

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ТАРАБАНОВИЧ Іван-Маркіян Юрійович

**Веб-орієнтована інформаційна система надання
адміністративних послуг у період воєнного стану / Web-oriented
information system for providing administrative services during martial
law**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-41
І.-М. Ю. Тарабанович

Науковий керівник:
к.т.н., доцент І. В. Турченко

Дипломний проект допущено до
захисту

«___»_____ 2024 р.

Завідувач кафедри
_____ М. П. Комар

Тернопіль – 2024

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
« ____ » _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Тарабанович Іван-Маркіян Юрійович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Веб-орієнтована інформаційна система надання адміністративних послуг у період воєнного стану / web-oriented information system for providing administrative services during martial law
керівник роботи Турченко Ірина Василівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 12 грудня 2023 р. № 753.

2. Строк подання студентом закінченої кваліфікаційної роботи 15 травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- провести опис предметної області;
- провести огляд існуючих веб-орієнтованих інформаційних систем для надання адміністративних послуг;
- зробити постановку задачі дослідження;
- представити загальну структуру, алгоритмічне та інформаційне забезпечення веб-орієнтованої інформаційної системи;
- представити архітектуру системи, вибір технологій та інструментів;
- розробити інтерфейс користувача;
- програмно реалізувати та протестувати систему;

5. Перелік графічного матеріалу в роботі:

- схема алгоритму авторизації користувача;
- схема алгоритму взаємодії користувача з веб-орієнтованою інформаційною системою.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2024 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2024 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2024 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2024 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2024 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 20.05.2024 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту у системі «Unichек».	до 10.06.2024 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 14.06.2024 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії.	до 14.06. 2024 р.	

Студент _____ І.Ю. Тарабанович
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ І.В. Турченко
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Веб-орієнтована інформаційна система надання адміністративних послуг у період воєнного стану» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 61 сторінок і містить 23 ілюстрації, 2 таблиці, 4 додатки та 26 використаних джерел.

Мета роботи полягає у розробці веб-орієнтованої інформаційної системи для надання адміністративних послуг у період воєнного стану. Ця система має забезпечити безперебійне, надійне та ефективне обслуговування громадян через мережу інтернет.

У роботі використовувалися методи системного аналізу і синтезу, порівняльного аналізу. Проведено аналіз існуючих рішень, таких як національні веб-портали "ДІЯ" та "iGov", що використовуються для надання електронних державних послуг. Представлено загальну структуру, алгоритмічне та інформаційне забезпечення веб-орієнтованої інформаційної системи, програмно реалізовано та протестовано систему.

У результаті розроблено веб-орієнтовану інформаційну систему надання адміністративних послуг у період воєнного стану, яка забезпечує оперативний доступ до переліку документів, які потрібно підготувати для подальшого використання та подання у вищі органи, координацію допомоги та інформаційну підтримку для військовослужбовців, їхніх сімей та внутрішньо переміщених осіб.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ПОСЛУГИ, КОРИСТУВАЧ, ІНТЕРФЕЙС.

ANNOTATION

The qualification work on the topic "Web-Oriented Information System for Providing Administrative Services During Martial Law" for the Bachelor's degree in the specialty 122 "Computer Science" of the educational program "Computer Science" is written in 61 pages and contains 23 illustrations, 2 tables, 4 appendices, and 26 sources.

The goal of this work is to develop a web-oriented information system for providing administrative services during martial law. This system aims to ensure uninterrupted, reliable, and efficient service to citizens via the internet.

The work used methods of system analysis and synthesis, comparative analysis. An analysis of existing solutions, such as the national web portals "DIIA" and "iGov" which are used to provide electronic government services was carried out. The general structure, algorithmic and information support of the web-oriented information system is presented, the system is implemented and tested.

As a result, a web-oriented information system for providing administrative services during martial law was developed, ensuring quick access to the list of documents needed for further submission to higher authorities, coordination of assistance, and informational support for military personnel, their families, and internally displaced persons.

Keywords: INFORMATION SYSTEM, SERVICES, USER, INTERFACE.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та постановка задачі проектування.....	9
1.1 Теоретичні підходи до розробки веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану	9
1.2 Огляд і аналіз існуючих аналогів	12
1.4 Постановка задачі проектування	18
2 Загальна структура та інформаційне забезпечення системи надання адміністративних послуг у період воєнного стану	20
2.1 Загальна структура проекрованої системи	20
2.2 Алгоритмічне забезпечення системи	23
2.3 Інформаційне забезпечення веб-орієнтованої системи.....	27
3 Реалізація веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану	31
3.1 Програмна реалізація	31
3.2 Інтерфейс користувача.....	35
3.3 Тестування інформаційної системи.....	40
Висновки	43
Список використаних джерел	44
Додаток А Програмний код верстки сторінок веб-орієнтованої інформаційної системи	46
Додаток Б Тест-кейси.....	51
Додаток В Довідка про використання результатів	53
Додаток Г Копія опублікованих результатів.....	55

ВСТУП

Надання адміністративних послуг у період воєнного стану відрізняється від звичайних умов мирного часу з-за ряду специфічних чинників, які виникають в умовах війни. Наприклад, обмеження доступу до різних територій та об'єктів, включаючи установи, де надаються юридичні послуги. Це може бути обумовлено безпековими причинами або стратегічними обставинами. Саме така невизначеність та нестабільність можуть ускладнити процес надання адміністративних послуг. Непередбачувані обставини та велика кількість факторів, які впливають на умови життя громадян, можуть змінювати вимоги та потреби у допомозі [1].

У таких умовах розробка веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану, що забезпечить швидкий та зручний доступ до інформації та консультаційну підтримку для військовослужбовців, їх сімей та внутрішньо переміщених осіб, є дуже актуальною задачею.

Метою кваліфікаційної роботи є створення веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану. Для реалізації мети необхідно виконати наступні завдання:

- описати предметну область;
- провести аналіз існуючих аналогів та зробити постановку задачі проектування;
- представити загальну структуру, алгоритмічне та інформаційне забезпечення веб-орієнтованої інформаційної системи;
- програмно реалізувати та протестувати систему.

Об'єкт проєктування: методи та технології отримання, зберігання, обробки, передачі та використання інформації.

Предметна область проєктування: методи та технології розробки веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану.

Результати, отримані у кваліфікаційній роботі, використані в діяльності кафедри адміністративного права та судочинства Західноукраїнського національного університету, що підтверджується довідкою про використання (Додаток В).

Публікації та апробація кваліфікаційної роботи.

Результати опубліковано та апробовано в матеріалах VII міжнародної науково-практичної конференції «Сучасні світові тенденції розвитку науки та інформаційних технологій», м. Одеса, 30-31.05.2024 р. (Додаток Г).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ПРОЕКТУВАННЯ

1.1 Теоретичні підходи до розробки веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану

Надання адміністративних послуг у період воєнного стану є важливою складовою діяльності державних органів та місцевого самоврядування, ось деякі причини:

- у період воєнного стану громадяни залишаються учасниками соціальних відносин, які потребують певних адміністративних послуг, таких як доступ до медичної допомоги, соціальних виплат, житлово-комунальних послуг тощо;
- надання адміністративних послуг в період воєнного стану допомагає забезпечити певний рівень нормального функціонування суспільства, що сприяє збереженню соціального порядку та підтримці громадської стабільності;
- захист прав і свобод громадян, послуги можуть бути необхідним інструментом для цього, зокрема у зв'язку з організацією евакуації, захистом від переслідування, забезпеченням доступу до правосуддя тощо;
- швидке та ефективне надання адміністративних послуг сприяє підтримці довіри громадян до влади та сприяє стабілізації ситуації в країні під час кризи.

Від початку введення воєнного стану деякі адміністративні послуги надаються або обмежуються [1]. Під час війни для громадян України наявність додатку «Дія» є дуже важливим і корисним, адже у ньому є електронне відображення інформації, що міститься у їх документах.

Також на час воєнного стану введені істотні зміни з питань соціального захисту. Разом з тим, з'явилися категорії громадян, яких у мирний час не було у таких великих кількостях..

У такий час юридична обізнаність громадян стає дуже важливою, оскільки це дозволяє їм знати свої права та обов'язки в умовах надзвичайної ситуації. Знання юридичних аспектів може допомогти громадянам захищати свої інтереси,

отримувати необхідні підтримку та послуги, а також виявляти розуміння щодо обмежень, що можуть бути введені в умовах воєнного стану.

Сьогодні найлегший шлях зробити доступними інформаційні послуги – це створення веб-орієнтованої системи, яка буде доступною для усіх, хто володіє мобільним пристроєм та Інтернетом.

Веб-орієнтована інформаційна система є програмою, яка виконує функції через веб-браузер, незалежно від пристрою та платформи [14]. Він може бути простим наприклад, дошкою оголошень або контактною формою на Веб-орієнтованій інформаційній системі, або складним як текстовий процесор або мобільна гра з багатьма користувачами [5].

Зберігання та обробка інформації в Інтернеті відбувається за допомогою веб-орієнтованих інформаційних систем, які можуть використовуватися як у локальних, так і в мережевих середовищах. Вони базуються на веб-додатках, що дозволяють автоматизовано виконувати різноманітні завдання на веб-серверах та стороні користувача. Користувацькі інтерфейси веб-додатків реалізуються через веб-браузери. Для створення веб-додатків використовуються клієнтські та серверні технології.

Процес створення веб-орієнтованих інформаційних систем аналогічний процесу написання програм. Він включає стадії визначення вимог, аналізу, реалізації, проектування та тестування. При роботі з веб-платформою, незалежно від використовуваних технологій, користувач взаємодіє з ним так само, як і з звичайною програмою: він вводить дані, отримує інформацію, взаємодіє з вікнами і меню, зберігає дані на сервері та отримує відповіді [14]. Для зручності користувачів, системі буде наданий власний домен, тобто унікальне ідентифікаційне посилання, за яким можна перейти на неї з будь-якого пристрою, що має підключення до Інтернету [5].

При підборі імені домена потрібно враховувати специфіку ресурсу. Важливо підійти до цього питання серйозно, так як воно впливає на просування сайту в інтернеті. Правильне ім'я дає користувачеві коротку інформацію про те, що є на сторінці і чи варто туди заходити [4].

При створенні власного домену слід дотримуватись наступних правил, щоб посилання на систему було ефективним та унікальним:

- назва повинна бути змістовною та придатною до читання, варто придумати щось креативне та оригінальне;
- варто уникати довгих речень і складних словосполучень, доцільно використовувати короткі вирази;
- не використовувати сторонні символи (цифри, тире, значки, відсотки), якщо вони не містять сенсу, адже вони погано запам'ятовуватимуться користувачем;
- для створення домену системи в назві варто зазначити діяльність, це дозволить користувачам легко знаходити та використовувати систему, необхідно лише ввести домен у веб-браузері, щоб отримати доступ до всіх її функцій та можливостей.

