз даних. Ця система керування базами даних (СКБД) із відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку був дуже схожий на mSQL, але з часом він зріс і зараз є однією з найпопулярніших систем управління базами даних. В основному він використовується для створення динамічних вебсторінок завдяки чудовій підтримці з різних мов програмування [19].

MySQL вважається придатним рішенням для малих та середніх застосунків. Код необробленого сервера компілюється на багатьох платформах. Функціональність сервера найбільш очевидна в системах UNIX з багато потоковою підтримкою, що покращує загальну продуктивність системи.

MySQL пропонує широкий спектр можливостей, які роблять його привабливим для різних сфер застосування:

- 1) він пропонує всебічну підтримку безлічі мов програмування і пов'язаних з ними середовищ, що робить інтеграцію дуже простою;
- 2) підтримується необмежена кількість користувачів, що одночасно працюють із базою даних;
- 3) його підтримують усі популярні операційні системи та дистрибутиви, включно з Linux, Windows, macOS і Debian;
- 4) він має дуже низькі або нульові експлуатаційні витрати і може обробляти більше даних. Завдяки цій економічній доцільності MySQL підходить для малого бізнесу та індивідуальних розробників;
- 5) тріади CIA підтримуються MySQL, оскільки вона використовує рівень SSL для шифрування збережених даних.

MySQL пропонує широкий спектр можливостей, які роблять його привабливим для різних сфер застосування: Той факт, що багато технологічних

гігантів покладаються на неї, ще більше зміцнює їх природне становище.

Перевагами є:

- відкритий код та активна спільнота: MySQL є вільним програмним забезпеченням з відкритим кодом, що дає користувачам свободу вносити зміни та розповсюджувати його безкоштовно. Велика спільнота розробників підтримує MySQL, надаючи підтримку та оновлення;

- простота використання: MySQL має зрозумілий та легкий у використанні інтерфейс, що дозволяє швидко створювати та керувати базами даних. Велика кількість документації та ресурсів для навчання сприяють швидкому оволодінню цією СУБД;

- надійність та стабільність: MySQL відомий своєю стабільністю та надійністю. Він може працювати без збоїв протягом тривалого часу і забезпечує механізми відновлення в разі аварій;

- широкі можливості розширення: MySQL підтримує велику кількість розширень, які дозволяють розробникам розширювати функціональність бази даних відповідно до своїх потреб. Застосування різноманітних розширень, таких як реплікація, кластеризація та повнотекстовий пошук, дозволяє підвищити продуктивність та ефективність роботи з даними.

Незважаючи на свої переваги, MySQL має також кілька недоліків, які варто враховувати:

- обмежена підтримка високої навантаженості: У порівнянні з деякими іншими СКБД, MySQL може мати обмежену масштабованість для дуже великих навантажень та великих обсягів даних. Для таких сценаріїв можуть знадобитись додаткові налаштування та оптимізації, аби забезпечити потрібну продуктивність;

- обмежені функціональні можливості: У порівнянні з деякими енттерпрайз-рівневими СКБД, MySQL може мати обмежені функціональні можливості. Наприклад, підтримка аналітики даних, складних операцій згортання та з'єднання даних може бути обмеженою;

– обмежена підтримка скалірування: Хоча MySQL підтримує горизонтальне скалірування, воно може вимагати додаткових налаштувань та конфігурацій. Вибір правильних стратегій розподілу даних та налаштування кластерів може бути викликом для розробників.

Незважаючи на недоліки, MySQL залишається однією з найпопулярніших та надійних СКБД, яка використовується в різних сферах та має широку спільноту розробників, яка активно підтримує і вдосконалює цю платформу.

2.5.2 PhpMyAdmin

При розробці вебресурсів або перенесення сайтів потрібно отримати доступ до БД – бази даних проєктів на сервері. Найпростіший метод – командний рядок. Інший варіант – графічний інтерфейс, що допомагає уявити дані у вигляді ієрархічних списків. PhpMyAdmin – популярний інструмент для роботи з БД, має відкритий код, написаний на мові PHP. Працює через браузер, може бути інтегрований в інтерфейс програмного забезпечення або клієнтську панель управління хостингом.

PhpMyAdmin – це вебпрограма з відкритим кодом, написана на PHP, з графічним вебінтерфейсом для управління базами даних MySQL або MariaDB. Ви можете використовувати phpMyAdmin для управління сервером MySQL, виконання SQL-запитів та перегляду та редагування вмісту таблиць баз даних через ваш браузер. Ця програма користується великою популярністю у веброзробників, оскільки дозволяє керувати базою даних MySQL з будь-якого комп'ютера, підключеного до Інтернету, без необхідності вводити команди SQL через зручний інтерфейс та без встановлення додаткового програмного забезпечення.

PhpMyAdmin сьогодні дійсно широко використовується. Останнє пов'язано з тим, що розробники інтенсивно розробляють свої продукти, враховуючи всі нововведення в СКБД MySQL. Переважна більшість українських постачальників використовують цей застосунок як панель управління, щоб надати своїм клієнтам можливість керувати призначеними ними базами даних.

Висновки до розділу 2

В другому розділі було описано технології та методи, що використовувались для розробки вебзастосунку соціального захисту:

- 1) Java Script;
- 2) Html;
- 3) Css;
- 4) Php;
- 5) MySQL;

Кожну технологію було детально описано, наведено переваги у використанні. Вищеперераховані мови та технології – це невід’ємна частина розроблюваного вебзастосунку онлайн-доступу до соціальних послуг для громадян.

3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ВЕБЗАСТОСУНКУ ДОСТУПУ ДО СОЦІАЛЬНИХ ПОСЛУГ

3.1 Сценарії використання

Сценарії використання – це детальний опис послідовності подій, які можуть відбуватися між користувачами та системою. Вони визначають, як користувачі взаємодіють з системою, які дії вони виконують та які результати вони очікують. Сценарії використання допомагають розкрити потенційні можливості системи, проявити її потенційну цінність та показати, як система може бути використана для вирішення реальних проблем [20].

Сценарії використання мають кілька цілей:

- розуміння функціональних потреб користувачів: Сценарії допомагають ідентифікувати, які дії виконують користувачі та як вони взаємодіють з системою. Це дозволяє розробникам отримати глибоке розуміння того, як користувачі сприймають та використовують систему;
- визначення функціональних вимог: Сценарії допомагають виявити потреби користувачів та встановити функціональні вимоги для системи. Вони дозволяють визначити, які функції та можливості повинна мати система, щоб задовольнити потреби користувачів;
- оцінка взаємодії: Сценарії дозволяють оцінити, наскільки ефективно система задовольняє потреби користувачів. Вони допомагають ідентифікувати можливі проблеми взаємодії та виявити області для поліпшення.

Таблиця 3.1 – Опис прецеденту «Анкетування»

Use case section	Comment
Usecase Name	Анкетування
Scope	Інформаційний вебзастосунок соціального захисту населення Миколаївської територіальної громади
Level	Успішно пройти анкетування
Primary Actor	Користувач
Stakeholders and interests	Користувач – надання інформації для аналізу потреб населення та покращення соціальних програм
Preconditions	Користувач перейшов на сторінку анкетування
Success guarantee	Користувач потрібен використовувати браузер з підтримкою HTML 5, CSS 3
Main Success Scenario	Користувач проходить анкетування
Extensions	<ol style="list-style-type: none"> 1. Користувач залишив порожніми основні поля : 2. Користувач перейшов до сторінки «Анкетування» 3. Користувач залишив порожнім обов'язкове поле 4. Система видасть повідомлення про порожнє поле
Special Requirements	Швидкість мережі потрібна бути більшою ніж 40 кбайт/с

Таблиця 3.2 – Опис прецеденту «Реєстрація на зустріч»

Use case section	Comment
Usecase Name	Реєстрація на зустріч
Scope	Інформаційний вебзастосунок соціального захисту населення Миколаївської територіальної громади
Level	Успішно зареєструватися на зустріч із працівниками соціального захисту
Primary Actor	Користувач
Stakeholders and interests	Користувач успішно зареєструвався на зустріч
Preconditions	Користувач перейшов на сторінку «Реєстрація на зустріч»
Success guarantee	Користувач має використовувати браузер з підтримкою HTML 5, CSS 3
Main Success Scenario	Користувач обирає, з ким та на яку тему хоче назначити зустріч
Extensions	<ol style="list-style-type: none"> 1. Користувач відправив некоректний запит; 2. Користувач відправляє некоректний запит ; 3. Користувач натиснув кнопку Відправити. Система видала повідомлення про помилку
Special Requirements	Швидкість мережі має бути більшою ніж 40 кбайт/с

Таблиця 3.3 – Опис прецеденту «Реєстрація на гуманітарну допомогу»

Use case section	Comment
Usecase Name	Реєстрація на гуманітарну допомогу
Scope	Інформаційний вебзастосунок соціального захисту населення Миколаївської територіальної громади
Level	Успішно зареєструватися на отримання гуманітарної допомоги
Primary Actor	Користувач
Stakeholders and interests	Користувач успішно зареєструвався на отримання допомоги
Preconditions	Користувач перейшов на сторінку «Реєстрація на гуманітарну допомогу»
Success guarantee	Користувач потрібен використовувати браузер з підтримкою HTML 5, CSS 3
Main Success Scenario	Користувач вказує свій статус і реєструється на допомогу
Extensions	<ol style="list-style-type: none"> 1. Користувач відправив некоректний запит; 2. Користувач відправляє некоректний запит Користувач натиснув кнопку «Відправити»; 3. Система видала повідомлення про помилку.
Special Requirements	Швидкість мережі потрібна бути більшою ніж 40 кбайт/с

Сценарії використання описують, як користувачі будуть виконувати завдання на вебсайті і як система буде реагувати на їх запити. Вони включають послідовність кроків, що ведуть до досягнення мети користувача. Сценарії

використання мають важливе значення, оскільки допомагають уточнити поведінку системи і виявити можливі проблеми. Вони також визначають цілі, які допомагають оцінити вартість і складність системи. Система представляється як набір акторів, які взаємодіють з нею через прецеденти. Актори можуть бути будь-якими зовнішніми сутностями, які взаємодіють з системою. Таким чином, кожен сценарій визначає дії, які виконує сама система. На рис. 3.1 представлена діаграма використання програмного забезпечення.

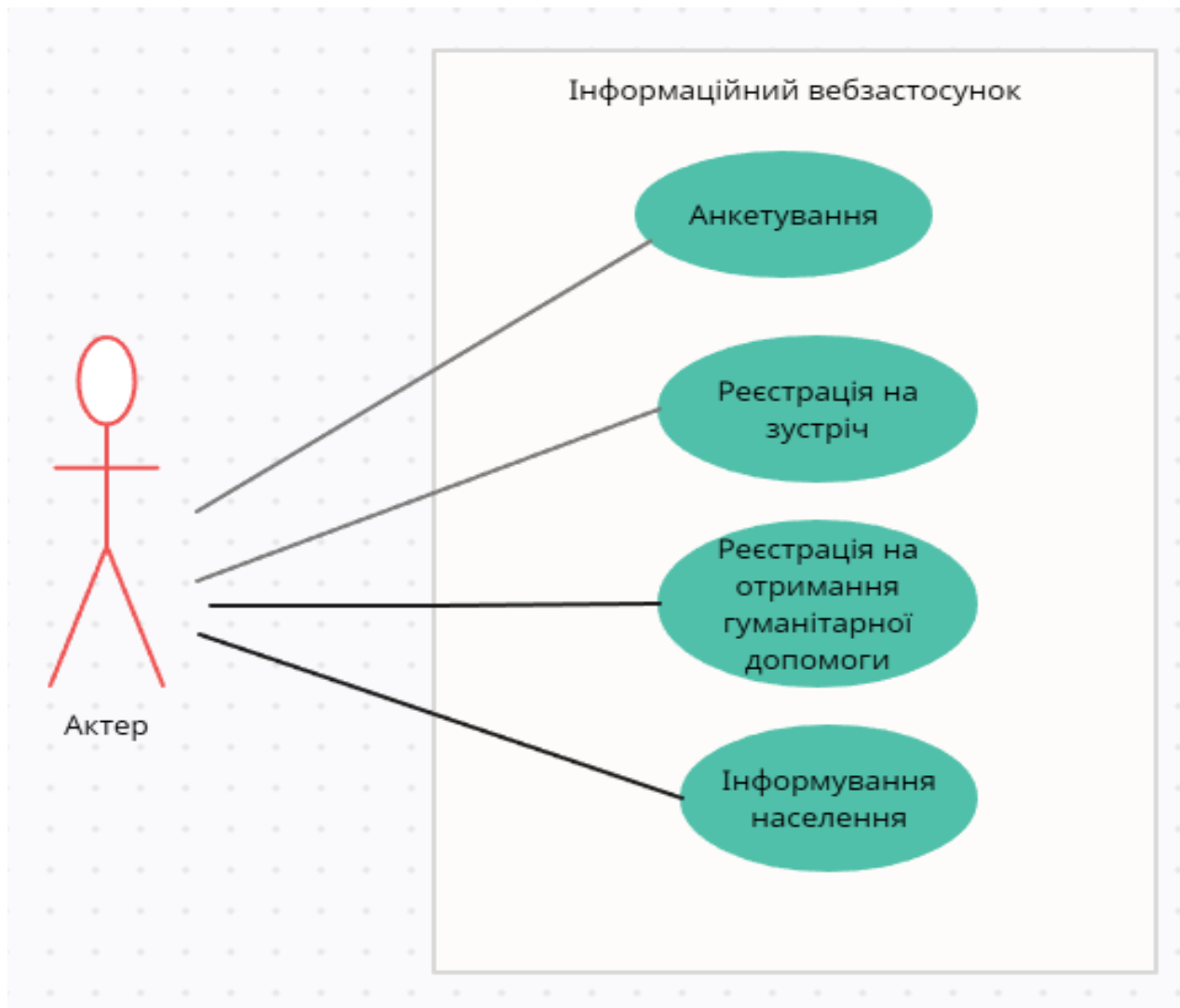


Рисунок 3.1 – Діаграма використання

3.2 Макет користувацького інтерфейсу

При розробці інтерфейсу системи було розроблено макети. Макет інтерфейсу – це візуальне представлення користувацького інтерфейсу (UI) застосунку, вебсайту або програмного продукту. Він служить моделлю, яка

демонструє, як буде виглядати та функціонувати інтерфейс після його реалізації.