Перелік характеристик, компонентів та приклади використання ІС наведено нижче.

Характеристики веб-орієнтованих систем:

- доступність через веб-браузер: користувачі можуть отримати доступ до системи з будь-якого пристрою, який має доступ до Інтернету та веб-браузера;
- веб-орієнтовані системи можуть бути використані на будь-якій операційній системі, яка підтримує веб-браузер;
- централізоване управління: дані та логіка додатку зберігаються на сервері, що дозволяє централізовано керувати доступом та оновленнями;
- гнучкість: веб-орієнтовані системи зазвичай мають гнучкість у розширенні функціональності та можуть швидко адаптуватися до змінних потреб користувачів [6].

Компоненти веб-орієнтованої інформаційної системи:

- клієнтська частина (Frontend): інтерфейс, який бачить користувач із веб-браузера. Включає в себе веб-сторінки, які відображають дані та забезпечують взаємодію з користувачем;

- серверна частина (Backend): сервер, на якому зберігаються дані та обробляються запити від клієнтської частини. Включає в себе базу даних, логіку додатку та інші серверні складові.

Приклади використання [6]:

- електронна комерція: наприклад, онлайн магазини, де користувачі можуть переглядати товари, робити замовлення та здійснювати оплату через Інтернет;

- системи управління відносинами з клієнтами (CRM): додатки, що дозволяють організаціям вести облік клієнтів, контактів та взаємодіяти з ними через веб-інтерфейс;

- системи управління виробництвом: додатки для управління процесами виробництва, які забезпечують доступ до даних та звітів через веб;

- портал надання публічної інформації: це веб-орієнтовані системи або портали, які надають публічні дані та інформацію для громадськості, можуть включати інформацію про урядові послуги, статистику, права та обов'язки громадян тощо.

- системи керування документами: вони дозволяють організаціям зберігати, організовувати та ділитися документами та інформацією всередині організації;

- електронні бібліотеки та архіви: ці системи надають доступ до цифрових ресурсів, таких як книги, наукові статті, архівні матеріали тощо;

- системи управління знаннями: вони дозволяють організаціям збирати, організовувати та ділитися знаннями та експертизою всередині організації [6].

1.2 Огляд і аналіз існуючих аналогів

Проведення огляду і аналізу існуючих аналогів перед розробкою нової системи має важливе значення, оскільки допомагає уникнути помилок, використовувати найкращі практики та успішні підходи, орієнтуватися на потреби користувачів і оптимізувати використання ресурсів, дозволяє ідентифікувати недоліки та проблеми, які можуть виникнути, зрозуміти очікування користувачів

та оптимізувати процес розробки, забезпечуючи успішне впровадження та використання нової системи.

Для огляду та аналізу існуючих рішень було взято за приклад такі веб-системи:

- веб-портал “ДІЯ”;
- портал державних послуг “IGOV”.

“Дія” (скорочено від «Держава і я») – це веб-портал і однойменний мобільний додаток, які є частиною проекту «держава в смартфоні» Міністерства цифрової трансформації України [2].

Метою проєкту "Дія" є перетворення процесу отримання державних послуг в Україні на швидкий та прозорий онлайн-процес. Замість того, щоб годинами стояти в чергах до офісів чиновників, громадяни можуть звертатися до "Дії" через Інтернет, щоб отримати різні документи, які необхідні для відкриття/закриття ФОП, оформлення допомоги з безробіття та інших адміністративних процедур. Платформа також забезпечує можливість зберігання електронних копій різних документів, таких як паспорти, студентські квитки та інші, які мають таку ж силу, як і оригінали. це робить процес взаємодії з державою більш зручним і ефективним для громадян.

Для доступу до особистого кабінету на порталі "Дія" потрібно підтвердити свою особу через процедуру верифікації. Це можна зробити за допомогою BankID, MobileID або особистого ключа – електронного цифрового підпису (ЕЦП) [2]. На рисунку 1.1 представлено форму для входу у портал “ДІЯ”. ЕЦП можна згенерувати в Приват24 і зберігати його на будь-якому носії [2].

Після успішної авторизації на порталі користувач потрапляє в меню, яке складається з чотирьох вкладок: "Зверніть увагу", "Мої документи", "Послуги" і "Відомості про мене". На вкладці "Зверніть увагу" зберігаються важливі повідомлення від порталу, якщо такі існують. Вкладка "Мої документи" відображає електронні документи користувача, а в решті вкладок – доступні відповідні послуги та відомості про користувача.

Авторизуватись з ID GOV UA

Державна система електронної ідентифікації, можна увійти за допомогою BankID або MobileID

Увійдіть за допомогою особистого ключа

Файловий ключ Апаратний ключ

Центр сертифікації ключів

Перетягніть сюди файл ключа або завантажте його зі свого носія

Пароль для ключа

Далі

Рисунок 1.1 – Авторизація та вхід у портал «Дія»

На рисунку 1.2 зображено початкове меню порталу. На вкладці "Послуги" доступний "архів" особистих дій на порталі "Дія", включаючи замовлені послуги, отримані документи і "чернетки" – послуги, замовлення яких не було завершено.

Вітаємо,

Зверніть увагу Мой документи Послуги Відомості про мене

Замовлені послуги Чернетки Отримані документи

Ви ще не замовляли послуги на сайті "Дія". Але ви можете переглянути популярні послуги.

Замовити ще

- Відкриття ФОП
- Закриття ФОП
- Внесення змін про ФОП
- Допомога по безробіттю

Рисунок 1.2 – Меню розділу «послуги»

У розділі "Відомості про мене" розміщені основні особисті дані користувача, які знаходяться у різних реєстрах, включаючи інформацію про його бізнес (якщо такий існує), а також про рухоме і нерухоме майно. Щоб їх переглянути, потрібно клікнути на відповідний розділ. Деякі з них поки не активні [2].

Але основний функціонал порталу – в меню «Послуги». Натиснувши на нього, можна побачити перелік послуг, який доступний як приватним особам, так і бізнесу, зображено на рисунку 1.3.

Довідки та витяги Послуги Довідка з Реєстру застрахованих осіб → Витяг про нормативну грошову оцінку → Витяг про земельну ділянку → Довідка про відсутність судимості → Від А до Я +	Транспорт Послуги Ліцензія на послуги з автоперевезень → Верифікація техпаспорта → Замовлення індивідуального номерного знака → Верифікація водійського посвідчення → Відновлення та обмін посвідчення водія → Призначення належного користувача → Від А до Я +
Навколишнє середовище Послуги Декларація про відходи → Дозвіл на водокористування → Від А до Я +	Земля, будівництво, нерухомість Послуги Відомості про власника землі → Витяг про нормативну грошову оцінку → Витяг про земельну ділянку → Інформація про осіб, що переглядали відомості про земельну ділянку → Від А до Я +
Безпека та правопорядок Послуги Перевірка автоперевізника → Довідка про відсутність судимості → Від А до Я +	Підприємництво Послуги Відкриття ФОП → Змінювання даних ФОП → Закриття ФОП → Підписання документів → Від А до Я +
Ліцензії та дозволи Послуги Ліцензії з протипожежної діяльності → Ліцензії на послуги з автоперевезень → Від А до Я +	Пенсії, пільги та допомога Послуги Довідка з Реєстру застрахованих осіб → Муниципальна няня → Допомога при народженні дитини → Допомога по безробіттю → Від А до Я +
Сім'я Послуги Я планую дитину Від А до Я +	

Рисунок 1.3 – Перелік послуг порталу «Дія»

Переваги користування порталом «Дія»: зрозумілий інтерфейс, швидкий доступ, безпека, наявність мобільного додатка, в якому зберігаються документи в електронному вигляді.

Недоліки: не всі опції працюють. При спробі замовити послугу або оформити документ на порталі "Дія", спершу з'являється діалогове вікно з поясненням про те, для чого потрібен документ і як це зробити. Однак, при натисканні кнопки "оформити", з'являється повідомлення, що "послуга перебуває на етапі реінжинірингу і незабаром буде доступна". Деякі функції розділів для громадян і

бізнесу дублюють одна одну. А розділи «Земля, будівництво, нерухомість» повністю ідентичні [2].

Більшість доступних послуг на порталі "Дія" наразі просто перенаправляють користувача на сайти відповідних державних установ, які надають ці послуги.

Веб-портал державних послуг iGov має дуже широкий спектр послуг, які надають державні органи України громадянам [3]. На рисунку 1.4 зображено його повне меню. У його реалізації беруть участь сотні і тисячі волонтерів не лише з України, а й з усього світу.

Для того, щоб скористатися послугами порталу, користувач повинен мати ЕЦП або оформлену банківську карту, а також обліковий запис в системі ідентифікації громадянина BankID [3].

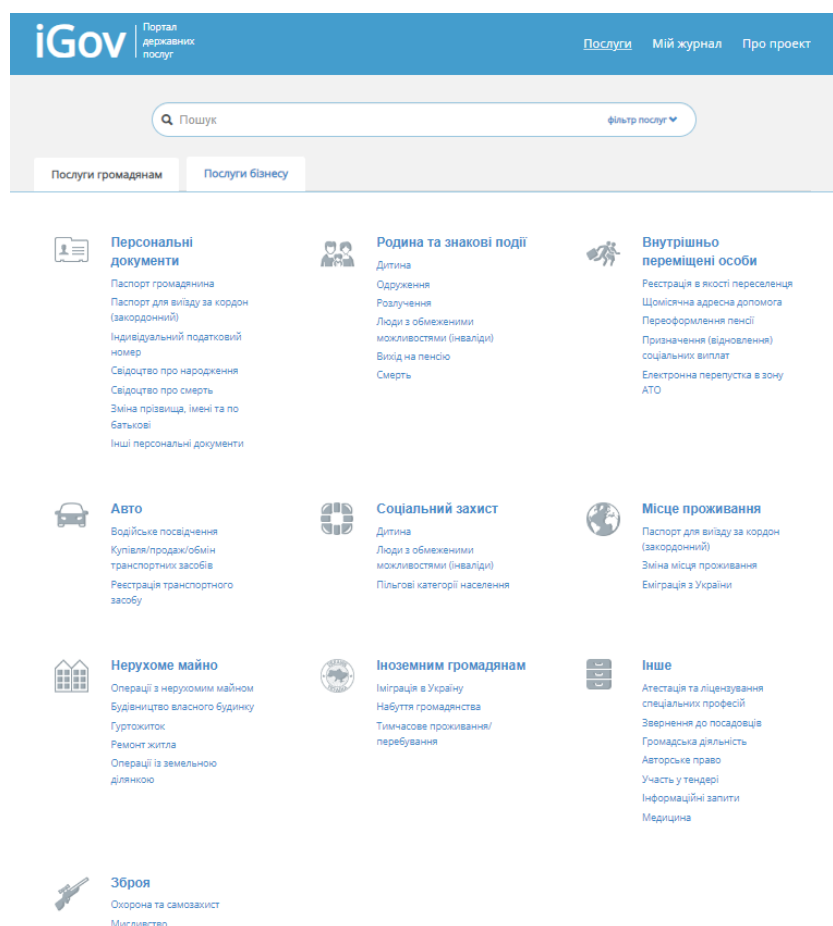


Рисунок 1.4 – Головна сторінка системи iGov

Для роботи portalу не потрібно встановлювати жодного ліцензованого програмного забезпечення на комп'ютери користувачів, оскільки це open source – відкрите програмне забезпечення. Це значно економить кошти місцевих бюджетів. Крім того, відповідальні особи можуть розглядати електронні заявки в автоматизованому режимі, незалежно від їх місцезнаходження [3].

Проте, цей портал недоступний в деяких областях країни. Також наповнення послуг для бізнесу є пустою. Живий пошук також не працює, логотип не перенаправляє на головну сторінку системи. На рисунку 1.5 зображено сторінку послуг для бізнесу portalу iGov. Кнопки перемикавання між послугами не працюють.

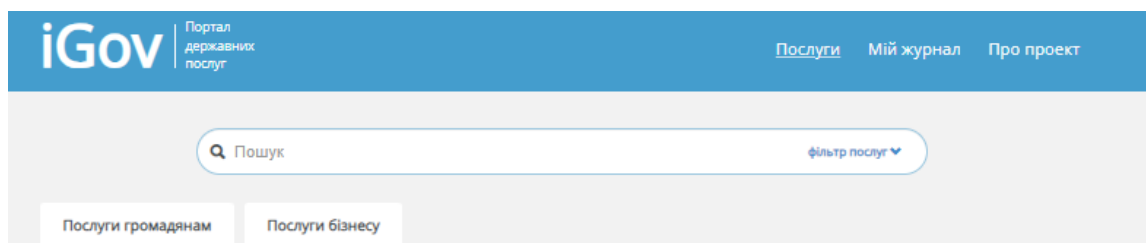


Рисунок 1.5 – Сторінка послуг для бізнесу portalу iGov

На рисунку 1.6 зображено пошук у системі, який не працює належним чином.

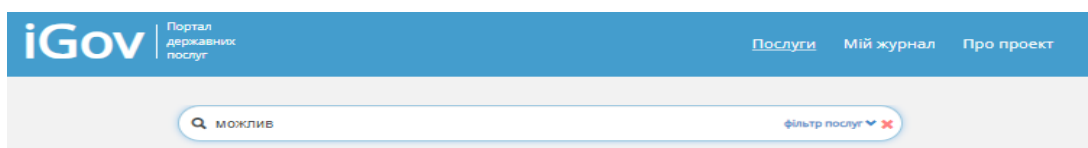


Рисунок 1.6 – Неробочий живий пошук

На рисунку 1.7 зображено фільтр послуг для регіонів, який не застосовується при використанні.

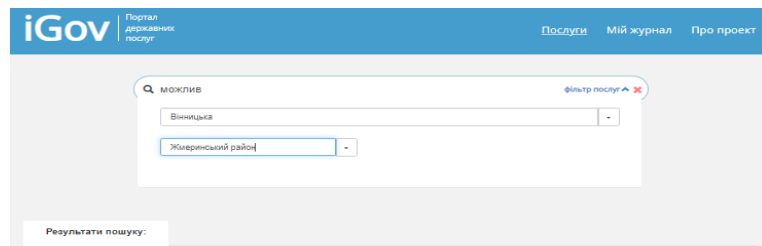


Рисунок 1.7 – Неробочий фільтр населених пунктів

Отже, портал "iGov" та система "Дія" є двома різними підходами до надання державних послуг онлайн. Ось деякі відмінності між цими системами:

Функціональність:

- "iGov" базується на відкритому програмному забезпеченні та залучає волонтерів з усього світу до розробки та підтримки системи [3];
- "Дія": управління цією системою здійснюється урядовими структурами, і вона розвивається у межах урядової стратегії та пріоритетів.

Доступність та охопленість:

- "iGov" може бути доступним у всіх регіонах, але, можливо, не у всіх випадках охоплює повний спектр державних послуг;
- "Дія" створена з метою надання широкого спектру державних послуг та є доступною для всіх громадян та бізнесу в Україні [2].

1.4 Постановка задачі проектування

Під час огляду існуючих аналогів було виявлено, що більшість систем управління адміністративними послугами не враховують специфіку періоду воєнного стану. Зазвичай ці системи спрямовані на нормальний режим роботи, і їхні функції та можливості обмежені стандартними ситуаціями. Це вказує на необхідність розробки спеціалізованої інформаційної системи, яка враховуватиме особливості адміністративного обслуговування в умовах воєнного стану та забезпечить необхідну функціональність та ефективність управління в цей період.

Згідно цього можна сформулювати вимоги до проєктованої системи надання

адміністративних послуг.

Функціональні вимоги:

- реєстрація та авторизація користувачів;
- можливість швидкого переходу між сторінками.

Вимоги до зовнішнього вигляду (UI/UX):

- сучасний та привабливий дизайн, який відповідає сучасним веб-стандартам;
- адаптивний дизайн, який забезпечить коректне відображення сайту на різних пристроях і екранних розмірах.
- інтуїтивний і простий у використанні інтерфейс користувача.

Безпека:

- захист особистих даних користувачів та даних сайту від несанкціонованого доступу;
- використання HTTPS-протоколу для забезпечення безпеки під час передачі даних.

Швидкодія та продуктивність:

- мінімізація часу завантаження сторінок для покращення користувацького досвіду;
- оптимізація роботи системи для підвищення продуктивності та масштабованості.

Наявність системи підтримки або контактної інформації для користувачів, які потребують допомоги чи консультації.

Ряд завдань, які буде виконувати система:

- дозвіл користувачам створювати облікові записи у системі;
- перевірка прав користувача на доступ до певних функцій або ресурсів системи;
- спілкування з конкретним юристом, або заповнення форми зворотнього зв'язку;
- можливість завантаження прикладу оформлення заяв.

2 ЗАГАЛЬНА СТРУКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ НАДАННЯ АДМІНІСТРАТИВНИХ ПОСЛУГ У ПЕРІОД ВОЄННОГО СТАНУ

2.1 Загальна структура проектованої системи

Основні функції веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану включатимуть:

а) надання адміністративних послуг для трьох категорій осіб:

- військовослужбовці;
- сім'ї військовослужбовців;
- внутрішньопереміщені особи;

б) перегляд списку документів, необхідних для соціальних виплат чи оформлення довідок:

- забезпечення доступу до переліку документів, які потрібно надати для отримання соціальних виплат;
- інформування про необхідність конкретних документів для виконання певних адміністративних процедур.

в) допомога в оформленні заявок та документів:

- надання підтримки користувачам у процесі заповнення заявок на адміністративні послуги;
- вказівки щодо правильного заповнення форм та необхідних кроків для завершення процедур [14].

Структура проектованої веб-орієнтованої інформаційної системи надання адміністративних послуг базується на моделі клієнт-сервер та використовує технології веб-розробки для створення зручного і ефективного інтерфейсу для користувачів.

Клієнт-серверна архітектура в мережі зображена на рисунку 2.1.

Клієнтська частина системи є ключовою складовою, оскільки саме через неї користувачі взаємодіють з інформаційною системою. Основними елементами клієнтської частини є користувацький інтерфейс та функціональність [6].

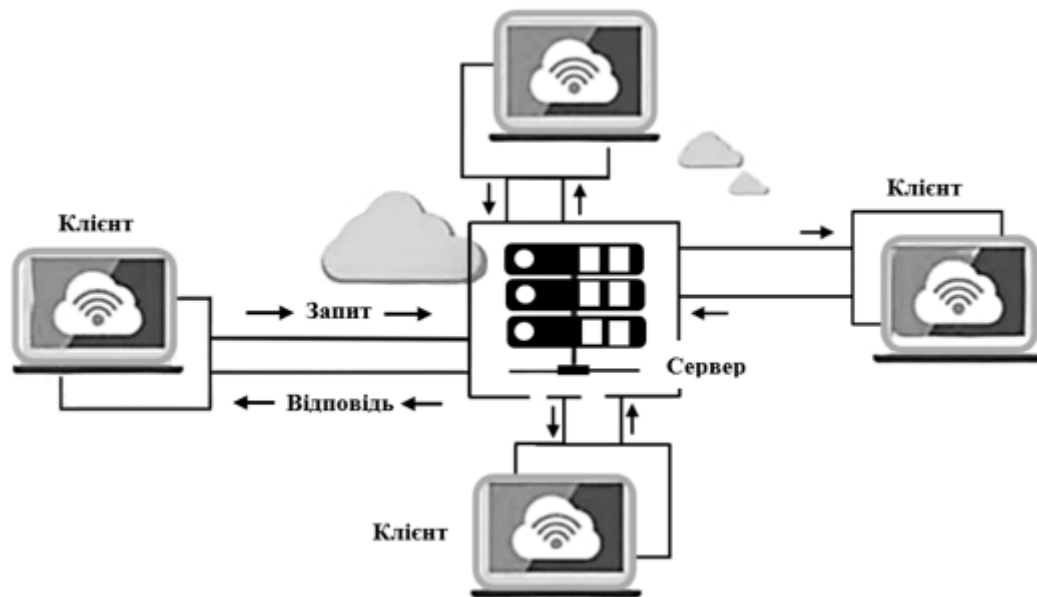


Рисунок 2.1 – Клієнт-серверна архітектура

Зовнішній вигляд системи буде розроблений з використанням HTML (дозволить структурувати вміст сторінки, визначати заголовки, абзаци, списки, таблиці, розміщувати зображення та створювати посилання на інші сторінки чи ресурси в Інтернеті), CSS (відповідатиме за те, як виглядатимуть різні елементи на веб-сторінках і інших документах, створених з використанням HTML, що забезпечить розробку гарного та зручного для користувача вигляду сторінки, і JavaScript (дасть можливість реагувати на події користувача, такі як кліки мишею, введення тексту, натискання клавіш тощо, і відповідно до цього виконувати певні дії).

Основні можливості для користувацького інтерфейсу веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану включатимуть:

- навігація та меню чітка та зручна навігація по системі за допомогою меню, яке міститиме наступні кнопки: “логотип”, “головна”, “про нас”, “новини” “послуги”, “зворотній зв'язок” та кнопки увійти для авторизації клієнта;

- адаптивний дизайн інтерфейс має бути адаптивним, щоб забезпечити зручний доступ до системи з різних пристроїв (комп'ютерів, планшетів,

смартфонів) і розмірів екранів;

- зручне відображення різноманітної інформації для користувачів, включаючи інформацію про підготовку документів, про систему, статті новин, зображення. Це дозволяє користувачам отримувати необхідну інформацію у зручному для сприйняття форматі.