Головна мета створення макету інтерфейсу полягає в тому, щоб наглядно продемонструвати, як користувачі будуть взаємодіяти з програмою або вебсайтом. Він допомагає команді проєкту, замовнику та іншим зацікавленим сторонам краще зрозуміти, як інтерфейс буде працювати на різних етапах розробки.

Другою важливою метою при створенні макету інтерфейсу є визначення оптимальної структури та навігації. Він встановлює, як різні компоненти та екрани взаємодіють між собою, як користувачі будуть переходити від одного елемента до іншого та як зручно буде організована інформація.

Третя мета полягає у валідації концепції. Макет дозволяє перевірити, чи відповідає запропонована концепція інтерфейсу потребам та очікуванням користувачів. За допомогою макету можна збирати відгуки та повертатись до попередніх етапів розробки, якщо потрібно внести зміни або виправити проблеми. Ще одним важливим аспектом є скорочення часу та витрат. Розробка макету перед програмуванням дозволяє уникнути зайвих витрат часу і ресурсів. Виправлення проблем та внесення змін на етапі макету виявляється набагато простіше та дешевше, ніж після реалізації та програмування функціоналу.

Не менш важливим є те, що макет інтерфейсу допомагає вирішувати дизайнерські питання. Він дає змогу експериментувати з різними дизайнерськими рішеннями, включаючи колірну палітру, типографіку та використання графіки. Таким чином, макет інтерфейсу допомагає створити зовнішній вигляд інтерфейсу, який буде привабливим та зручним для користувачів.

Ключове місце при створенні макету, займає головна сторінка інформаційного вебзастосунку. На рис. 3.2 представлено макет головної сторінки.

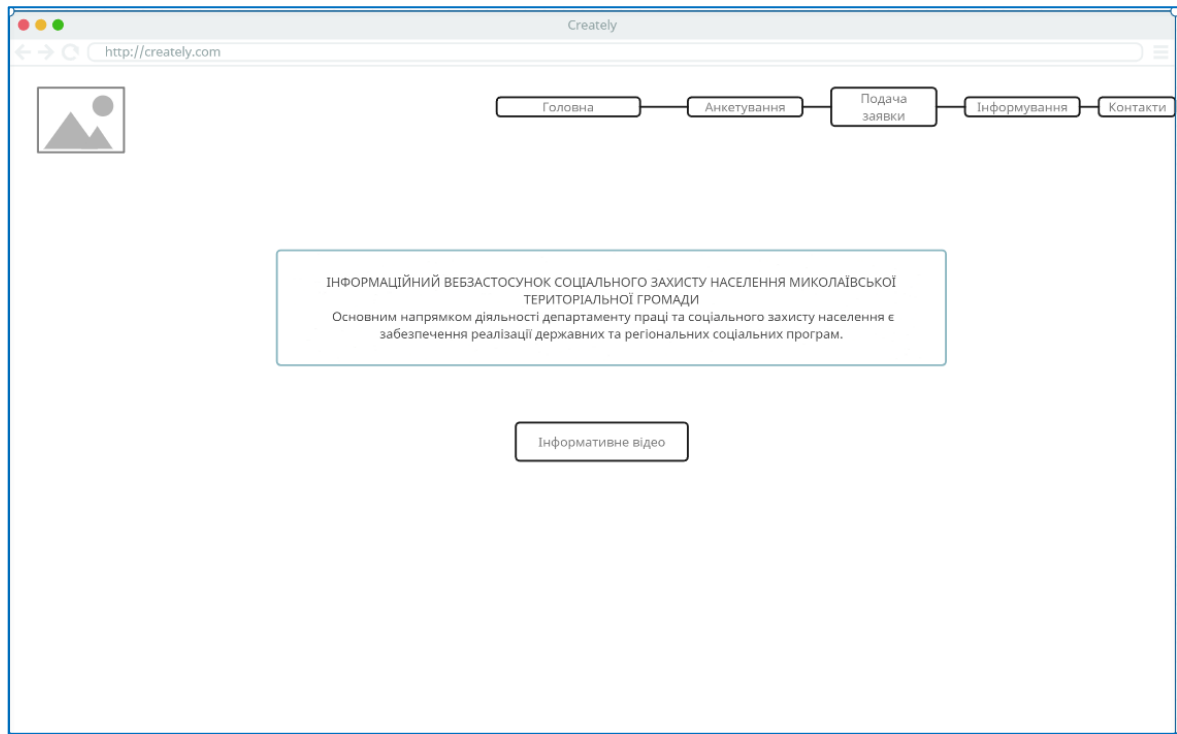


Рисунок 3.2 – Макет головної сторінки

Сторінка «Анкетування» створена для збору інформації про потреби та проблеми населення. Працівники соціального захисту можуть створювати анкети та опитувальники, які допомагають збирати важливі дані та розуміти потреби людей. Інформація, зібрана через анкетування, допомагає удосконалювати соціальний захист та пристосовувати програми та послуги до потреб населення. На рис. 3.3 представлено макет однієї із сторінок «Анкетування».

The image shows a browser window with the URL 'http://creately.com'. The page title is 'Анкетування'. The form contains the following elements:

- A text input field labeled 'Ваше ПІБ'.
- A text input field labeled 'Район'.
- A section titled 'Питання 1' containing four radio button options, each labeled 'Варіанти відповідей'.
- A button labeled 'Відправити' at the bottom.

Рисунок 3.3 – Макет сторінки «Анкетування»

Сторінка «Подача заявки» надає можливість користувачам зареєструватися для отримання особистої зустрічі зі співробітниками соціального захисту. Вона може містити форми, в яких користувачі можуть внести свої особисті дані, вибрати зручну дату та час для зустрічі, а також вказати, яку саме допомогу вони потребують. Це спрощує процес організації зустрічей та дозволяє соціальним працівникам краще планувати свою роботу. Також зазначений розділ надає можливість користувачам зареєструватися на отримання гуманітарної допомоги. Крім заповнення особистих даних та вибору дати та часу зустрічі зі співробітниками соціального захисту, користувачі можуть заповнити анкету на отримання гуманітарної допомоги. Це дозволяє ефективніше виявляти людей, які потребують невідкладної допомоги та прискорює процес надання гуманітарної допомоги. На рис 3.4 представлено макет сторінки подачі заявки.

Реєстрація на зустріч

Ім'я

Призвище

Номер телефону

Оберіть організацію

Item 1
Item 2
Item 3
Item 4
Item 5

Статус

Відправити

Рисунок 3.4 – Макет сторінки подачі заявки

Сторінка «Інформування» надає інформацію та послуги, пов'язані з соціальним захистом населення. Основна мета такої сторінки полягає в тому, щоб надати громадянам доступ до інформації про соціальні програми, пільги, послуги та права, які надаються на території Миколаївської територіальної громади. На рис. 3.5 представлено макет сторінки інформування.

НОВИНИ ТА ОГЛОШЕННЯ

Новина

Новина

Новина

Новина

1 2 3 4

Рисунок 3.5 – Макет сторінки інформування

Сторінка «Контакти» містить інформацію про співробітників соціального захисту. Тут можуть бути вказані номери телефонів, місце знаходження, електронні адреси або інші засоби зв'язку. Цей розділ допомагає користувачам знайти необхідну контактну інформацію та звернутися за допомогою чи консультацію. На рис. 3.6 представлено макет сторінки «Контакти».

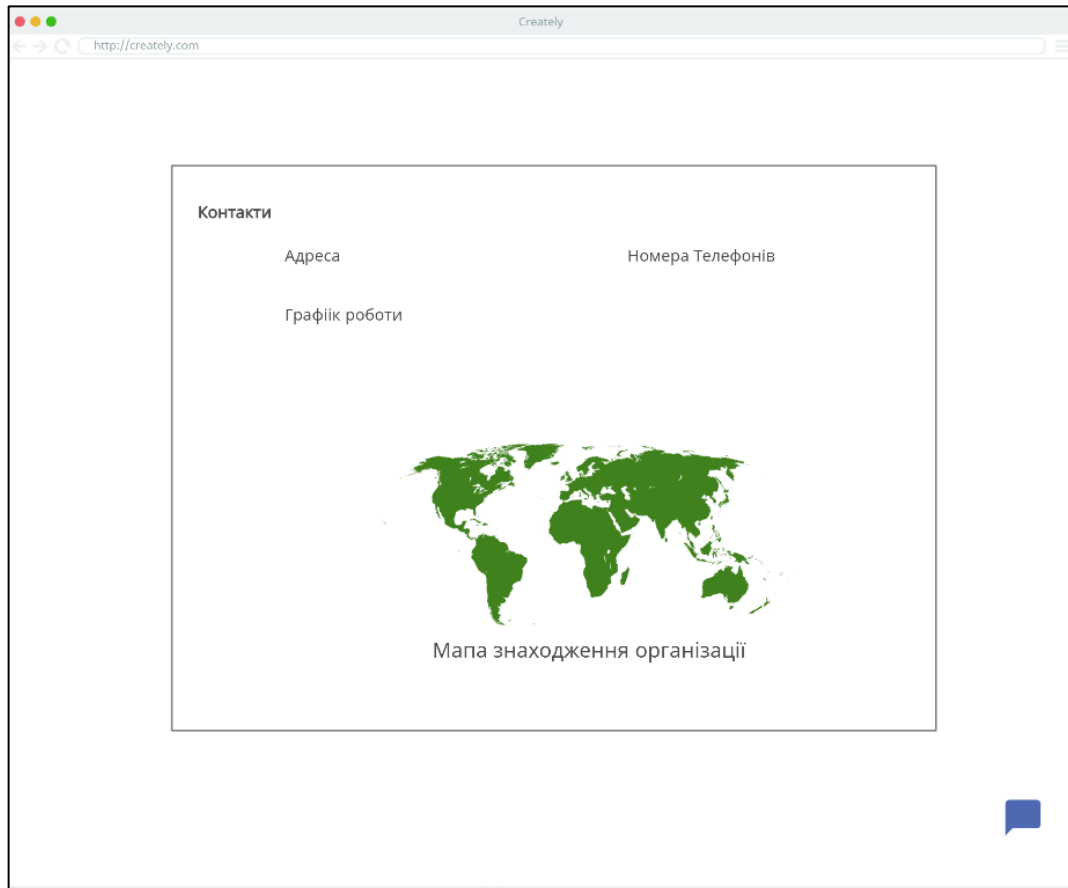


Рисунок 3.6 – Макет сторінки контакти

Висновки до розділу 3

В третьому розділі під час проведеного аналізу було виявлено основні функції вебзастосунку та сценарії їх використання. Цей аналіз спрямований на розуміння потреб користувачів і визначення ключових можливостей, які вебзастосунок повинен надавати для задоволення цих потреб.

Також було розроблено макети, у яких відображаються розташування елементів на сторінках вебзастосунку.

4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ВЕБЗАСТОСУНКУ

4.1 Опис інтерфейсу програми

Після успішного завершення процесу створення сценаріїв використання та розроблення користувацьких макетів, наш вебзастосунок готовий до повноцінного використання. При першому відвідуванні нашого вебзастосунку, користувач отримує доступ до головної сторінки (рис. 4.1). Вебзастосунок був розроблений з орієнтацією на інтуїтивне сприйняття користувачем. На вебсайті він знайде повний набір необхідних функцій.

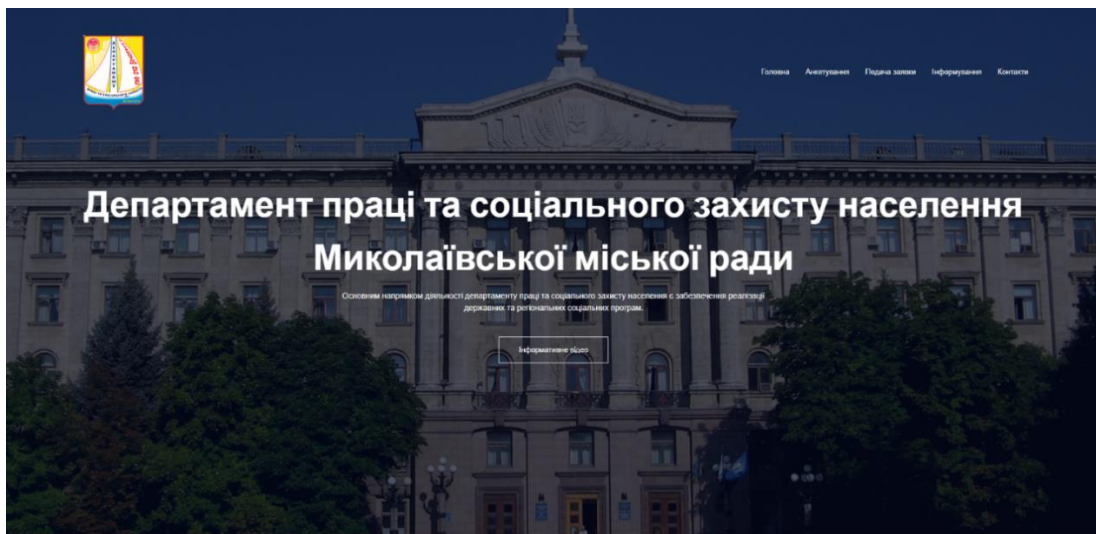


Рисунок 4.1 – Головна сторінка вебзастосунку

Після натиснення на кнопку «Інформативне відео» перед користувачем висвітиться відео з короткою інформацією про соціальний захист (рис. 4.2).

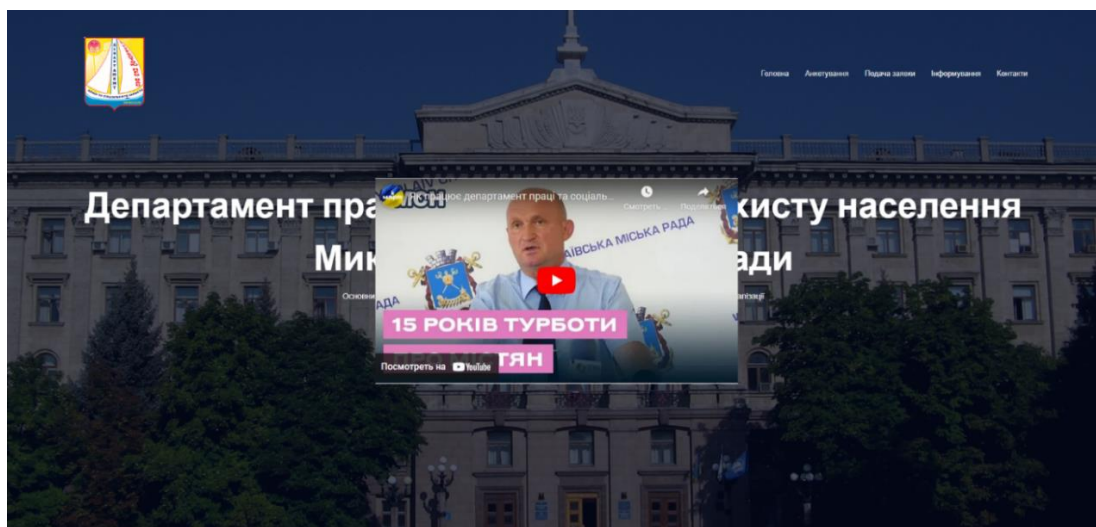


Рисунок 4.2 – Інформативне відео

Наступним кроком користувач може вибрати для себе пункт представлений у меню; при наведенні курсору на пункт «Анкетування» перед ним буде представлений вибір з переліку анкет (рис. 4.3).

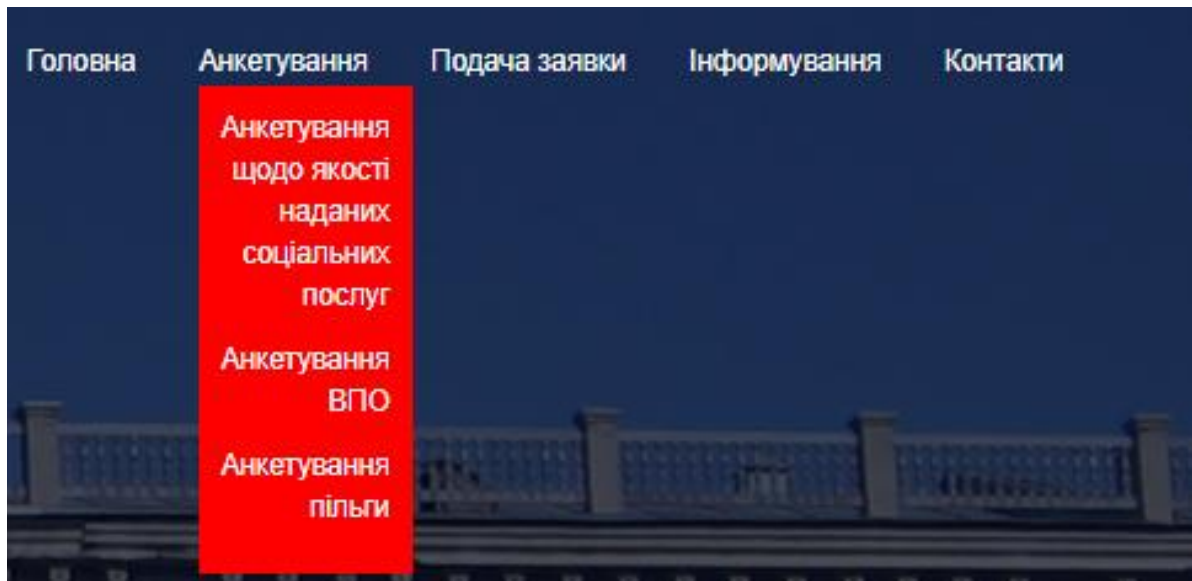


Рисунок 4.3 – Перелік анкет

Після вибору користувачем пункту «Анкетування ВПО» здійснюється перехід на нову сторінку проходження анкети. На рис. 4.4 представлена відповідна сторінка.

Опитувальник потреб внутрішньо переміщених осіб

ПІБ(повністю):	Скопенко Іван Вікторович	
Телефон:	0730660535	
Потреби внутрішньо переміщених осіб	Важливо	Не важливо
Індивідуальне тимчасове житло	<input checked="" type="radio"/>	<input type="radio"/>
Індивідуальне постійне житло	<input type="radio"/>	<input checked="" type="radio"/>
Медичні послуги	<input checked="" type="radio"/>	<input type="radio"/>
Фінансова підтримка	<input checked="" type="radio"/>	<input type="radio"/>
Продукти харчування	<input checked="" type="radio"/>	<input type="radio"/>
Робота за фактом	<input checked="" type="radio"/>	<input type="radio"/>
Перекваліфікація (отримання іншої спеціальності)	<input type="radio"/>	<input checked="" type="radio"/>
Навчання дітей	<input checked="" type="radio"/>	<input type="radio"/>
Соціальна підтримка	<input checked="" type="radio"/>	<input type="radio"/>
Інформаційна підтримка	<input type="radio"/>	<input checked="" type="radio"/>
Пошук роботи	<input checked="" type="radio"/>	<input type="radio"/>
Доступ до інтернету	<input checked="" type="radio"/>	<input type="radio"/>

Відправити

Рисунок 4.4 – Проходження анкетування

На рис. 4.5 показано, як при наведенні курсору на пункт «Подача заявок», користувач зможе вибрати із представленого перед ним переліку пункти «Реєстрація на зустріч» або «Реєстрація на гуманітарну допомогу».

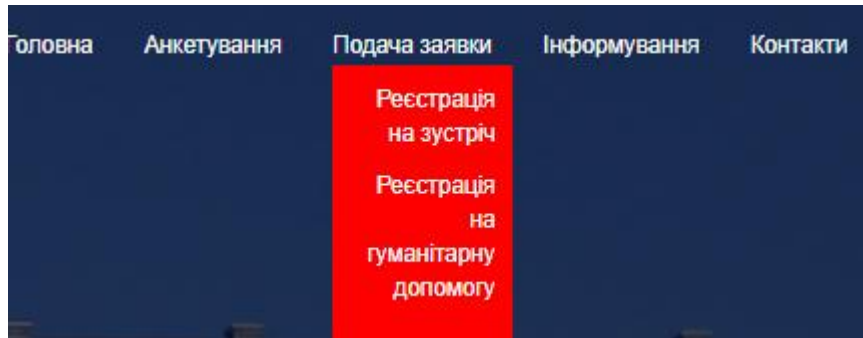


Рисунок 4.5 – Подача заявок

При виборі пункту «Реєстрація на зустріч» користувач переходить на нову сторінку; при заповненні усіх вірних даних, вони будуть зберігатися у базі даних та користувачу повідомиться про успішну відправку форми. На рис. 4.6 представлена реєстрація на зустріч до організації.

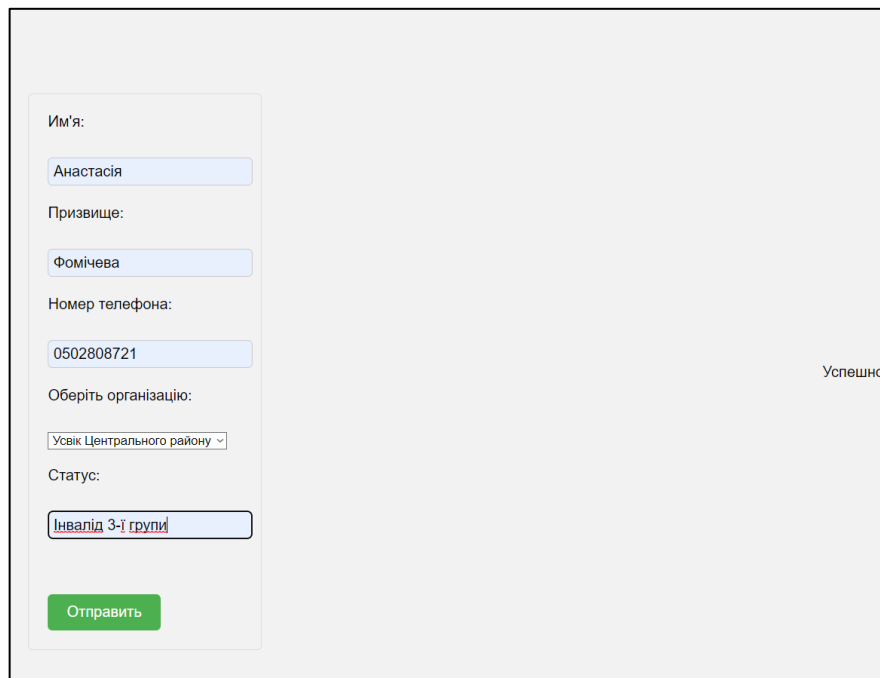
A screenshot of a registration form. The form contains the following fields: 'Имя:' with the value 'Анастасія', 'Призвище:' with the value 'Фомічева', 'Номер телефона:' with the value '0502808721', 'Оберіть організацію:' with a dropdown menu showing 'Усвік Центрального району', and 'Статус:' with the value 'Інвалід 3-ї групи'. A green button labeled 'Отправить' is at the bottom. The word 'Успешно' is visible on the right side of the form.

Рисунок 4.6 – Реєстрація на зустріч

Програмна реалізація форми «Реєстрації на зустріч» наведена на рис. 4.7.

```
<form method="get">
  <label for="name">Им'я:</label><br>
  <input type="text" id="name" name="name"><br>

  <label for="surname">Призвище:</label><br>
  <input type="text" id="surname" name="surname"><br>

  <label for="phone">Номер телефона:</label><br>
  <input type="tel" id="phone" name="phone"><br>

  <label for="organization">Оберіть організацію:</label><br>
  <select id="organization" name="organization">
    <option value="Департамент Праці">Департамент Праці</o
    <option value="Усвік Центр">Усвік Центрального району<
    <option value="Усвік Завод">Усвік Заводського району</
    <option value="Усвік Інгул">Усвік Інгульського району<
    <option value="Усвік Корабл">Усвік Корабельного району<
    <option value="Терцентр">Територіальний Центр</option>
  </select><br><br>
  <label for="status">Статус:</label><br>
  <input type="text" id="status" name="status"><br><br>

  <input type="submit" name="formSubmit">
</form>
```

Рисунок 4.7 – Реалізація форми «Реєстрації на зустріч»

На рис. 4.8–4.9 наведено програмну реалізацію підключення бази даних та заповнену базу даних.

```
<?php
if (isset($_GET['formSubmit'])){
    $nameform = $_GET['name'];
    $surnameform = $_GET['surname'];
    $phoneform = $_GET['phone'];
    $organizationform = $_GET['organization'];
    $statusform = $_GET['status'];
    $mysqli = new mysqli("localhost", "root", "", "pie-18");
    if ($mysqli->connect_errno) {
        echo "Извините, возникла ошибка на сайте";
        exit;
    }
    $name = $mysqli->real_escape_string($nameform);
    $surname = $mysqli->real_escape_string($surnameform);
    $phone = $mysqli->real_escape_string($phoneform);
    $organization = $mysqli->real_escape_string($organizationform);
    $status = $mysqli->real_escape_string($statusform);
    $query = "INSERT INTO users (name,surname,phone,organizati
    if ($mysqli->query($query) === false) {
        echo "Ошибка: " . $mysqli->error;
        exit;
    } else {
        echo "Успішно" . '<br>';
    }
    $mysqli->close();
}
?>
```

Рисунок 4.8 – Підключення бази даних

name	surname	phone	organization	status
Vanya	Skopenko	730660535	Департамент Праці	ВПО
Vanya	fomichev	502808721	Усвік Інгул	ВПО
Ірина	Чорна	502808721	Усвік Інгул	ВПО
Vanya	Чорна	730660535	Терцентр	ВПО
Анастасія	Фомічева	502808721	Усвік Центр	Інвалід 3-ї групи
Анастасія	Фомічева	502808721	Усвік Центр	Інвалід 3-ї групи

Рисунок 4.9 – База даних форми

На рис. 4.10 наведено, як користувач також може зареєструватися на отримання гуманітарної допомоги.