Функціональність – це основна характеристика будь-якої програмної системи чи продукту, яка визначає його здатність виконувати певні завдання та надавати користувачам певний набір можливостей. Вибір правильної функціональності є ключовим етапом у розробці, оскільки від цього залежить ефективність, зручність та цінність програмного продукту для кінцевих користувачів. Функціональність включає в себе такі операції та можливості системи [6]:

- автентифікація дасть можливість користувачам входити до системи за допомогою унікального ідентифікатора (логіну, електронної пошти) та пароля. Після входу користувачам надається доступ до інформації в розділі меню “послуги”;

- форми та введення даних забезпечить можливість для користувачів заповнювати форми з введенням даних, таких як відправки повідомлення в розділі “зворотній зв’язок”, заповнення полів реєстрації та автентифікації. При цьому система може забезпечувати перевірку правильності введених даних та виводити відповідні повідомлення про помилки.

Серверна частина веб-орієнтованої інформаційної системи надання адміністративних послуг грає важливу роль у забезпеченні функціональності та надійності системи. Основні можливості серверної частини включають:

- серверна частина забезпечує зберігання та управління даними, що включає в себе базу даних для зберігання інформації про користувачів, їхні дії та дані, необхідні для роботи системи;

- сервер приймає запити від клієнтської частини (наприклад, від форм введення даних або кнопок користувача), обробляє їх та виконує необхідні

операції, щоб отримати або оновити дані у базі даних;

- серверна частина відповідає за забезпечення безпеки системи, включаючи автентифікацію користувачів, перевірку доступу до ресурсів та захист від несанкціонованого доступу.

2.2 Алгоритмічне забезпечення системи

Написання алгоритмів є основною частиною для будь-якої розробки. Вони визначають, як система обробляє дані, приймає рішення та взаємодіє з користувачами. Побудова алгоритмів включає в себе декомпозицію проблеми на підзадачі, розробку логіки для кожної з них та забезпечення інтеграції в єдину структуру [7].

Структура алгоритмів може варіюватися від простих послідовностей дій до складних систем, що включають умови, цикли та рекурсії. Для забезпечення високої якості програмного забезпечення алгоритми повинні бути ретельно протестовані та оптимізовані. Важливість алгоритмів полягає в тому, що вони не лише впливають на продуктивність та стабільність системи, але й визначають її здатність реагувати на непередбачувані ситуації, що особливо актуально у процесі розробки веб орієнтованої інформаційної системи надання адміністративних послуг [8].

Алгоритм авторизації користувача представлений на рисунку 2.2.

Детальний опис етапів алгоритму взаємодії користувача з веб-орієнтованою інформаційною системою наведений нижче.

Початковий процес авторизації: користувач починає процес авторизації, вводячи свій Email та пароль у відповідні поля форми.

Перевірка заповнення полів:

- заповнення форми (користувач заповнює поля форми Email та паролем);
- перевірка заповнення (система перевіряє наявність введених даних у всіх необхідних полях форми, якщо поля не заповнені, система відправляє

повідомлення "Поля не заповнено" і повертає користувача до етапу введення даних.

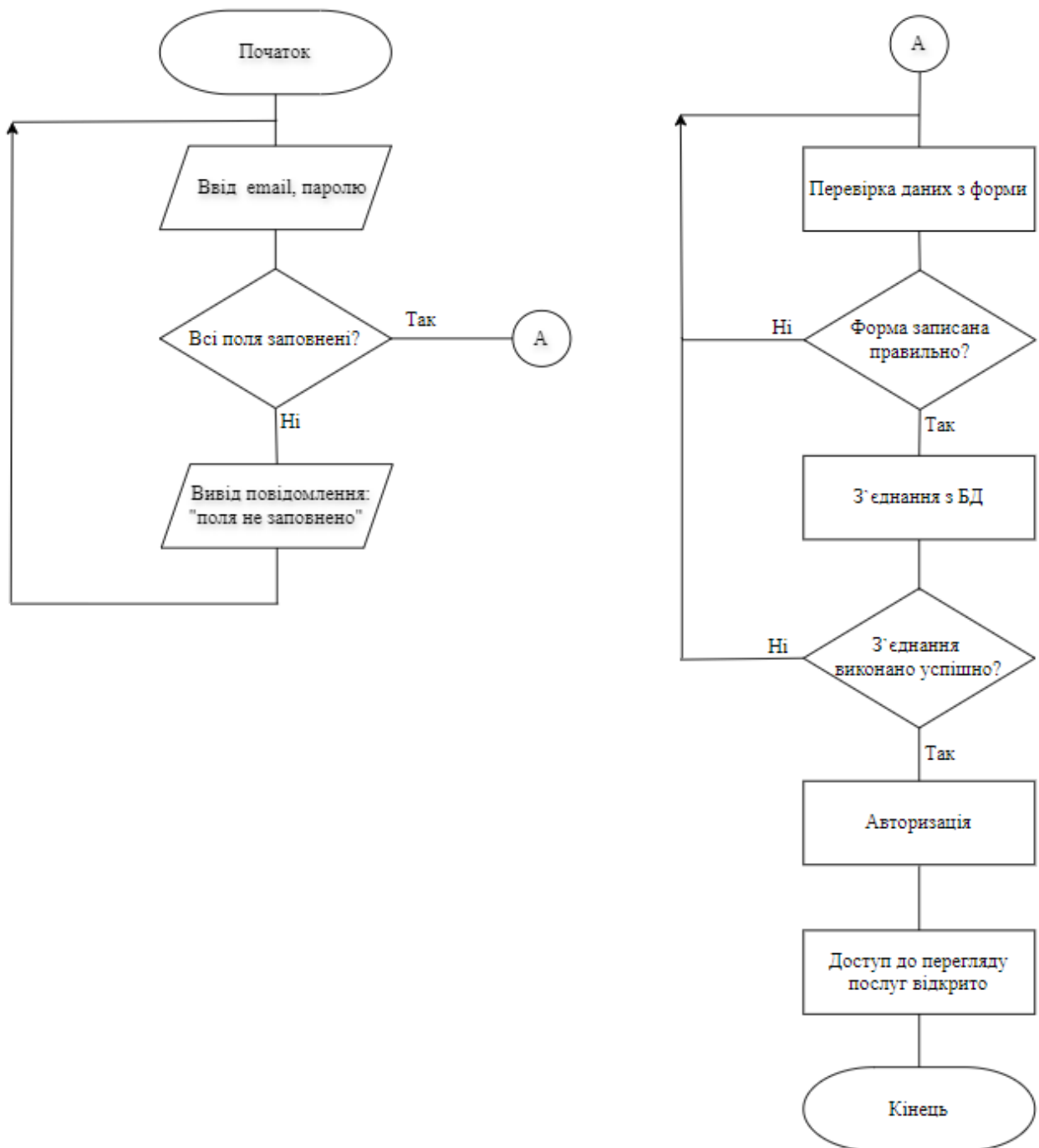


Рисунок 2.2 – Схема алгоритму авторизації користувача

Перевірка коректності введених даних: валідація форми (дані, введені у форму, проходять перевірку на коректність, якщо форма заповнена неправильно, процес повертається до введення даних з відповідним повідомленням про помилку).

З'єднання з базою даних:

- відправка форми (після успішного заповнення і валідації форми, дані відправляються для перевірки);
- перевірка з'єднання (система намагається встановити з'єднання з базою даних, якщо з'єднання не вдалося, система повторно перевіряє дані з форми і намагається встановити з'єднання знову).

Перевірка користувача у базі даних: ідентифікація користувача (після успішного встановлення з'єднання, система перевіряє, чи існує користувач з введеними Email та паролем у базі даних, якщо користувача не існує, процес повертається до введення даних з відповідним повідомленням).

Якщо відбулося співпадіння введених даних з даними у базі, користувач авторизується і отримує доступ до послуг.

Алгоритм взаємодії користувача з веб-орієнтованою інформаційною системою представлений на рисунку 2.3.

Детальний опис етапів алгоритму взаємодії користувача з веб-орієнтованою інформаційною системою наведений нижче.

Початкові дії:

- ініціалізація системи (система починає свою роботу, користувач готується до її запуску);
- запуск локального хосту (користувач запускає локальний хост, де розміщена система);
- відкриття браузера (після запуску локального хосту відкривається браузер).

Перевірка підключення до Інтернету: система перевіряє наявність підключення до Інтернету. Якщо підключення є, процес продовжується. Якщо підключення немає, з'являється повідомлення про помилку, і користувач повинен вжити заходів для вирішення проблеми.

Взаємодія з системою:

- перехід на головну сторінку (користувач переходить за відповідним

посиланням, щоб потрапити на головну сторінку системи; система завантажується);

– перевірка підключення до Інтернету (система знову перевіряє наявність підключення до мережі; якщо підключення є, процес продовжується, якщо підключення немає, то з'являється повідомлення про помилку, і користувач повинен вжити заходів для вирішення проблеми).