Анкета на гуманітарну допомогу

Район:

ПІБ:

Ідентифікаційний код:

Номер телефону:

Статус:
Особа похилого віку (старше 65)

Надіслати

Успішно

Рисунок 4.10 – Реєстрація на отримання гуманітарної допомоги

Програмна реалізація форми реєстрації на отримання гуманітарної допомоги наведена на рис. 4.11.

```

<form method="get">
  <h2>Анкета на гуманітарну допомогу</h2>

  <label for="district">Район:</label>
  <input type="text" id="district" name="district" required><br><br>

  <label for="fullname">ПІБ:</label>
  <input type="text" id="fullname" name="fullname" required><br><br>

  <label for="code">Ідентифікаційний код:</label>
  <input type="text" id="code" name="code" required><br><br>
  <label for="phone">Номер телефону:</label>
  <input type="text" id="phone" name="phone" required><br><br>

  <label for="status">Статус:</label>
  <select id="status" name="status">
    <option value="elderly">Особа похилого віку (старше 65)</option>
    <option value="disability_1">Особа з інвалідністю 1 групи</option>
    <option value="disability_2">Особа з інвалідністю 2 групи</option>
    <option value="disability_3">Особа з інвалідністю 3 групи</option>
    <option value="social_assistance">Отримувач державної соціальної допомоги малозабезпеченим сім'ям</option>
    <option value="single_mother">Одинокa мати</option>
    <option value="parents_disability_child">Батьки, які виховують дитину з інвалідністю</option>
    <option value="guardians">Опікуни</option>
  </select><br><br>

  <input type="submit" name="formSubmit" value="Надіслати">
</form>

```

Рисунок 4.11 – Реалізація форми реєстрації отримання гуманітарної допомоги

```

<?php
if (isset($_GET['formSubmit'])){
  $districtform = $_GET['district'];
  $fullnameform = $_GET['fullname'];
  $codeform = $_GET['code'];
  $phoneform = $_GET['phone'];
  $statusform = $_GET['status'];
  $connection = new mysqli("localhost", "root", "", "bazadanix");
  if ($connection->connect_errno) {
    echo "Вибачте, сталася помилка на сайті";
    exit;
  }
  $district = $connection->real_escape_string($districtform);
  $fullname = $connection->real_escape_string($fullnameform);
  $code = $connection->real_escape_string($codeform);
  $phone = $connection->real_escape_string($phoneform);
  $status = $connection->real_escape_string($statusform);
  $query = "INSERT INTO `poek` (`district`, `fullname`, `code`, `phone`, `status`) VALUES ('$district', '$fullname', '$code', '$phone', '$status')";
  if ($connection->query($query) === false) {
    echo "Помилка: " . $connection->error;
    exit;
  } else {
    echo "Успішно" . '<br>';
  }
  $connection->close();
}
?>

```

Рисунок 4.12 – Підключення бази даних

district	fullname	code	phone	status
Центральний	Скопенко Іван Вікторович	1231243124	730660535	elderly
Центральний	Скопенко Іван Вікторович	1231243124	730660535	elderly
Заводський	Фомічева Анастасія Кирилівна	1231243124	730660535	disability_3

Рисунок 4.13 – База даних форми

Наступним пунктом буде сторінка «Інформування», де користувач зможе отримати свіжу інформацію та послуги, пов'язані з соціальним захистом населення, які надаються на території Миколаївської територіальної громади. На рис. 4.14 зображена сторінка інформування.

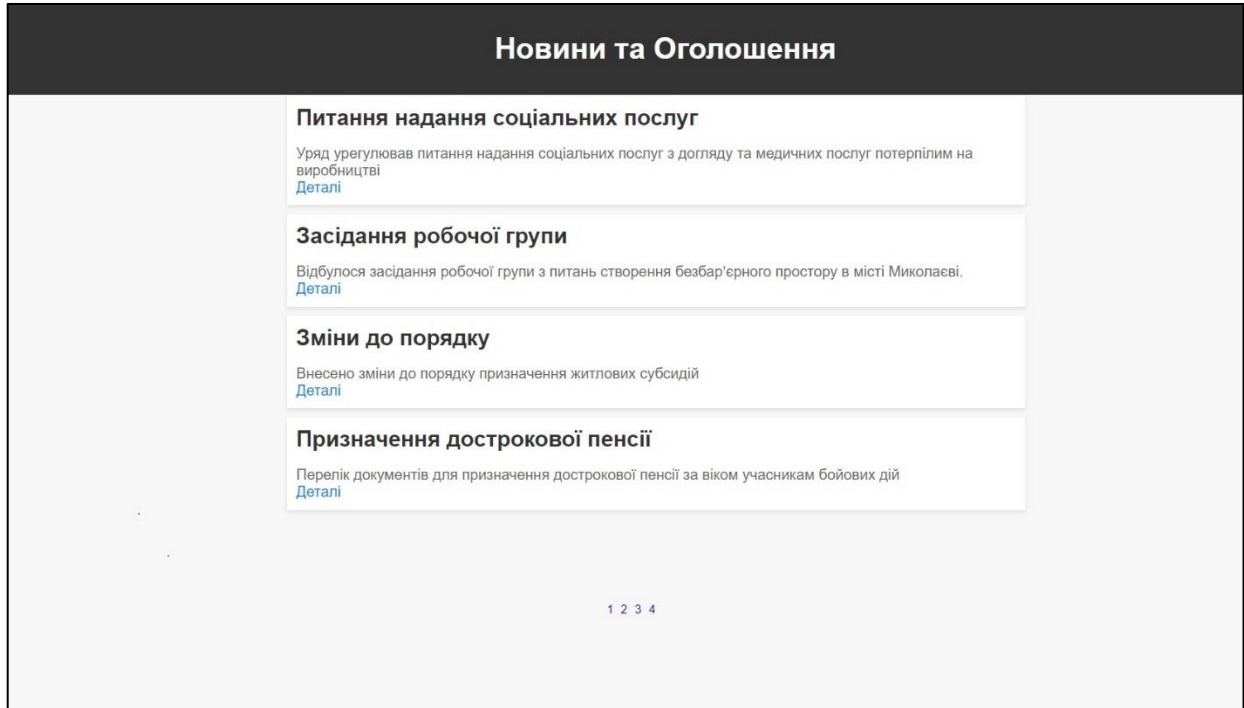


Рисунок 4.14 – Сторінка інформування

Коли користувач натисне кнопку «Деталі», він зразу перейде на сайт та дізнається усю повну інформацію.

Останнім пунктом буде «Контакти», у якому є вибір із представлених організацій, де користувач зможе обрати необхідний та дізнатися про номери телефонів, місцезнаходження, електронні адреси або інші засоби зв'язку (рис. 4.15).

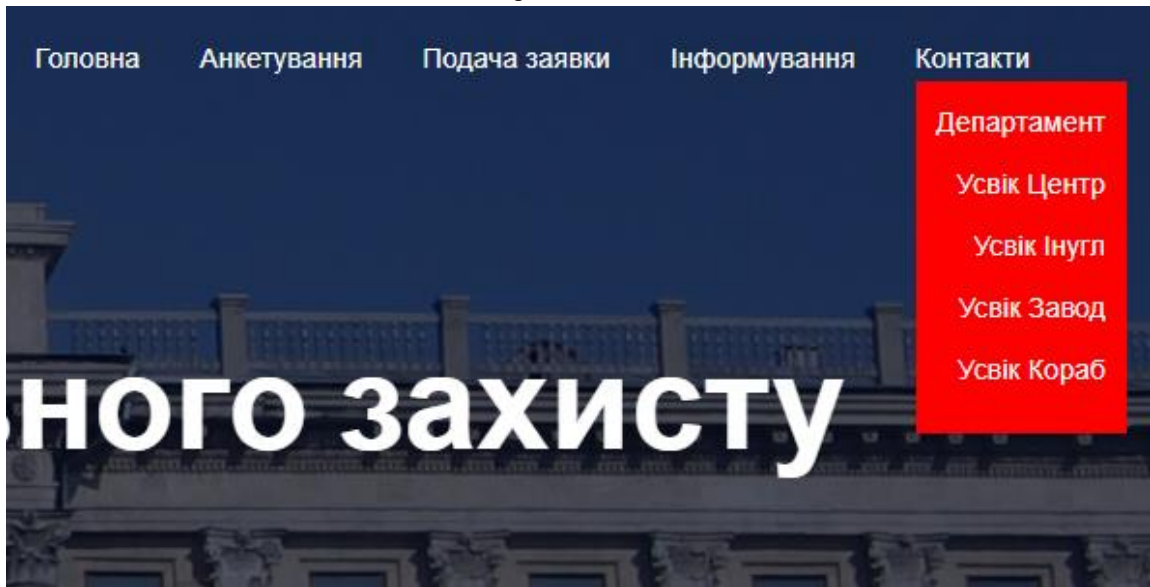


Рисунок 4.15 – Перелік контактів

На рис 4.16–4.20 представлені усі контактні форми організації.