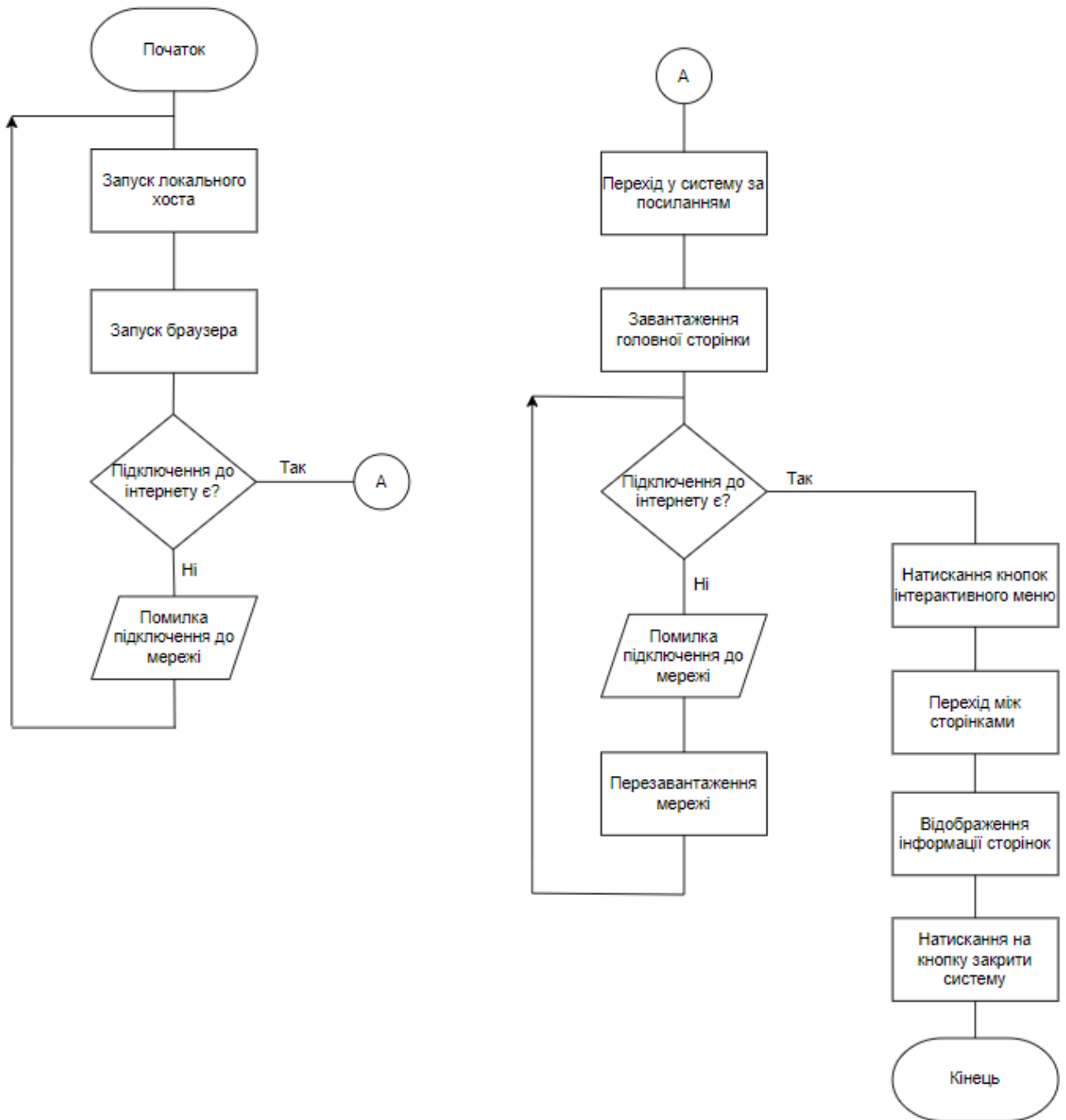


Рисунок 2.3 – Схема алгоритму взаємодії користувача з веб-орієнтованою інформаційною системою

Навігація по системі:

- інтерактивне меню (користувач використовує інтерактивне меню для навігації);
- перехід між сторінками (користувач може переходити між різними сторінками системи, на яких відображається необхідна інформація).

Завершення роботи:

- закриття системи (користувач натискає кнопку для закриття системи);
- завершення процесу (система закривається, процес завершено).

Узагальнюючи представлений алгоритм авторизації користувача та алгоритм взаємодії з веб-орієнтованою інформаційною системою, можна зазначити, що алгоритм авторизації гарантує правильну перевірку даних користувачів та безпечний доступ до послуг, а алгоритм взаємодії сприяє зручній навігації та безперебійній роботі системи за умов, що є підключення до інтернету [9].

2.3 Інформаційне забезпечення веб-орієнтованої системи

База даних для веб-орієнтованої інформаційної системи надання адміністративних послуг розроблена з використанням Entity Framework Core, що забезпечує об'єктно-реляційне відображення (ORM) між моделями даних та реляційною базою даних [10]. Вона зберігає дані про користувачів, включаючи їх ім'я, email та пароль, а також первинний ключ, який забезпечує унікальність кожного запису.

Основна модель даних для користувача містить наступні поля:

- Ім'я (Name): текстове поле, що зберігає ім'я користувача;
- Email (Email): текстове поле, що зберігає електронну пошту користувача. поле є унікальним для кожного користувача;
- пароль (Password): текстове поле, що зберігає хешований пароль користувача;
- первинний ключ (Id): унікальний ідентифікатор користувача, який

автоматично генерується базою даних.

Логіка бази даних, включаючи створення, читання, оновлення та видалення (CRUD) операцій, описана на мові програмування C#. Використовуючи Entity Framework Core, розробники можуть працювати з базою даних через об'єкти C#, що робить код більш зрозумілим і легким для підтримки.

Entity Framework Core підтримує різні підходи до роботи з базами даних, включаючи Code-First, Database-First та Model-First.

У системі використовується підхід Code-First, що дозволяє розробникам спочатку створювати моделі даних у коді, а потім автоматично генерувати базу даних на основі цих моделей. Цей підхід особливо зручний для проєктів, де структура бази даних може часто змінюватися під час розробки. Крім того, він дозволяє використовувати міграції для поступового оновлення бази даних, що є важливим аспектом для забезпечення безперервної роботи системи та збереження даних [10].

Моделі даних у системі описані з використанням C# класів. Наприклад, клас “User” містить властивості для збереження імені користувача, його електронної пошти, хешованого паролю та унікального ідентифікатора. Кожна властивість має відповідні атрибути для налаштування обмежень і правил валідації. Наприклад, електронна пошта повинна бути унікальною, а пароль - захищеним від простого зчитування. Для цього використовуються вбудовані механізми Entity Framework Core, такі як атрибути “Key”, “Required”, “MaxLength” [11].

Обробка даних користувачів включає кілька важливих етапів, таких як валідація введених даних, їх нормалізація та хешування паролів перед збереженням у базі даних.

Для хешування паролів використовується сучасний алгоритм bcrypt, який забезпечує високий рівень захисту даних. Це важливо для запобігання атакам типу "brute force" та захисту конфіденційної інформації користувачів.

Валідація даних здійснюється як на стороні клієнта, так і на стороні сервера. Це дозволяє забезпечити коректність введених даних перед їх обробкою і збереженням.

На стороні сервера, валідація виконується з використанням атрибутів валідації, що визначаються безпосередньо в моделях даних. Наприклад, електронна пошта повинна відповідати формату електронної адреси, а пароль повинен мати мінімальну довжину.

Система також підтримує роботу з ролями користувачів, що дозволяє розмежовувати доступ до різних функціональних можливостей в залежності від ролі. Це досягається через додаткову модель “Role” та таблицю зв'язків між користувачами і ролями. Така реалізація допомагає підтримувати високий рівень безпеки та дозволяє гнучко управляти правами доступу користувачів.

Фрагмент коду, який демонструє визначення класу “Role” на мові C#, представлено на рисунку 2.4.

```
public class Role
{
    [Key]
    public int RoleId { get; set; }

    [Required]
    [MaxLength(50)]
    public string RoleName { get; set; }

    public ICollection<User> Users { get; set; }
}
```

Рисунок 2.4 – Визначення класу “Role” на мові C#

Розробка та управління базою даних здійснюється у середовищі Visual Studio, яке надає всі необхідні інструменти для ефективної роботи з Entity Framework Core. Visual Studio дозволяє швидко генерувати міграції для оновлення схеми бази даних, а також зручно працювати з кодом на C#.

Рисунок 2.5 демонструє таблицю User з кількома записами користувачів, кожен з яких має унікальний ідентифікатор, ім'я, електронну пошту та пароль.

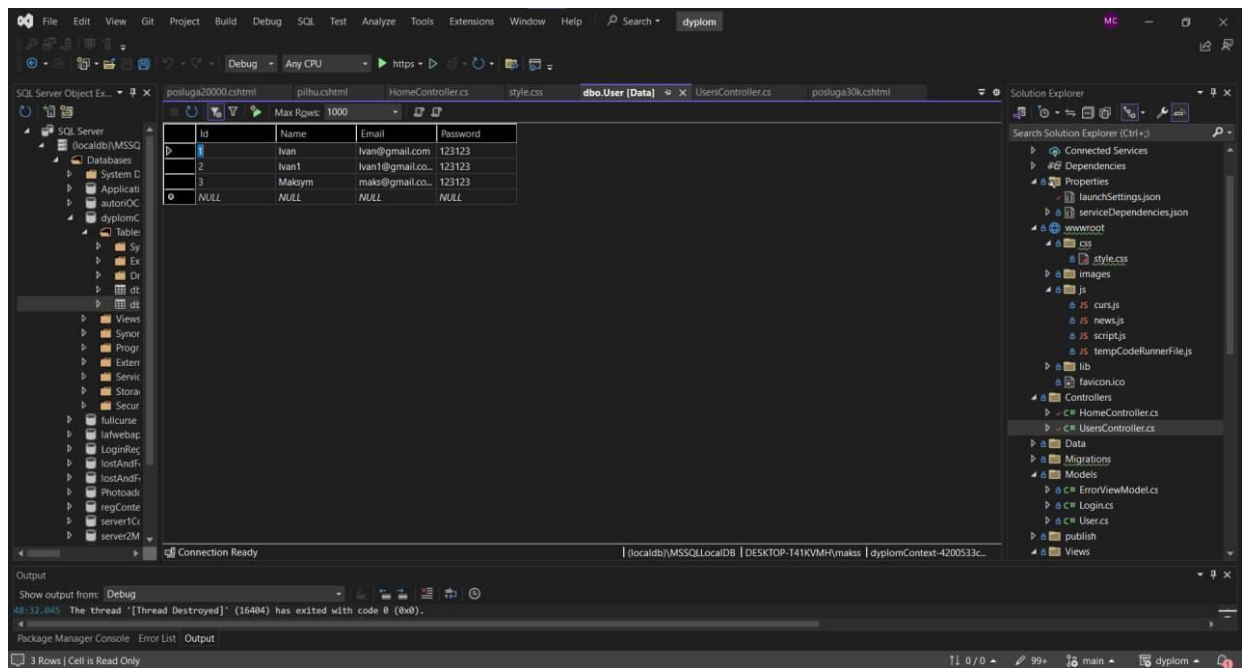


Рисунок 2.5 – Копія екрану середовища Visual Studio із записом користувачів в таблицю бази даних

Ця таблиця є частиною бази даних і використовується для управління інформацією про користувачів у системі. Поле Password зберігає хешовані паролі для забезпечення безпеки даних користувачів.

З РЕАЛІЗАЦІЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ НАДАННЯ АДМІНІСТРАТИВНИХ ПОСЛУГ У ПЕРІОД ВОЄННОГО СТАНУ СТАНУ

3.1 Програмна реалізація

Під час програмної реалізації системи було використано набір інструментів, який включав у себе HTML, CSS і JavaScript для фронтенду, а також мову програмування C# для розробки бекенду.

Першим етапом реалізації було створення інтерфейсу користувача, для цього було використано інструмент Figma для розробки дизайну і прототипування, щоб створити зручний і легкий у використанні інтерфейс. Figma має потужні засоби для створення прототипів, які дозволяють створювати інтерактивні макети з анімацією та переходами між сторінками, що демонструє, як буде виглядати кінцевий продукт та взаємодіяти з ним. Дана платформа дозволяє створювати компоненти та бібліотеки стилів, що спрощує процес розробки, дозволяє зберігати стандартні елементи дизайну та швидко їх застосувати [15].