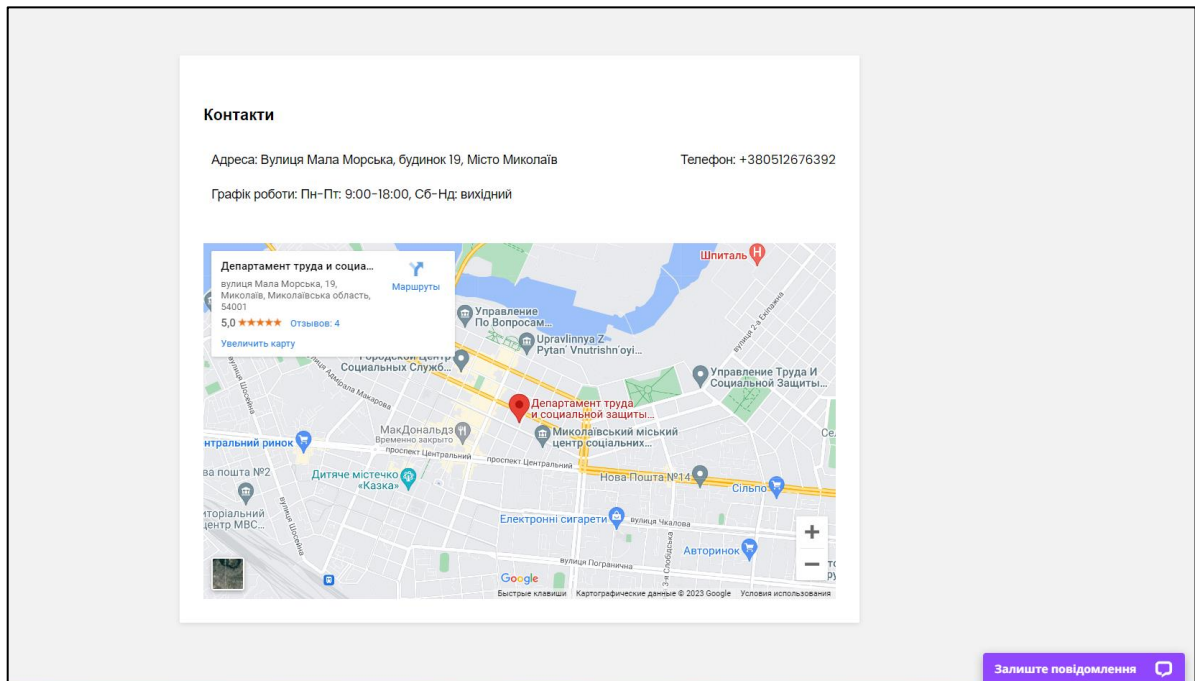


Рисунок 4.16 – Контакти Департаменту

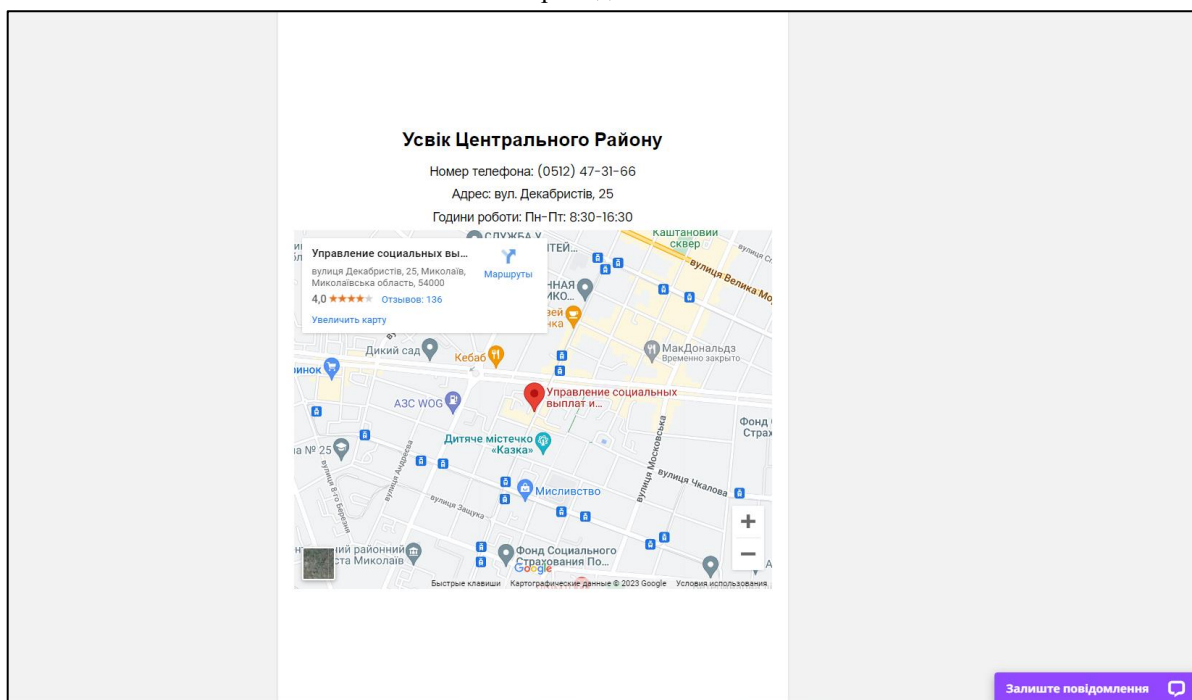


Рисунок 4.17 – Контакти УСВІК Центрального району

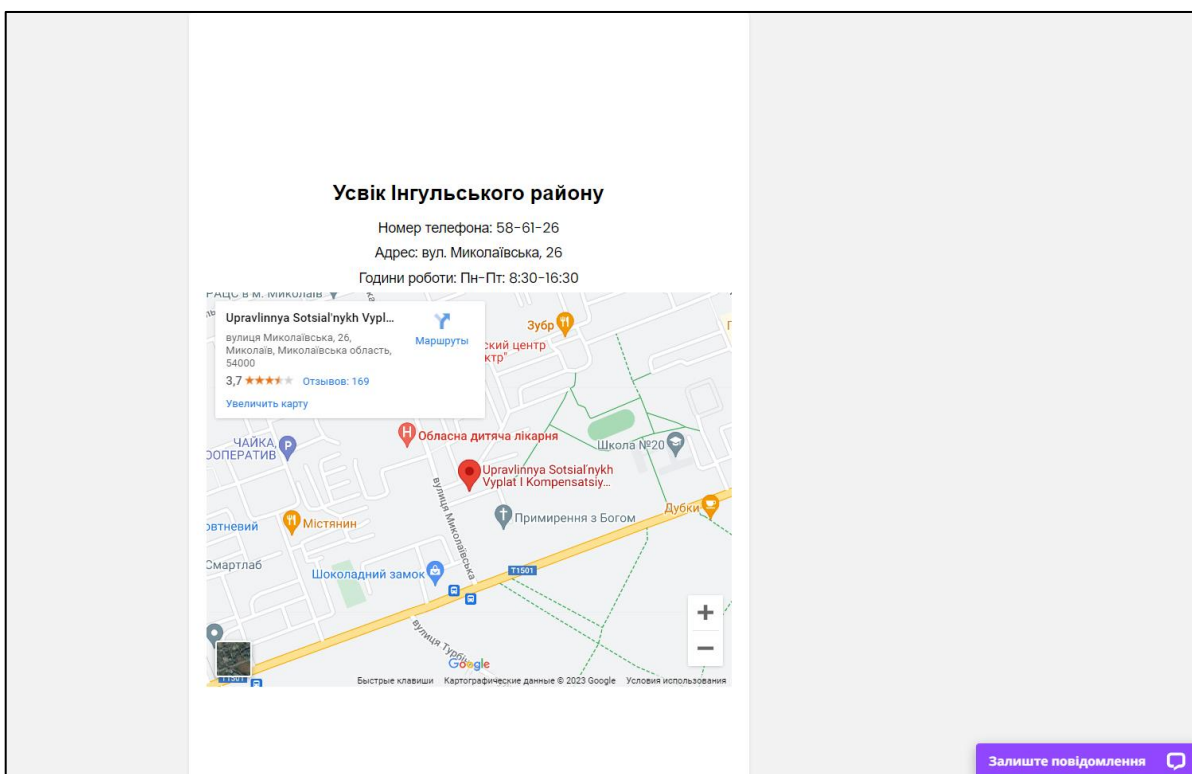


Рисунок 4.18 – Контакти УСВІК Інгульського району

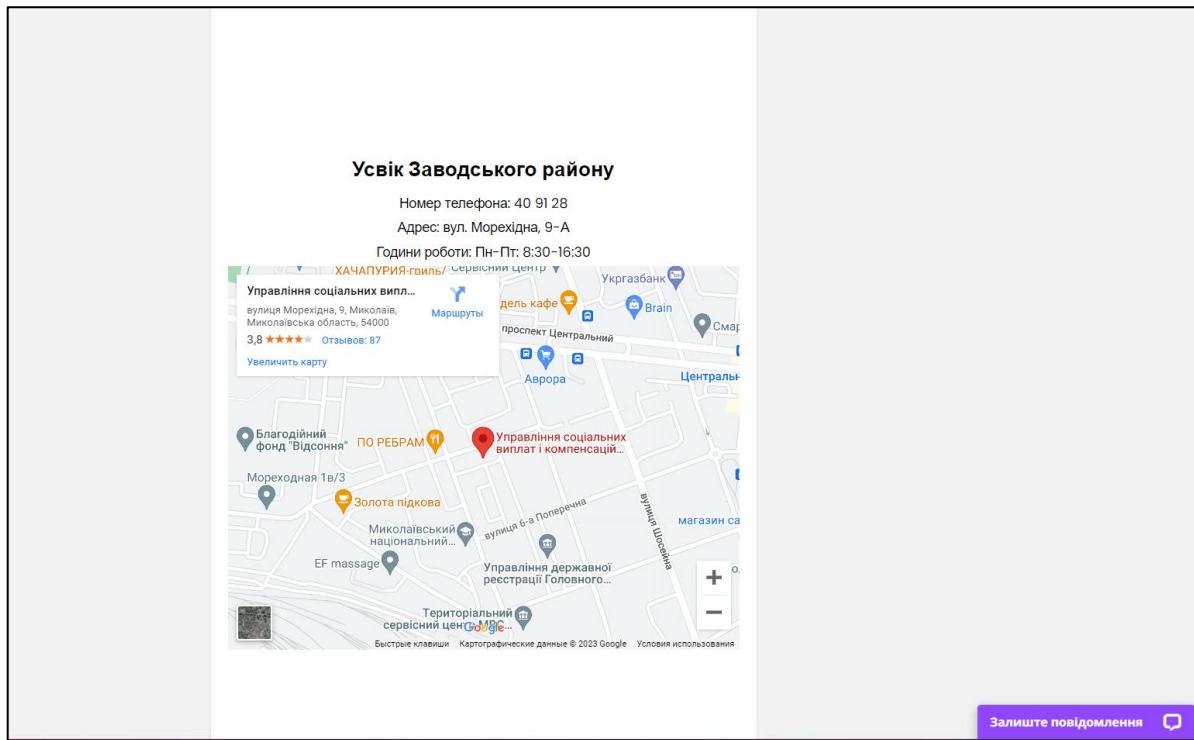


Рисунок 4.19 – Контакти УСВІК Заводського району

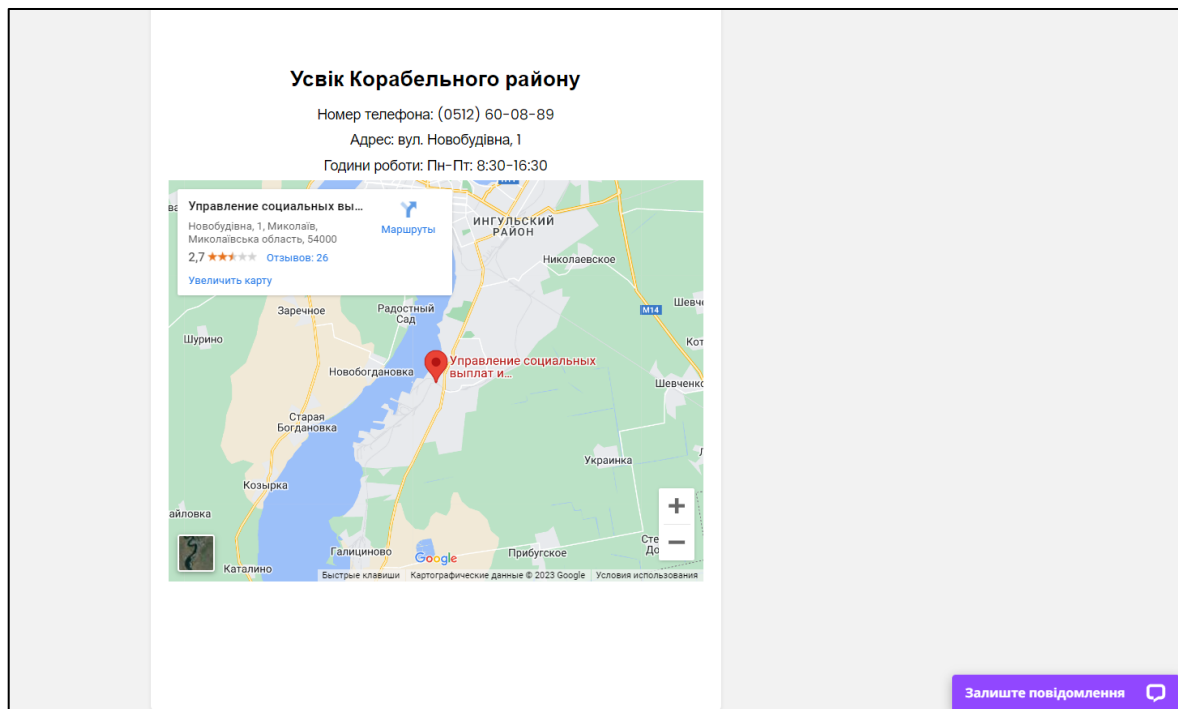


Рисунок 4.20 – Контакти УСВІК Корабельного району

Також на кожній сторінці знаходиться форма зворотного зв'язку. Для цього було застосовано сервіс Livechat.

Livechat – це програмне забезпечення, яке надає можливість вести живу онлайн-комунікацію з відвідувачами розробленого вебсайту чи застосунку. Він

дозволяє спілкуватися з клієнтами в режимі реального часу через чатове вікно на сайті [23].

Основні функції сервісу LiveChat включають:

- живий чат: співробітник УСВІК можете взаємодіяти з відвідувачами сайту в режимі реального часу. Клієнти можуть задавати питання, а відповідальний працівник соцзахисту можете надавати відповіді, надавати інформацію про послуги, допомагати вирішувати проблеми та багато іншого;

- контроль черги: співробітник УСВІК можете бачити, скільки відвідувачів знаходяться в черзі і очікують на обслуговування. Це дозволяє відповідальному працівнику соцзахисту керувати своїм часом та призначати пріоритети;

- аналітика: LiveChat надає статистику та аналітичні дані щодо задіяних чатів, включаючи час очікування, тривалість розмови та інші метрики. Це допомагає співробітнику УСВІК аналізувати продуктивність наявного сервісу підтримки та знаходити способи його вдосконалення;

- підтримка кількох каналів: LiveChat дозволяє інтегрувати різні канали комунікації, такі як чат, електронна пошта, соціальні мережі тощо, щоб відповідальні працівники соцзахисту могли ефективно спілкуватися з клієнтами з використанням різних засобів;

- налаштування та персоналізація: співробітник УСВІК може налаштовувати вигляд та функціональність чатового вікна, щоб воно відповідало іміджу та завданням установи соцзахисту.

Загалом, сервіс LiveChat надає зручний спосіб взаємодії з клієнтами в режимі реального часу та покращення якості обслуговування розробленого вебсайту.

На рис. 4.21 представлено код скрипту Livechat, який додається до розроблюваного вебзастосунку. Він дозволяє активувати та використовувати сервіс LiveChat, цей скрипт підключає необхідні ресурси та інструменти для взаємодії з LiveChat. Коли відвідувачі перебувають на розробленому сайті, вони можуть натиснути на посилання «Залиште повідомлення» й будуть

перенаправлені на чатове вікно LiveChat, де вони зможуть взаємодіяти з операторами чату УСВІК. Цей скрипт також додає елемент «noscript», який відображається у випадку, якщо відвідувач вимкнув JavaScript у своєму браузері. У такому разі відвідувач може скористатися посиланням «Залиште повідомлення», щоб перейти до сторінки LiveChat і отримати підтримку.

Важливо зазначити, що цей код працює в зв'язці з ліцензійним ключем (license key), який вказується у рядку `window.__lc.license = 15568989`. Користувач повинен замінити цей ключ на свій власний ліцензійний ключ LiveChat, який йому було видано при реєстрації на сервісі LiveChat.

```

<!-- Start of LiveChat (www.livechat.com) code -->
<script>
window.__lc = window.__lc || {};
window.__lc.license = 'YOUR_LICENSE_KEY';
(function(n,t,c){
  function i(n){
    return e._h?e._h.apply(null,n):e._q.push(n)
  }
  var e={
    _q:[],
    _h:null,
    _v:"2.0",
    on:function(){
      i(["on",c.call(arguments)])
    },
    once:function(){
      i(["once",c.call(arguments)])
    },
    off:function(){
      i(["off",c.call(arguments)])
    },
    get:function(){
      if(!e._h)throw new Error("[LiveChatWidget] You can't use getters before load.");
      return i(["get",c.call(arguments)])
    },
    call:function(){
      i(["call",c.call(arguments)])
    },
    init:function(){
      var n=t.createElement("script");
      n.async=!0;
      n.type="text/javascript";
      n.src="https://cdn.livechatinc.com/tracking.js";
      t.head.appendChild(n)
    }
  };
  !n.__lc.asyncInit&&e.init();
  n.LiveChatWidget=n.LiveChatWidget||e
})(window,document,[].slice)
</script>
<noscript>
<a href="https://www.livechat.com/chat-with/YOUR_LICENSE_KEY/" rel="nofollow">Chat with us</a>,
powered by <a href="https://www.livechat.com/?welcome" rel="noopener nofollow" target="_blank">LiveChat</a>
</noscript>
<!-- End of LiveChat code -->

```

Рисунок 4.21 – Скрипт Livechat

На рис. 4.22–4.25 представлена робота із Livechat.

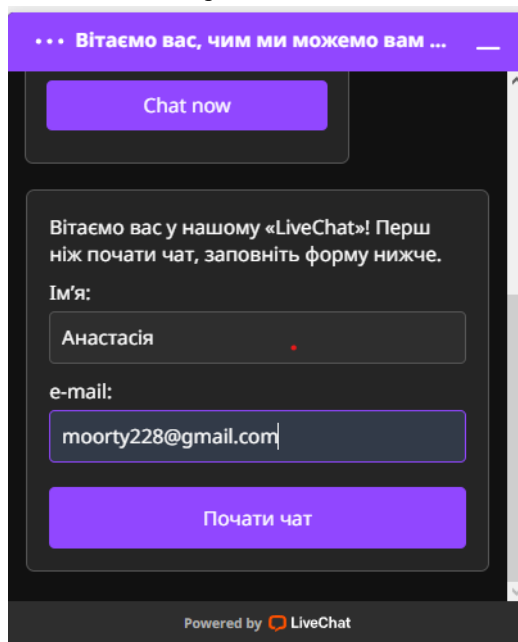


Рисунок 4.22 – Початок чату

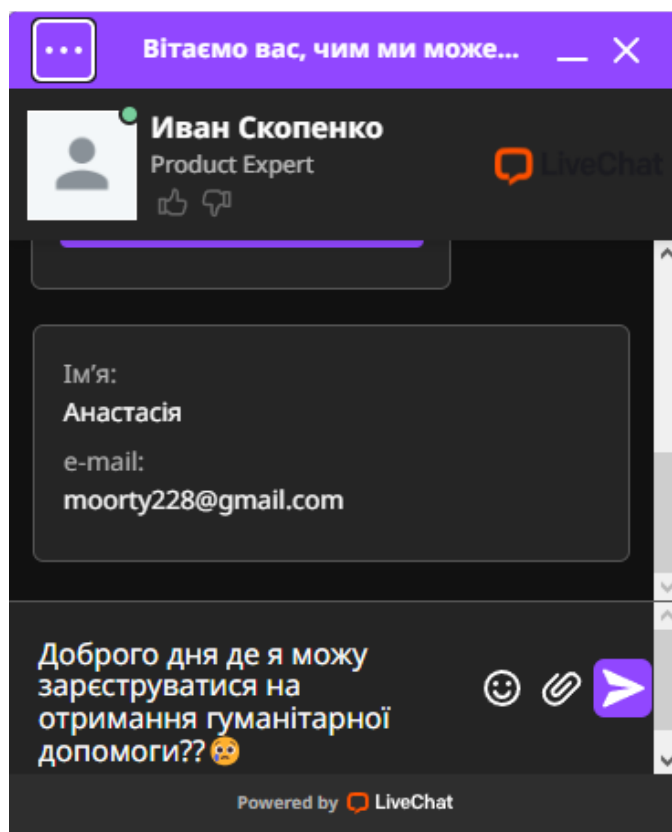


Рисунок 4.23 – Відправлення повідомлення

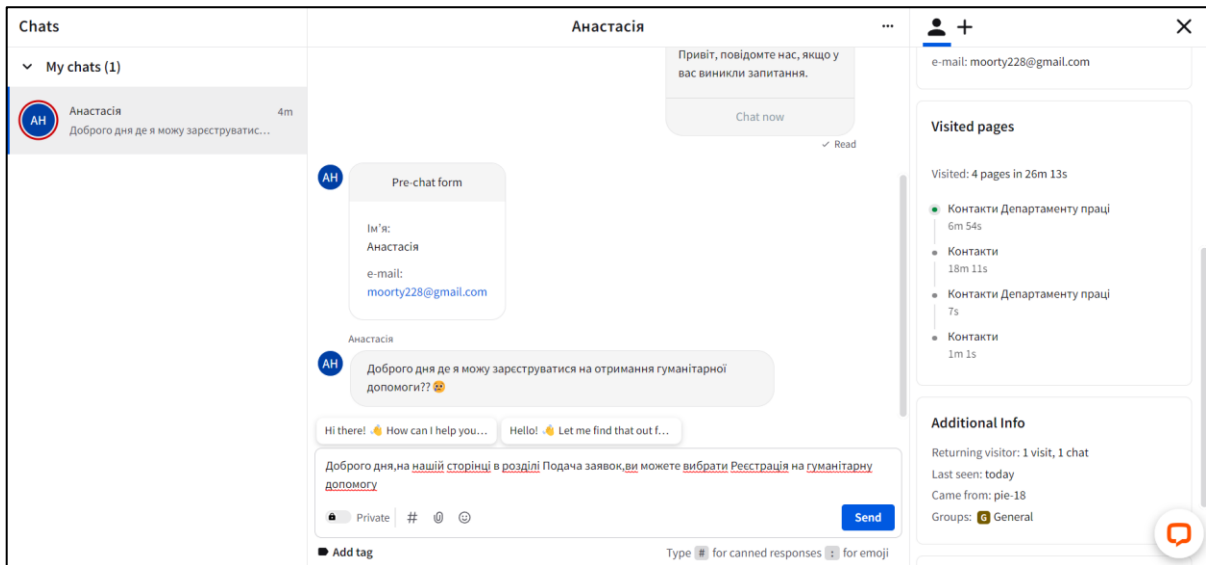


Рисунок 4.24 – Сторінка адміністратора

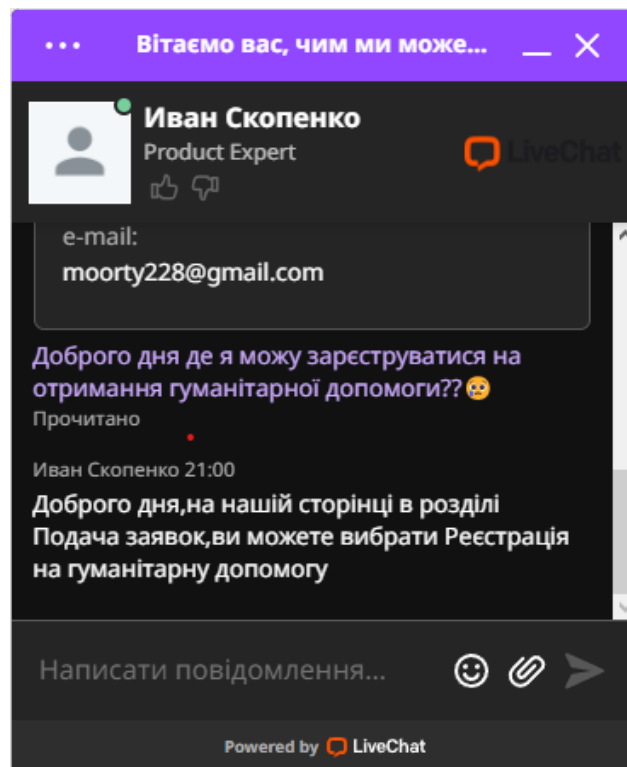


Рисунок 4.25 – Отримання повідомлення

4.2 Тестування вебзастосунку

У табл. 4.1–4.2 показані результати проведення тестування вебзастосунку.

Таблиця 4.1– Тестування проходження анкетування

User	Користувач
Objective.	Пройти анкетування
<p>A successful scenario:</p> <ol style="list-style-type: none"> 1. Користувач перейшов до форми анкетування. 2. Користувач заповнив необхідні дані. 3. Система отримала результат. 4. Користувач отримав повідомлення про успішне проходження. 	
Result	Користувач пройшов анкетування
Expansion	
1	Немає доступу до інтернет-підключення. Користувач не зможе ввести дані
2	Не всі заповнені поля при проходженні анкети. Користувач не зможе продовжити анкетування.
3	Відсутній доступ до БД. Система видає повідомлення про помилку. Користувач не зможе продовжити анкетування.

Таблиця 4.2 – Подача заявки на зустріч

User	Користувач
Objective.	Подача заявок на зустріч із представниками, отримання гуманітарної допомоги
A successful scenario: 1. Користувач вводить свої данні. 1. Користувач обрав організацію. 2. Система виводить повідомлення про успішну реєстрацію.	
Result	Користувач зареєструвався на зустріч, отримання гуманітарної допомоги
Expansion	
1	Користувач невірно ввів дані. Система виводить повідомлення про помилку. Користувач не може продовжувати реєстрацію.
2	Немає доступу до БД. Система виводить повідомлення про помилку. Користувач не може продовжувати реєстрацію

Висновки до розділу 4

Було розроблено інформаційний вебзастосунок соціального захисту населення Миколаївської територіальної громади, заснований на зібраних міркуваннях, спостереженнях та висновках.

Функціонал готового вебзастосунку має в собі розділи: Головна, Анкетування, Подача заявок, Інформування, Контакти. В ході роботи було показано, як кожен розділ працює та як користувач взаємодіє із ним.

Також було проведено тестування застосунку, під час якого було отримано позитивні результати.

ВИСНОВКИ

У ході виконання бакалаврської кваліфікаційної роботи було підтверджено актуальність розробки інформаційного вебзастосунку соціального захисту населення Миколаївської територіальної громади. В рамках огляду останніх публікацій та існуючих аналогів було виявлено, що інформаційні вебзастосунки соціального захисту населення активно розвиваються в інших регіонах та мають значний потенціал для поліпшення надання соціальних послуг та забезпечення доступу до необхідної інформації. Ці аналоги стали важливим джерелом для визначення найкращих практик та використання їх у власному проєкті.

Було проаналізовано технологій розробки та обрано платформу веброзробки та базу даних, які найкраще відповідають потребам проєкту. За допомогою цих технологій було розроблено інформаційний вебзастосунок, що забезпечує доступ до соціальних послуг для населення Миколаївської територіальної громади.

Макет сайту було розроблено з урахуванням зручного та інтуїтивно зрозумілого інтерфейсу для користувачів. Це дозволило створити зручну та ефективну систему надання соціальних послуг та отримання необхідної інформації.

Наприкінці було проведено тестування розробленого вебзастосунку, яке підтвердило його функціональність та надійність. Тестування дозволило виявити деякі недоліки та помилки, які були виправлені.

Зважаючи на актуальність інформаційного вебзастосунку соціального захисту населення Миколаївської територіальної громади, можна зробити висновок про необхідність його запровадження. Використання даного вебзастосунку сприятиме покращенню доступу до соціальних послуг та оптимізації процесу надання допомоги населенню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Скопенко І. В., Журавська І. М. Інформаційний вебзастосунок системи соціального захисту Миколаївської територіальної громади. *Інформаційні технології та інженерія* : тези доп. Всеукр. наук.-практ. конф. Миколаїв, 07–10 лют. 2023 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2023. С. 118–120.
2. Офіційний вебсайт департаменту соціальної політики Чернівецької міської ради. URL: <https://dsp.city.cv.ua/> (дата звернення 24.05.2023).
3. Офіційний вебсайт департаменту праці та соціальної політики Одеської міської ради. URL: <https://omr.gov.ua/ua/city/departments/dpsp/> (дата звернення 24.05.2023).
4. Офіційний вебсайт комунальної установи соціального захисту. URL: <https://dszn.smr.gov.ua/komunalni-ustanovi-departamentu/> (дата звернення 24.05.2023).
5. Закон України про правовий режим воєнного стану URL: <https://zakon.rada.gov.ua/laws/show/389-19#Text> (дата звернення 24.05.2023).
6. Закон України про надання соціальних послуг URL: <https://zakon.rada.gov.ua/laws/show/2671-19#Text> (дата звернення 24.05.2023).
7. Закон України про місцеве самоврядування URL: <https://zakon.rada.gov.ua/laws/show/280/97-%D0%B2%D1%80#Text> (дата звернення 24.05.2023).
8. Закон України про гуманітарну допомогу населення URL: <https://zakon.rada.gov.ua/laws/show/1192-14#Text> (дата звернення 24.05.2023).
9. Про звернення громадян : Закон України від 2 жовтня 1996 року № 393/96-ВР. URL: <https://zakon.rada.gov.ua/laws/show/393/96-%D0%B2%D1%80#Text> (дата звернення: 10.03.2023).
10. Про організацію надання соціальних послуг : Постанова Кабінету Міністрів України від 1 червня 2020 року № 587. URL: <https://zakon.rada.gov.ua/laws/show/587-2020-%D0%BF#Text> (дата звернення: 16.03.2023).