Найбільшим плюсом вибору для використання платформи Figma для роботи з дизайном було те, що вона має широкий вибір плагінів, які доповнюють його функціональність, включаючи плагіни для генерації коду, автоматизації завдань, та інші, також містить бібліотеки ресурсів, які містять іконки, зображення, шрифти, які було застосовано до системи.

Після завершення дизайну наступним етапом була верстка сторінок. Використовуючи HTML та CSS, був перетворений дизайн інтерфейсу з Figma на функціональні веб-сторінки. Під час розробки фронтенду, було враховано не лише зовнішній вигляд і зручність користування, а й адаптивність до різних пристроїв. Використання технік інтерфейсного дизайну, таких як резиновий дизайн і медіа-запити, дозволило створити інтерфейс, який коректно відображається на різних пристроях, включаючи комп'ютери, планшети та мобільні телефони [16].

Функціонал сторінок забезпечила мова програмування JavaScript. За допомогою вибраної мови було додано різноманітні ефекти та взаємодію з користувачем. Наприклад: анімації при кліках на кнопки або при наведення

курсору на елементи, приховане меню для кращої адаптації, що зробило інтерфейс більш привабливим та цікавим. Написання скриптів валідації форм, які перевіряли правильність введених даних у формах перед їх відправленням. Наприклад, була реалізована перевірка правильності введеної електронної адреси або відповідності пароля заданим вимогам. Це дозволило зменшити кількість помилок та забезпечило коректну роботу системи.

За допомогою JavaScript була забезпечена взаємодія з зовнішніми API, такими як OpenWeatherMap для отримання прогнозу погоди та Monobank для отримання курсу валют. Це дозволило системі автоматично оновлювати інформацію про погоду та курс валют без необхідності ручного оновлення [17].

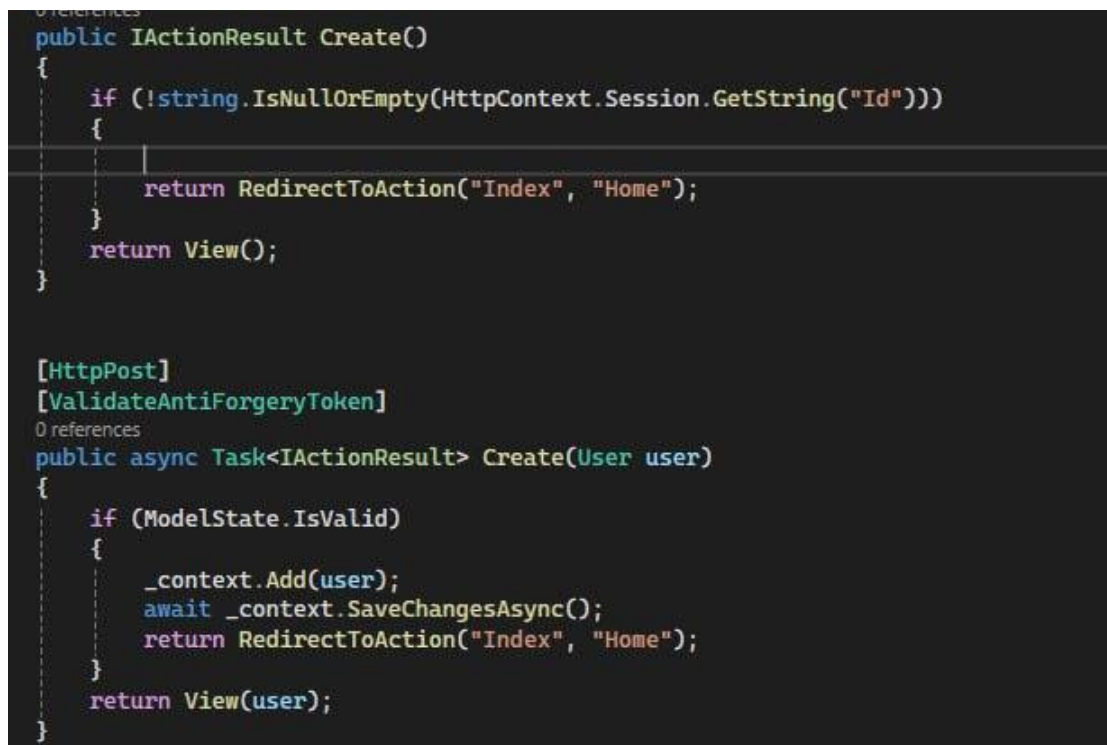
Для розробки бекенду використано мову програмування C# та середовища розробки Visual Studio. C# – це мова програмування з потужними функціональними можливостями, що дозволяють створювати складні, масштабовані та ефективні бекенд-додатки. Дана мова програмування має велику кількість вбудованих бібліотек та фреймворків, що полегшують розробку різноманітних функціональних можливостей, таких як обробка даних, мережева взаємодія, безпека, робота з базами даних тощо [11].

C# – це мова, яка підтримується .NET платформою і дозволяє розробникам створювати програми, які можуть працювати в різних операційних системах, включаючи Windows, Linux і macOS. Це дозволяє створювати міжплатформні програми, які працюють у різних середовищах [11].

Visual Studio – це одне з найпопулярніших та потужних інтегрованих середовищ розробки (IDE), яке дозволяє розробникам створювати програмне забезпечення на різних мовах програмування включає в себе багатий набір корисних інструментів. Дане середовище має автодоповнення коду, можливість відлагодження в реальному часі, інтеграцію з системами контролю версій та інші корисні функції, які значно спрощують процес розробки. Крім того, Visual Studio надає інструменти для виявлення та виправлення помилок в коді, що допомагає забезпечити високу якість програмного забезпечення [12].

Фрагмент коду розробки логіки перевірки ведених даних користувачем

зображено на копії екрану, рисунок 3.1.



```
0 references
public IActionResult Create()
{
    if (!string.IsNullOrEmpty(HttpContext.Session.GetString("Id")))
    {
        return RedirectToAction("Index", "Home");
    }
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create(User user)
{
    if (ModelState.IsValid)
    {
        _context.Add(user);
        await _context.SaveChangesAsync();
        return RedirectToAction("Index", "Home");
    }
    return View(user);
}
```

Рисунок 3.1 – Копія екрану з кодом перевірки введених даних користувачем

За допомогою мови програмування C# та середовища Visual Studio було реалізовано логіку обробки відправки даних з форм до серверної частини системи. Для цього використовувалися методи GET та POST. Наприклад, при відправці форми на сервер за допомогою JavaScript, дані були передані на серверну сторону з використанням методу POST. За допомогою C# була написана програмна логіка, яка приймала ці дані, проводила їх перевірку на коректність, обробляла та зберігала в базі даних. Крім того, метод GET використовувався для отримання даних з сервера, наприклад, при завантаженні сторінки або виконанні певних дій користувачем. Після успішної обробки даних на сервері за допомогою методів GET та POST, відповідні дані були відправлені назад до клієнтського браузера для відображення на сторінці користувача.

На етапі версток сторінок до системи було використано середовище Visual Studio Code яке має деяку спільність з Visual Studio у певних аспектах, що дозволяє

розробникам працювати в одному середовищі та використовувати різні інструменти із Visual Studio.

Visual Studio Code відзначається простим та інтуїтивно зрозумілим інтерфейсом, який дозволяє швидко розпочати роботу без додаткового налаштування. Проте, він також надає широкі можливості для налаштування та розширення за допомогою різноманітних плагінів і розширень.

Велика кількість розширень та плагінів для VS Code дозволяє розробникам налаштовувати редактор під свої потреби та використовувати його для роботи з різними мовами програмування та технологіями [13].

VS Code має багато корисних функцій, таких як автодоповнення коду, відлагодження в реальному часі, інтеграція з системами контролю версій, підтримка роботи з Docker, вбудована консоль та інші, що значно полегшують процес розробки [13].

Копія екрану, що представлена на рисунку 3.2, демонструє фрагмент коду розробки головної сторінки у середовищі Visual Studio Code.

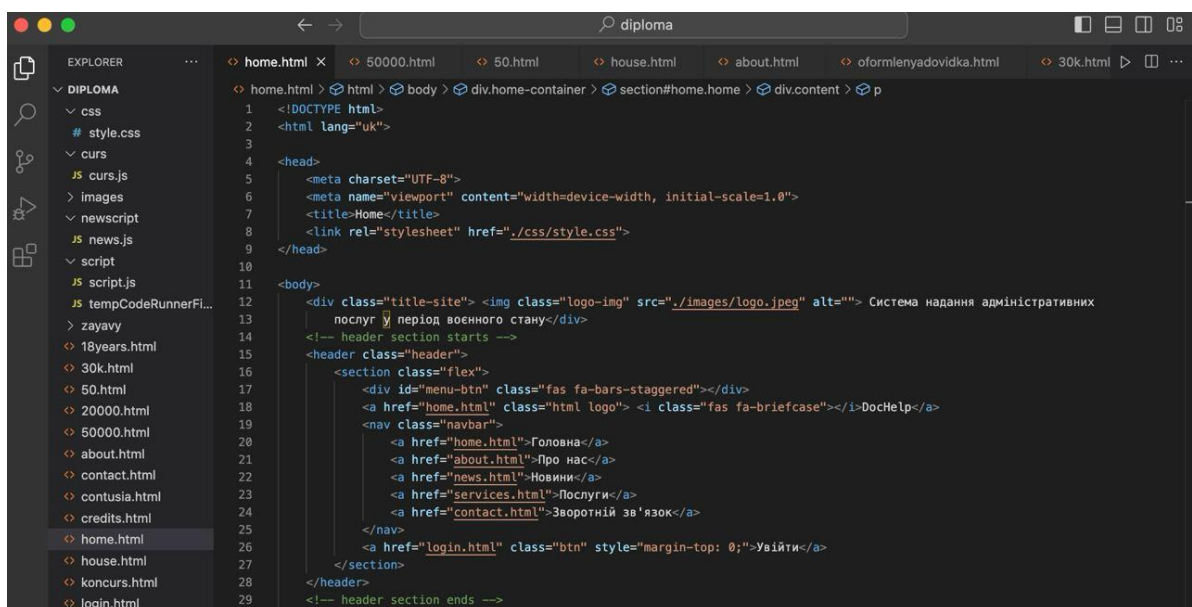


Рисунок 3.2 – Розробка головної сторінки у середовищі Visual Studio Code

Під час розробки бекенду було використано фреймворк ASP.NET Core, що

надавав широкі можливості з високою продуктивністю та масштабованістю. Використання ASP.NET Core забезпечило швидку та ефективну обробку запитів користувачів, за допомогою чого було реалізовано різноманітні функціональні можливості системи.