11. Постанова КМУ від 27 грудня 2001 року № 1751 *Про затвердження Порядку призначення і виплати державної допомоги сім'ям з дітьми* (зі змінами)
URL: <https://zakon.rada.gov.ua/laws/show/587-2020-%D0%BF#Text> (дата звернення 24.05.2023).
12. JavaScript history. URL: https://www.w3schools.com/js/js_history.asp (дата звернення 24.05.2023).
13. Webdesigner Depot. URL: <https://www.webdesignerdepot.com/> (дата звернення 24.05.2023).
14. Smashing Magazine. URL: <https://www.smashingmagazine.com/> (дата звернення 24.05.2023).
15. Learn to Code HTML & CSS. URL: <https://learn.shayhowe.com/html-css/> (дата звернення 24.05.2023).
16. Офіційна документація PHP. URL: <https://www.php.net/manual/en/> (дата звернення 24.05.2023).
17. PHP: The Right Way – комплексний гайд по розробці на PHP. URL: <https://phprightway.com/> (дата звернення 24.05.2023).
18. Акопян С. Як написати і продати свій PHP скрипт. 2021. 86 с.
19. MySQL документація. URL: <https://dev.mysql.com/doc/> (дата звернення 24.05.2023).
20. Що таке use case та для чого вони потрібні. URL: <https://training.qatestlab.com/blog/technical-articles/what-is-a-use-case-and-what-are-they-for/> (дата звернення 24.05.2023).
21. Офіційний сайт Symfony. URL: <https://symfony.com/> (дата звернення: 26.04.2023).
22. Miller B. D. Principles of web design. Allworth, 2022. 280 p.
23. Hong P. Practical web design: Learn the fundamentals of web design with HTML5, CSS3, Bootstrap, JQuery, and Vue.js. Packt Publ., 2018. 368 p.
24. Dorst K. Notes on design: How creative practice works. BIS Publishers, 2018. 208 p.
25. Wasserman E. A. As if by design: How creative behaviors really evolve.

Cambridge University Press, 2021. 328 p.

26. Структура сайту: основні види та правила їх розробки. URL: <https://webtune.com.ua/statti/web-rozrobka/struktura-sajtu/> (дата звернення 24.05.2023).

27. Livechat. URL: <https://www.livechat.com/ru/> (дата звернення 24.05.2023).

28. Huang W., Yang P. Application analysis and research of artificial intelligence technology in the creative stage of web design. *BCP Social Sciences & Humanities*. 2022. Vol. 20. P.535–543. DOI: 10.54691/bcpssh.v20i.2389.

ДОДАТОК А

Код програми

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Департамент праці та соціального захисту населення Миколаївської територіальної
громади</title>
  <link rel="stylesheet" href="style.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
<link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,600;1,700&dis
play=swap" rel="stylesheet">
<link      rel="stylesheet"      href="https://cdn.jsdelivr.net/npm/@fontawesome/fontawesome-
free@6.2.1/css/fontawesome.min.css">
</head>
<body>
  <section class="header">
    <nav>
      <a href="index.html"></a>
      <div class="nav-links" id="navLinks">
        <ul>
          <li><a href="index.html" >Головна</a></li>

          <li class="dropdown">
            <a href="#">Анкетування</a>
            <div class="dropdown-menu">
              <a href="test.php">Анкетування щодо якості наданих соціальних послуг</a>
              <a href="test2.php">Анкетування ВПО</a>
              <a href="#">Анкетування пільги</a>
            </div>
          </li>

          <li class="dropdown">
            <a href="">Подача заявки</a>
            <div class="dropdown-menu">
              <a href="register.php">Реєстрація на зустріч</a>
              <a href="pock.php">Реєстрація на гуманітарну допомогу</a>
            </div>
          </li>

          <li class="dropdown">
            <a href="news.html">Інформування</a>
            <div class="dropdown-menu">
            </div>
          </li>

          <li class="dropdown">
            <a href="#">Контакти</a>
            <div class="dropdown-menu">
              <a href="contact.html">Департамент</a>
              <a href="contact1.html">Усвік Центр</a>
              <a href="contact2.html">Усвік Інугл</a>
              <a href="contact3.html">Усвік Завод</a>
              <a href="contact4.html">Усвік Кораб</a>
            </div>
          </li>
        </ul>
      </div>
    </nav>
  </section>
</body>
</html>
```

```
</nav>
<div class="text-box">
  <h4>Департамент праці та соціального захисту населення<br>Миколаївської міської ради</h4>
  <p>Основним напрямком діяльності департаменту праці та соціального захисту населення є
забезпечення реалізації<br>державних та регіональних соціальних програм.</p>
  <a href="video.html" class="hero-btn">Інформативне відео</a>
</div>
</section>
<!-------Javascript for Toggle Menu----->
<script>

  var navLinks= document.getElementById("navLinks")
  function showMenu(){
    navLinks.style.right="0";
  }
  function hideMenu(){
    navLinks.style.right="-200px";
  }
</script>
</body>
</html>
```

Test.php

```
<!DOCTYPE html>
<html>
<head>
  <title>Анкета щодо якості наданих соціальних послуг</title>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,600;1,700&dis
play=swap" rel="stylesheet">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fontawesome/fontawesome-
free@6.2.1/css/fontawesome.min.css">
</head>
<body>

  <form>
    <label for="name">Ваше фіо:</label>
    <input type="text" id="name" name="name"><br><br>
    <label for="age">Район</label>
    <input type="text" id="neighborhood" name="neighborhood"><br><br>
    </select><br><br>
    <h2>Вопрос 1</h2>
    <p>Найменування отриманої послуги</p>
    <input type="radio" id="question1a" name="question1" value="a">
    <label for="question1a">А) Консультування</label><br>
    <input type="radio" id="question1b" name="question1" value="b">
    <label for="question1b">В) Посередництво</label><br>
    <input type="radio" id="question1c" name="question1" value="c">
    <label for="question1c">С) Представництво інтересів</label><br>
    <input type="radio" id="question1d" name="question1" value="d">
    <label for="question1d">D) Надання притулку</label><br>
    <input type="radio" id="question1e" name="question1" value="e">
    <label for="question1e">Е) Соціальна профілактика</label><br>
    <input type="radio" id="question1f" name="question1" value="f">
    <label for="question1f">F) Догляд та виховання дітей в умовах, наближених до сімейних</label><br>
    <input type="radio" id="question1g" name="question1" value="g">
    <label for="question1g">G) Підтримане проживання осіб похилого віку та осіб з
```

інвалідністю</label>

<input type="radio" id="question1h" name="question1" value="h">
<label for="question1h">H) Підтримане проживання бездомних осіб</label>

<input type="radio" id="question1i" name="question1" value="i">
<label for="question1i">I) Соціальний супровід сімей/осіб, які перебувають у складних життєвих обставинах</label>

<input type="radio" id="question1h" name="question1" value="h">
<label for="question1h">J) Фізичний супровід осіб з інвалідністю, які мають порушення опорно-рухового апарату та пересування на кріслах колісних, з інтелектуальними, сенсорними, фізичними, моторними, психічними та поведінковими порушеннями осіб</label>

<label for="question1j">Інше:</label>

<input type="text" id="question1_input" name="question1__input">

<h2>Вопрос 2</h2>
<p>Надавач соціальних послуг</p>
<input type="radio" id="question2a" name="question2" value="a">
<label for="question1a">A) Міський територіальний центр соціального обслуговування (надання соціальних послуг)</label>

<input type="radio" id="question2b" name="question2" value="b">
<label for="question1b">B) Центрі реінтеграції бездомних громадян</label>

<input type="radio" id="question2c" name="question2" value="c">
<label for="question1c">C) Міський центр комплексної реабілітації для дітей з інвалідністю</label>

<input type="radio" id="question2d" name="question2" value="d">
<label for="question1d">D) Міський геріатричний будинок мелосердя імені Святого Миколая</label>

<input type="radio" id="question2e" name="question2" value="e">
<label for="question1e">E) Благодійна організація «Миколаївський центр соціальної реабілітації «Відновлення»</label>

<input type="radio" id="question2f" name="question2" value="f">
<label for="question1f">F) Догляд та виховання дітей в умовах, наближених до сімейних</label>

<input type="radio" id="question2g" name="question2" value="g">
<label for="question1g">G) Підтримане проживання осіб похилого віку та осіб з інвалідністю</label>

<input type="radio" id="question2h" name="question2" value="h">
<label for="question1h">H) Громадська організація "Пенітенціарна ініціатива"</label>

<h2>Вопрос 3</h2>
<p>Звідки Ви дізналися про соціальну послугу, яку отримали?</p>
<label for="question3j">Інше:</label>

<input type="text" id="question3_input" name="question3__input">

<h2>Вопрос 4</h2>
<p>Чи відповідає Вашим потребам послуга, яку Ви отримали (отримуєте)?</p>
<input type="radio" id="question4a" name="question4" value="a">
<label for="question4a">A) Так</label>

<input type="radio" id="question4b" name="question4" value="b">
<label for="question4b">B) Ні</label>

<label for="question4">Інше:</label>

<input type="text" id="question4_input" name="question4__input">

<h2>Вопрос 5</h2>
<p>Чи задоволені Ви ставленням до себе соціального працівника?</p>
<input type="radio" id="q5a1" name="question5" value="yes">
<label for="q5a1">Да</label>

<input type="radio" id="q5a2" name="question5" value="no">
<label for="q5a2">Нет</label>

<input type="radio" id="q5a3" name="question5" value="other">
<label for="q5a3">Другое</label>

<input type="text" id="question5_input" name="question5__input">

<h2>Вопрос 6</h2>
<p>Чи влаштує Вас перелік соціальних послуг, які надає надавач соціальних послуг?</p>
<input type="radio" id="q6a1" name="question6" value="yes">

```
<label for="q6a1">Да</label><br>
<input type="radio" id="q6a2" name="question6" value="no">
<label for="q6a2">Нет</label><br>
<input type="radio" id="q6a3" name="question6" value="other">
<label for="q6a3">Другое</label><br>
<input type="text" id="question6_input" name="question6__input"><br><br>

<h2>Вопрос 7</h2>
<p>Чи зручно Вам відвідувати надавача соціальних послуг?</p>
<input type="radio" id="q7a1" name="question7" value="yes">
<label for="q7a1">Да</label><br>
<input type="radio" id="q3a2" name="question7" value="no">
<label for="q7a2">Нет</label><br>
<input type="radio" id="q3a3" name="question3" value="other">
<label for="q7a3">Другое</label><br>
<input type="text" id="question7_input" name="question7__input"><br><br>

<h2>Вопрос 8</h2>
<p>Чи відповідає надання соціальної послуги Вашому індивідуальному плану?</p>
<input type="radio" id="q8a1" name="question8" value="yes">
<label for="q8a1">Да</label><br>
<input type="radio" id="q8a2" name="question8" value="no">
<label for="q8a2">Нет</label><br>
<input type="radio" id="q8a3" name="question8" value="other">
<label for="q8a3">Другое</label><br>
<input type="text" id="question8_input" name="question8__input"><br><br>

<h2>Вопрос 9</h2>
<p>Побажання щодо підвищення якості соціальної послуги чи діяльності надавача соціальних послуг</p>
<input type="text" id="question9_input" name="question9__input"><br><br>
<input type="submit" value="Отправить">
</form>

<?php
if (isset($_GET['formSubmit'])) {
    $nameform = $_GET['name'];
    $neighborhoodform = $_GET['neighborhood'];
    $question1form = $_GET['question1'];
    $question1_inputform = $_GET['question1_input'];
    $question2form = $_GET['question2'];
    $question3_inputform = $_GET['question3_input'];
    $question4form = $_GET['question4'];
    $question4_inputform = $_GET['question4_input'];
    $question5form = $_GET['question5'];
    $question5_inputform = $_GET['question5_input'];
    $question6form = $_GET['question6'];
    $question6_inputform = $_GET['question6_input'];
    $question7form = $_GET['question7'];
    $question7_inputform = $_GET['question7_input'];
    $question8form = $_GET['question8'];
    $question8_inputform = $_GET['question8_input'];
    $question9_inputform = $_GET['question9_input'];
    $connection = new mysqli("localhost", "root", "", "test1");
    if ($connection->connect_errno) {
        echo "Вибачте, сталася помилка на сайті";
        exit;
    }
    $name = $connection->real_escape_string($nameform);
    $neighborhood = $connection->real_escape_string($neighborhoodform);
    $question1 = $connection->real_escape_string($question1form);
    $question1_input = $connection->real_escape_string($question1_inputform);
    $question2 = $connection->real_escape_string($question2form);
    $question3_input = $connection->real_escape_string($question3_inputform);
```

```

$question4 = $connection->real_escape_string($question4form);
$question4_input = $connection->real_escape_string($question4_inputform);
$question5 = $connection->real_escape_string($question5form);
$question5_input = $connection->real_escape_string($question5_inputform);
$question6 = $connection->real_escape_string($question6form);
$question6_input = $connection->real_escape_string($question6_inputform);
$question7 = $connection->real_escape_string($question7form);
$question7_input = $connection->real_escape_string($question7_inputform);
$question8 = $connection->real_escape_string($question8form);
$question8_input = $connection->real_escape_string($question8_inputform);
$question9_input = $connection->real_escape_string($question9_inputform);
$connection = new mysqli("localhost", "root", "", "test1");
if ($connection->connect_errno) {
    echo "Вибачте, сталася помилка на сайті";
    exit;
}
$query = "INSERT INTO `test`(name, neighborhood, question1, question1_input, question2, question3_input,
question4, question4_input, question5, question5_input, question6, question6_input, question7, question7_input,
question8, question8_input, question9_input)
VALUES ('$name', '$neighborhood', '$question1', '$question1_input', '$question2', '$question3_input',
'$question4', '$question4_input', '$question5', '$question5_input', '$question6', '$question6_input', '$question7',
'$question7_input', '$question8', '$question8_input', '$question9_input')";
if ($connection->query($query) === false) {
    echo "Помилка: " . $connection->error;
    exit;
} else {
    echo "Успішно" . '<br>';
}
$connection->close();
}
?>
</body>
</html>

```

Register.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <form method="get">
        <label for="name">Имя:</label><br>
        <input type="text" id="name" name="name"><br>

        <label for="surname">Призвище:</label><br>
        <input type="text" id="surname" name="surname"><br>

        <label for="phone">Номер телефона:</label><br>
        <input type="tel" id="phone" name="phone"><br>

        <label for="organization">Оберіть організацію:</label><br>
        <select id="organization" name="organization">
            <option value="Департамент Праці">Департамент Праці</option>
            <option value="Усвік Центр">Усвік Центрального району</option>
            <option value="Усвік Завод">Усвік Заводського району</option>
            <option value="Усвік Інгул">Усвік Інгульського району</option>
            <option value="Усвік Кораб">Усвік Корабельного району</option>
        </select>
    </form>

```



```

        <option value="Терцентр">Територіальний Центр</option>
    </select><br><br>
    <label for="status">Статус:</label><br>
    <input type="text" id="status" name="status"><br><br>

    <input type="submit" name="formSubmit">
</form>

<?php
if (isset($_GET['formSubmit'])){
    $nameform = $_GET['name'];
    $surnameform = $_GET['surname'];
    $phoneform = $_GET['phone'];
    $organizationform = $_GET['organization'];
    $statusform = $_GET['status'];
    $mysqli = new mysqli("localhost", "root", "", "pie-18");
    if ($mysqli->connect_errno) {
        echo "Извините, возникла ошибка на сайте";
        exit;
    }
    $name = $mysqli->real_escape_string($nameform);
    $surname = $mysqli->real_escape_string($surnameform);
    $phone = $mysqli->real_escape_string($phoneform);
    $organization = $mysqli->real_escape_string($organizationform);
    $status = $mysqli->real_escape_string($statusform);
    $query = "INSERT INTO users (name,surname,phone,organization,status) VALUES
('$name','$surname','$phone','$organization','$status')";
    if ($mysqli->query($query) === false) {
        echo "Ошибка: " . $mysqli->error;
        exit;
    } else {
        echo "Успішно" . '<br>';
    }
    $mysqli->close();
}
?>
</body>
</html>

```

Contact.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Контакти Департаменту праці</title>
    <link rel="stylesheet" href="style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,600;1,700&dis
play=swap" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"
integrity="sha512-
Fzz+y+x1yFviwktBwJW8uP7LYYZZBdF+6zIrgXbPf6zJJT0TmmTp0/4UH4Qe6k0LuJdQOxH0fdX+HPNOy
xJiGQ==" crossorigin="anonymous"/>
</head>
<body>
    <div class="contact-container">
        <h2>Контакти</h2>
        <div class="contact-info">

```

```

<div class="contact-item">
  <div class="contact-icon"><i class="fas fa-map-marker-alt"></i></div>
  <div class="contact-text">Адреса: Вулиця Мала Морська, будинок 19, Місто Миколаїв</div>
</div>
<div class="contact-item">
  <div class="contact-icon"><i class="fas fa-phone-alt"></i></div>
  <div class="contact-text">Телефон: +380512676392</div>
</div>
<div class="contact-item">
  <div class="contact-icon"><i class="fa-solid fa-clock"></i></div>
  <div class="contact-text">Графік роботи: Пн-Пт: 9:00-18:00, Сб-Нд: вихідний</div>
</div>
</div>
<div class="map-container">
  <!-- здесь может быть встроенная карта, например, Google Maps -->
  <iframe
src="https://www.google.com/maps/embed?pb=!1m14!1m3!1d10890.417269027908!2d31.9966225!3d46.9694638!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x40c5c92fd7095251%3A0xa6115fcc55cd0a20!2z0JTQtdC_0LDRgNGC0LDQvNC10L3RgiDRgtGA0YPQtNCwINC4INGB0L7RhtC40LDQu9GM0L3QvtC5INC30LDRidC40YLRiyDQvdCw0YHQtdC70LXQvdC40Y8g0J3QuNC60L7Qu9Cw0LXQstGB0LrQvtCz0L4g0LPQvtGA0L7QtNGB0LrQvtCz0L4g0YHQvtCy0LXRgtCw!5e0!3m2!1sru!2sua!4v1234567890" width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy"></iframe>
  </div>
</div>
</body>
<!-- Start of LiveChat (www.livechat.com) code -->
<script>
  window.__lc = window.__lc || {};
  window.__lc.license = 15568989;
  ;(function(n,t,c){function i(n){return e._h?e._h.apply(null,n):e._q.push(n)}var e={_q:[],_h:null,_v:"2.0",on:function(){i(["on",c.call(arguments)])},once:function(){i(["once",c.call(arguments)])},off:function(){i(["off",c.call(arguments)])},get:function(){if(!e._h)throw new Error("[LiveChatWidget] You can't use getters before load.");return i(["get",c.call(arguments)])},call:function(){i(["call",c.call(arguments)])},init:function(){var n=t.createElement("script");n.async=!0,n.type="text/javascript",n.src="https://cdn.livechatinc.com/tracking.js",t.head.appendChild(n)};!n.__lc.asyncInit&&e.init(),n.LiveChatWidget=n.LiveChatWidget|e})(window,document,[],.slice))
</script>
<noscript><a href="https://www.livechat.com/chat-with/15568989/" rel="nofollow">Chat with us</a>, powered by <a href="https://www.livechat.com/?welcome" rel="noopener nofollow" target="_blank">LiveChat</a></noscript>
<!-- End of LiveChat code -->

```

ДОДАТОК Б

Матеріали апробації роботи

Міністерство освіти і науки України
Чорноморський національний університет
імені Петра Могили



«Інформаційні технології та інженерія»

*Всеукраїнська науково-практична конференція
молодих вчених, аспірантів і студентів*

ТЕЗИ

7–10 лютого 2023 року

Миколаїв – 2023

Кузнецов С. С. Інтелектуальні компоненти комп'ютерної гри на базі рушія Render	102
Ракетко В. С., Антіпова К. О. Програмне забезпечення класифікації Telegram-коментарів для аналізу настроїв та визначення суспільної думки	104
Сирова В. В., Чуйко Г. П. Використання математичного моделювання у відеоіграх та штучному інтелекті. Скіншений автомат	106
Степанчук Д. К., Калодіба І. О. Розробка генератора синтаксичних аналізаторів з підтримкою кількох алфавітів у входних правилах	108
Слюсар С. Д., Бірей К. О. Універсальний програмний комплекс системи контролю ваги	109
Явницька Д. В., Калодіба І. О. Процедура генерація при створенні комп'ютерних ігор	112

Вебтехнології та вебдизайн

Олійник М. Г., Болубан І. М. Оптимізація продуктивності прогресивного вебзастосунок бібліотеки на основі моделі RAII	114
Панасюк О. О., Швед А. В. Сервіси та інструменти вебаналітики	116
Скопенко І. В., Журавська І. М. Інформаційний вебзастосунок системи соціального захисту Миколаївської територіальної громади	118
Шляпка С. О., Кузаківська І. В. Розширення Google Chrome та кореляція їх розвитку з покращенням користувацького досвіду	120

Інформаційні технології у навчальному процесі

Ishchenko O., Ishchenko N. An overview of prerequisites for IoT and AI-driven tutoring in higher education	123
Бычкова О. Можливість засобів сучасних технологій в навчальному процесі студентів філологічних спеціальностей	125
Зуй О. М., Коренкова Г. В., Стугалов С. А. Особливості використання графічного редактору FIGMA у навчальному процесі	127
Корфанова О. В., Корфанов О. Є. Комп'ютерне оцінювання рівня сформованості хімічних компетентностей у здобувачів вищої технічної освіти	128

канали, оптимізувати ваші кампанії та приймати обґрунтовані рішення про те, на чому зосередити свої зусилля.

Використання декількох метрик у послідній може забезпечити більш повне розуміння активності користувачів.

Попередніми інструментами для аналізу метричних даних є Google Analytics, Adobe Analytics [3].

Google Analytics – популярний сервіс вебаналітики, що пропонує такі функції, як відстеження перегляд сторінок і сеансів, сегментація користувачів і постановка цілей.

Adobe Analytics – це потужний сервіс вебаналітики, що дає змогу відстежувати поведінку користувачів, збирати дані та складати звіти.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Світлул А. Маркетингова аналітика. Як підкріпити інтуїцію даними. Наш Форум, 2019. 220 с.
2. Schmidt E., Rosenberg J. How Google works. Grand Central Publishing, 2017. 320 p.
3. 34 Best Web Analytics Tools To Use In 2023. URL: <https://www.leadfeeder.com/blog/web-site-analytics-tools/> (дата звернення: 25.01.2023)

УДК 004.75

Скопенко І. В., Журавська І. М.
Черноморський національний університет ім. Петра Могили,
Миколаїв, Україна

ІНФОРМАЦІЙНИЙ ВЕБЗАСТОСУНОК СИСТЕМИ СОЦІАЛЬНОГО ЗАХИСТУ МИКОЛАЇВСЬКОЇ ТЕРИТОРІАЛЬНОЇ ГРОМАДИ

Стрімкий ріст проблем громади у період пандемії і війни загострив потребу у віддалених зверненнях про соціальний захист. Ефективними технологіями організації процесів комунікації і допомоги є цифрові і застосування мережних, хмарних сервісів і вебтехнологій. Незважаючи на загальну потребу і актуальність, організація роботи таких систем зустрічає систематичні проблеми і неузгодженості. Різний рівень загального розвитку громади та службовців ускладнює і гальмує процес проходження звернень. Актуальність проблеми віддалених звернень, проблем впровадження системи та не готовності до роботи з нею у своїй сукупності є

безперечна. Її підтверджує зростання звернень зареєстрованих департаментом соціального захисту населення Миколаївської міської ради та перелік скарг на роботу системи із боку громадян та персоналу. У зв'язку з цим одним із завдань Департаменту, є пошук інновацій для забезпечення процесу розгляду і надання соціальних послуг та соціальної підтримки незахисних верств населення. Одним з інновацій, що розглядається та апробується є розробка інформаційного супроводу з використанням цифрових технологій.

Для реалізації головного завдання в роботі департаменту як актуальну задачу поставлено допомогти людям в організації надання соціальних послуг.

Гіпотезою дослідження було висунуто: приклади, підказки, тренування, що забезпечуються вікнами опцій, які необхідно використовувати з тлумаченням термінів, з функціональними моделями дій та звуковими та відео тлумаченнями є допомоготою сукупних кроків, що покращать рівень програмного інтерфейсу і забезпечать роботу у системі електронних звернень.

Метою є: підвищити ефективність роботи з системою допомоги шляхом впровадження додаткового сайту підтримки, що застосовує засоби сучасних видів комунікацій для розтлумачень і прикладів з урахуванням різного рівня підготовленості.

Об'єктом було обрано процеси збору інформації, її обробки і збереження.

Предметом моделі, методи, алгоритми аналізу і формування консультативної допомоги при роботі із системою електронних звернень громадян.

Для досягнення поставленої мети було сформульовано наступні задачі:

- провести аналіз предметної області та існуючих аналогів;
- розробити структуру та провести моделювання і випробування інформаційного вебзастосунку;
- описати технології, структуру, тестові протоколи, за допомогою яких застосунок потрібно представити на тестове випробування для завершення проекту.

Аналіз проблемних випадків і труднощів з якими зіткнулись співробітники департаменту та громадянами при роботі з електронною системою звернень дозволив усвідомити наочну потребу застосування динамічних алгоритмів. Після аналізу та перших спроб застосувати традиційні методи будови сайту і особливо інтерфейсу став очевидним необхідність застосунок мовою РНР сайт робить його статичним та складним у розумінні. А також головна ідея інтерактивної взаємодії такими програмними засобами створена не буде. У зв'язку з цим було

взято за основу алгоритм створення застосунку з використанням JavaScript для реалізації кращого алгоритму комунікаційної дії користувача та комп'ютера. Середовищем для розробки програми було обрано IntelliJ IDEA. Кінцевий продукт буде мати широкий спектр можливостей, а саме буде містити інформацію про надання соціальних послуг, умови надання соціальних послуг, аналіз якості надання соціальних послуг та можливості оформлення компенсації за надання соціальних послуг.

Успіхи розв'язку більшість поставлених на етапі планування вимог буде реалізовано шляхом спільних зусиль співробітників департаменту та за оцінкою матеріалів тестових прототипів застосунку.

Створення вебзастосунків як інструменту консультації і повільно, що ґрунтується на принципі інтерактивної взаємодії відкриває можливість покращення обслуговування громадян різного рівня готовності інструментами цифрових технологій.

УДК 004

*Шлявак С. О., Куляковська І. В.,
Черноморський національний університет ім. Петра Могили,
Миколаїв, Україна*

РОЗШИРЕННЯ GOOGLE CHROME ТА КОРЕЛЯЦІЯ ЇХ РОЗВИТКУ З ПОКРАЩЕННЯМ КОРИСТУВАЦЬКОГО ДОСВІДУ

Google Chrome пропонують ряд корисних функцій і переваг для користувачів. Розширення – це невеликі програмні програми, які можна додати до Chrome, щоб налаштувати та покращити вебперегляд користувача. Розширення доступні для різноманітних цілей, включаючи продуктивність, розваги, безпеку та конфіденційність.

Однією з найбільш популярних переваг розширень Google Chrome є підвищення продуктивності, яку вони пропонують. Надаючи користувачам легкий доступ до додаткових функцій, розширення можуть допомогти користувачам заощадити час, оптимізувати робочий процес і швидко отримати доступ до потрібної інформації. Популярні приклади розширень продуктивності включають LastPass, який безпечно зберігає паролі та дозволяє користувачам швидко входити в декілька облікових записів. Іншим прикладом є Grammarly, який перевіряє орфографію та граматику, коли користувачі вводять текст.