Фреймворк ASP.NET Core побудовано з урахуванням переносимості і може запускатися на різних платформах, включаючи Windows, macOS та Linux. Це дає розробникам гнучкість вибору середовища для розгортання своїх додатків. Даний фреймворк має вбудовані механізми безпеки, такі як захист від переповнення буфера, автентифікація та авторизація на рівні ресурсів, підтримка HTTPS та багато іншого. Це дозволяє розробникам забезпечувати безпеку своїх додатків з мінімальними зусиллями [19].

ASP.NET Core підтримує різні типи розширень, включаючи middleware, сервіси, фільтри та інші. Це дозволяє розробникам розширювати функціональність своїх додатків за допомогою сторонніх компонентів або створювати власні [19].

Програмний код верстки сторінок веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану представлений у додадку А.

3.2 Інтерфейс користувача

Інтерфейс користувача — це спосіб, за допомогою якого користувачі взаємодіють з програмним забезпеченням, веб-сайтами або іншими електронними пристроями. Він включає в себе всі елементи та екрани, з якими користувач може взаємодіяти, такі як кнопки, меню, поля введення, ілюстрації та текстові блоки. До прикладу на рисунку 3.3 зображено головну сторінку веб-орієнтованої інформаційної системи надання адміністративних послуг, що розроблена, де вгорі розміщене меню, з яким може взаємодіяти користувач [21]. Меню включає: «Головна», «Про нас», «Новини», «Послуги», «Зворотній зв'язок».

Дизайн інтерфейсу користувача сприяє зручності та ефективності взаємодії користувача з продуктом. Це включає в себе правильне розміщення елементів,

читабельність тексту, доступність та зручність використання [20].



Рисунок 3.3 – Головна сторінка системи

Для забезпечення безпечного та персоналізованого доступу до адміністративних послуг у період воєнного стану веб-орієнтована інформаційна система включає механізм авторизації користувачів. На рисунку 3.4 зображено екран авторизації користувача, який є важливим елементом системи безпеки та доступу.

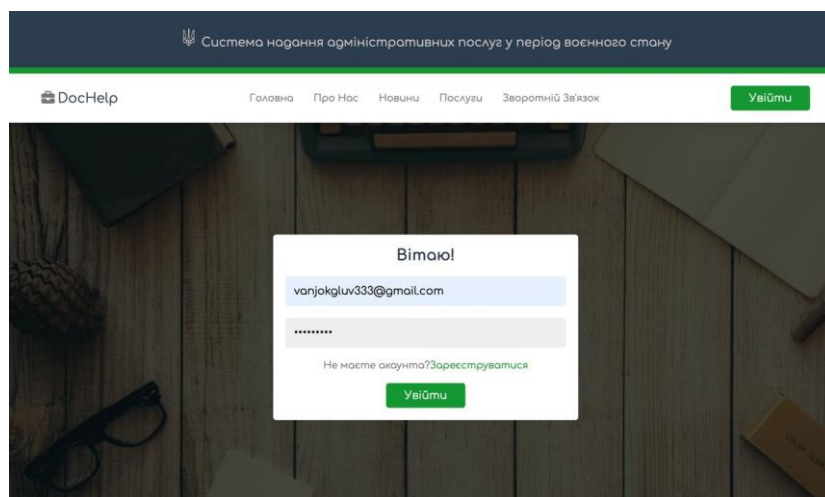


Рисунок 3.4 – Меню авторизації

На рисунку 3.5 представлено реєстраційну форму веб-орієнтованої інформаційної системи надання адміністративних послуг під час воєнного стану.

Реєстрація в системі є надзвичайно важливою, оскільки незареєстровані користувачі не отримують доступу до переліку адміністративних послуг, які надає система.

Рисунок 3.5 – Меню реєстрації

На рисунку 3.6 представлено фрагмент сторінки меню переліку послуг розробленої веб-орієнтованої інформаційної системи, оскільки перелік послуг є дуже великим.

Перелік послуг	
Військовослужбовцям ↓	
(50 000 грн) Призначення одноразової матеріальної допомоги на забезпечення спорядженням та технічними засобами у зв'язку з безпосередньою участю у бойових діях для подання заяви при участі у бойових діях	>
Призначення одноразової матеріальної допомоги на забезпечення спорядженням та технічними засобами у зв'язку з безпосередньою участю у бойових діях (доплата 30 000 грн)	>
(20 000 грн) Призначення одноразової матеріальної допомоги на забезпечення спорядженням та технічними засобами у зв'язку з безпосередньою участю у бойових діях для подання заяви військовослужбовцями	>
Надання одноразової матеріальної допомоги у зв'язку із народженням дитини (для військовослужбовців)	>
Призначення одноразової матеріальної допомоги одному з батьків, з нагоди дня народження дитини до 18 років (включно), батьки яких призвані на військову службу або беруть участь у заходах, необхідних для забезпечення оборони України	>
Призначення одноразової матеріальної допомоги особам, які отримали поранення (контузію, каліцтво) під час безпосередньої участі у заходах, необхідних для забезпечення оборони України	>
Надання одноразової матеріальної допомоги на оплату комунальних послуг військовим, які не мають статусу УБД	>
Надання на оплату комунальних послуг для учасників бойових дій та їх сімей з міського бюджету	>
Надання одноразової матеріальної допомоги Захисникам і Захисницям України для відновлення та реабілітації	>
Сім'ям військовослужбовців ↓	
Призначення одноразової грошової допомоги ("ОГД") сім'ям загиблих військових у розмірі 15 млн гривень.	>
Призначення 50% знижки по платі за користування комунальними послугами (газом, електроенергією та іншими послугами) та скрапленням баланним газом	>
Вступ поза конкурсом до державних та комунальних закладів вищої та фахової передвищої освіти на спеціальності, підготовка за якими здійснюється за рахунок коштів відповідно державного та місцевих бюджетів.	>
Призначення пенсії по втраті годувальника	>

Рисунок 3.6 – Меню переліку послуг системи

Ця сторінка містить структурований перелік доступних послуг, спрямованих на підтримку військовослужбовців, їх сімей та внутрішньопереміщених осіб, зокрема інформацію про матеріальну допомогу, соціальні виплати та пільги.

На рисунку 3.7 представлено сторінку з детальною інформацією про конкретну адміністративну послугу.

Ця сторінка містить повний опис послуги, включаючи умови отримання, необхідні документи та процедуру подання заявки, що значно полегшує користувачам доступ до важливої інформації та допомагає у швидкому оформленні необхідної допомоги.

Система надання адміністративних послуг у період воєнного стану

DocHelp Головна Про Нас Новини Послуги Зворотній Зв'язок Увійти

(50 000 грн) Призначення одноразової матеріальної допомоги на забезпечення спорядженням та технічними засобами у зв'язку з безпосередньою участю у бойових діях для подання заяви при участі у бойових діях

Право на отримання одноразової матеріальної допомоги на забезпечення спорядженням та технічними засобами мають особи, які: у 2024 році беруть безпосередню участь у бойових діях (діях зі знищення живої сили, бойової техніки і військових споруд противника, заволодіння територією, яку він обіймав, надання протидії наступу противника, відбиття його ударів й утримання захопленої своїми військами території), на момент подання заяви на таку допомогу перебувають на службі, що підтверджується окремим документом; у 2024 році беруть безпосередню участь у забезпеченні здійснення заходів з національної безпеки і оборони України відсічі і стримування збройної агресії російської федерації проти України, на момент подання заяви на таку допомогу перебувають на службі, що підтверджується відповідним документом;

Орган, що надає послугу: Структурні підрозділи з питань соціального захисту населення районних, районних у м. Києві та Севастополі держадміністрацій, виконавчих органів міських, районних у містах (у разі їх створення) рад

Місце подання документів та отримання результату: Центр надання адміністративних послуг

Перелік необхідних документів, вимоги до них та спосіб подання

Для оформлення одноразової допомоги заявник разом з заявою подає:

- паспорт одного з батьків чи документ, який посвідчує особу;
- код ІПН;
- довідка про реєстрацію місця проживання (якщо у вас ID-картка);
- довідка про перебування на обліку з приводу вагітності в закладах охорони здоров'я, які розташовані на території Тернопільської міської громади;
- витяг з реєстру територіальної громади про реєстрацію місця проживання особи (не менше, ніж за 6 місяців до дня народження дитини);
- свідоцтво про народження дитини;
- копія документа про безпосередню участь батька/матері новонародженої дитини у бойових діях (при наявності);
- реквізити рахунку (матері дитини), відкритого в установі банку

Перелік нормативно-правових документів, що регулюють надання послуги: Послуга надається відповідно до Закону України «Про затвердження Указу Президента України «Про введення воєнного стану в Україні», Закону України «Про правовий режим воєнного стану», Закону України «Про оборону України», Закону України «Про мобілізаційну підготовку та мобілізацію», Закону України «Про військовий обов'язок і військову службу», постанови Кабінету Міністрів України від 11 березня 2022 року №252 «Деякі питання формування та виконання місцевих бюджетів у період воєнного стану», розпорядження Кабінету Міністрів України від 24.02.2022 року №181-р «Питання запровадження та забезпечення здійснення заходів правового режиму воєнного стану в Україні», рішень органів місцевого самоврядування.

Документи, які отримує особа за результатами розгляду справи : Рішення органу місцевого самоврядування (у випадку передбачення даної послуги)

Рисунок 3.7 – Меню обраної послуги користувачем

Сторінка новин зображена на рисунку 3.8.

Наявність контактної інформації та форми зворотного зв'язку забезпечує зручний та швидкий спосіб комунікації з адміністраторами системи.



Рисунок 3.8 – Сторінка новин

Сторінка зворотного зв'язку (рисунок 3.9) є важливою частиною системи, оскільки дозволяє користувачам отримати необхідну підтримку та консультації.

Рисунок 3.9 – Послуга зворотнього зв'язку

Привабливий інтерфейс може покращити загальне враження від продукту та привернути інтерес користувачів до нього. Гармонійний колірна гама, якісні графічні елементи та зручне розташування елементів можуть зробити взаємодію з програмою більш приємною [22].

3.3 Тестування інформаційної системи

Тестування верстки системи є невід'ємною частиною процесу розробки веб-додатків і відіграє критичну роль у забезпеченні якості та коректності відображення вмісту для кінцевих користувачів. Важливість цього етапу впливає з різноманітних факторів.

Обґрунтування важливості тестування верстки системи:

- забезпечення сумісності: різні браузерери та пристрої можуть інтерпретувати верстку по-різному. Тестування верстки допомагає переконатися, що веб-сторінки відображаються правильно на різних браузерах (Chrome, Firefox, Safari, Edge тощо) та пристроях (комп'ютерах, планшетах, мобільних телефонах) [23];
- забезпечення коректності відображення: правильно виконана верстка гарантує, що контент сторінок відображається коректно і зручно для користувача. Тестування верстки допомагає виявити і виправити будь-які проблеми з розміщенням елементів, шрифтами, колірною схемою тощо;
- забезпечення відповідності дизайну: верстка повинна точно відображати дизайн, розроблений в середовищі Figma, що був погоджений із замовником. Тестування верстки допомагає переконатися, що реалізація веб-сторінок відповідає дизайнерським вимогам і очікуванням замовника;
- підтримка різних роздільних здатностей: верстка повинна бути адаптивною і підтримувати різні роздільні здатності екранів. Тестування верстки допомагає перевірити, як сторінки виглядають на різних екранах і роздільних здатностях, а також виявити та виправити проблеми з адаптацією [23].

Тест-кейс є важливим інструментом у процесі тестування програмного забезпечення, включаючи веб-додатки. Його основною метою є систематизація та документування специфікацій тестування, що дозволяє тестувальникам однозначно зрозуміти, як проводити тести, як очікувати відповіді від системи та як оцінювати їх результати. Також тест-кейс слугує як засіб контролю якості: він дозволяє виявляти, документувати та виправляти помилки, а також перевіряти, чи

відповідає програмне забезпечення вимогам та очікуванням користувачів [25].

У таблиці тест-кейсу зазвичай вказуються такі дані, як опис тесту, кроки, що потрібно виконати, очікуваний результат та фактичний результат. Опис тест-кейсу може містити інформацію про те, що саме буде тестуватися (наприклад, функціональність, інтерфейс, взаємодія), які аспекти системи будуть перевірятися, а також умови та обмеження, за яких проводиться тестування.

Помилки в програмуванні та розробці програмного забезпечення, це непередбачувані проблеми або дефекти, які виникають у роботі програми та можуть привести до некоректної поведінки або невідповідності очікуванням користувача. Ключові аспекти які варто знати про помилки:

- непередбачені проблеми: баги можуть виникати з різних причин, таких як помилки у коді програми, неправильне використання або неправильне інтегрування залежностей, або неправильність у взаємодії з певними системними або апаратними компонентами;

- вплив на користувачів: баги можуть мати різні наслідки для користувачів, від незначних помилок, які не впливають на функціональність програми, до серйозних проблем, які можуть привести до аварійної зупинки програми або неправильної обробки даних;

- виявлення та виправлення: виявлення багів зазвичай відбувається шляхом тестування програмного забезпечення, включаючи ручне та автоматизоване тестування. Після виявлення багів розробники виправляють їх, а нові версії програми випускаються з виправленнями [24].

Приклад помилки у верстці системи, помилка у розділі новини «стаття» виходить за свої межі, зображено на рисунку 3.6.

Помилку функціоналу перевірки вже зареєстрованого користувача у системі зображено на рисунку 3.7.

Після виявлення помилок тест-кейс допомагає виправити їх шляхом систематичного відслідковування та оцінки результатів тестування. Після

виправлення помилок тест-кейс може бути виконаний знову для перевірки та підтвердження успішного виправлення.



Рисунок 3.6 – Помилка вигляду дизайну розділу “новини”

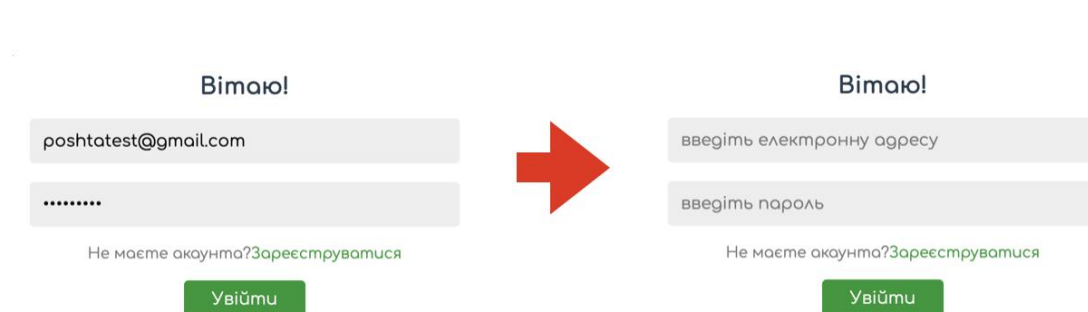


Рисунок 3.7 – Помилка функціоналу авторизації

Таким чином, тест-кейси сприяють покращенню якості програмного забезпечення. Тест-кейси представлені у додатку Б.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було:

1. проведено аналіз предметної області та існуючих аналогів, а саме: веб-портал "Дія" та портал державних послуг "IGOV", для визначення основних потреб та вимог до веб-орієнтованої інформаційної системи надання адміністративних послуг у період воєнного стану;
2. представлено загальну структуру, алгоритмічне та інформаційне забезпечення веб-орієнтованої інформаційної системи;
3. для створення зручного та інтуїтивного користувацького інтерфейсу були використані HTML, CSS та JavaScript. Entity Framework Core було використано для об'єктно-реляційного відображення даних та управління базою даних, а Visual Studio було обрано як основне середовище розробки для роботи з кодом;
4. реалізовано механізми автентифікації та авторизації користувачів для забезпечення безпечного доступу до системи; використання хешування паролів забезпечує захист конфіденційної інформації користувачів;
5. проведено тестування всіх компонентів системи.

У результаті розроблено веб-орієнтовану інформаційну систему надання адміністративних послуг у період воєнного стану, яка забезпечує оперативний доступ до переліку документів, які потрібно підготувати для подальшого використання та подання у вищі органи, координацію допомоги та інформаційну підтримку для військовослужбовців, їхніх сімей та внутрішньо переміщених осіб.

У майбутньому система може бути розширена для надання нових послуг та інтеграції з додатковими державними реєстрами.

Результати, отримані у кваліфікаційній роботі, будуть використовуватися в діяльності кафедри адміністративного права та судочинства Західноукраїнського національного університету, що підтверджується довідкою про використання (Додаток В).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Адміністративні послуги, які надаються та обмежуються під час війни. URL: <https://armyinform.com.ua/2022/06/17/administratyvni-poslugy-yaki-nadayutsya-ta-obmezhuyutsya-pid-chas-vijny/>
2. Веб-портал “ДІЯ”. URL: <https://diia.gov.ua/>
3. Веб-портал “IGOV”. URL: <https://igov.org.ua/>
4. Як створити власний домен? URL: <https://regeriy.ua/uk/blog/post/kak-sozdatdomen>
5. Клеппманн, М. Розробка додатків із інтенсивним використанням даних: великі ідеї, що лежать в основі надійних, масштабованих і зручних систем: монографія. Київ: Видавництво XYZ, 2017. 616 с.
6. Брайант, Р. Е., О’Галларон, Д. Р. Комп’ютерні системи: перспектива програміста: монографія. Київ: ABC, 2010. 976 с.
7. Седжвік, Р., Уейн, К. Алгоритми: монографія. Київ: DEF, 2011. 992 с.
8. Бхаргава, А. Алгоритми Грокінга: Ілюстрований посібник для програмістів та інших допитливих людей: монографія. Київ: GHI, 2016. 256 с.
9. Рафгарден, Т. Освітлені алгоритми: Основи: монографія. Київ: JKL, 2017. 218 с.
10. Документація по роботі з Entity Framework Core. URL: <https://learn.microsoft.com/en-us/ef/core/get-started/overview/first-app?tabs=netcore-cli>
11. Стеллман, Е., Грін, Дж. Head First C#: Посібник для навчання з програмування в реальному світі за допомогою C# і .NET Core: монографія. Київ: Видавництво XYZ, 2020. 800 с.
12. Документація по роботі з Visual Studio. URL: <https://learn.microsoft.com/en-us/visualstudio/windows/?view=vs-2022&preserve-view=true>
13. Документація по роботі з Visual Studio Code. URL: <https://code.visualstudio.com/docs>
14. Тарабанович І.Ю. Веб-орієнтована інформаційна система надання

адміністративних послуг у період воєнного стану. Сучасні світові тенденції розвитку науки та інформаційних технологій: матеріали VII міжнародної науково-практичної конференції, м. Одеса, 30-31.05.2024 р. С. XX-XX

15. Документація по роботі з Figma. URL: <https://help.figma.com/hc/en-us>
16. Джон Дакетт HTML і CSS: дизайн і створення веб-сайтів: 2011. 512 с.
17. Дженніфер Роббінс Навчання веб-дизайну: посібник для початківців з HTML, CSS, JavaScript і веб-графіки: 2018. 808 с.
18. David Flanagan JavaScript: Повний посібник : 2020. 706 с.
19. Документація по роботі з ASP.NET Core. URL: <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0>
20. Круг, С. Не змушуй мене думати, знову: підхід здорового глузду до веб-зручності: посібник. Київ: XYZ, 2014. 216 с.
21. Готелф, Д., Сейден, Дж. Learn UX: розробка чудових продуктів за допомогою гнучких команд: посібник. Київ: ABC, 2016. 208 с.
22. Норман, Д. Дизайн повсякденних речей: розширене видання: монографія. Київ: DEF, 2013. 368 с.
23. Хемблінг, Б. та ін. Тестування програмного забезпечення: Основний посібник для сертифікованих тестувальників ISTQB-BCS: монографія. Київ: Видавництво GHI, 2018. 296 с.
24. Грегори, Дж., Кріспін, Л. Стислий метод гнучкого тестування: короткий вступ: посібник. Київ: Видавництво JKL, 2019. 120 с.
25. Кінсбрунер, Е. Безперервне тестування для професіоналів DevOps: практичний посібник від експертів галузі: монографія. Київ: Видавництво MNO, 2020. 384 с.
26. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти.— Тернопіль: ЗУНУ, 2024.— 56 с.

Додаток А

Програмний код верстки сторінок веб-орієнтованої інформаційної системи

```
<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Home</title>
<link rel="stylesheet" href="/css/style.css">
</head>
<body>
<div class="title-site">  Система
надання адміністративних послуг у період воєнного стану</div>
<!-- header section starts -->
<header class="header">
<section class="flex">
<div id="menu-btn" class="fas fa-bars-staggered"></div>
<a href="home.html" class="html logo"> <i class="fas fa-briefcase"></i>DocHelp</a>
<nav class="navbar">
<a href="home.html">Головна</a>
<a href="about.html">Про нас</a>
<a href="news.html">Новини</a>
<a href="services.html">Послуги</a>
<a href="contact.html">Зворотній зв'язок</a>
</nav>
<a href="login.html" class="btn" style="margin-top: 0;">Увійти</a>
</section>
</header>
<!-- header section ends -->
<!-- home section starts -->
<div class="home-container">
<section class="home" id="home">
<div class="content">
<h3>Скористайтесь нашими послугами онлайн</h3>
<p>Наша платформа створена для того, щоб полегшити ваше життя, забезпечуючи
надійну та зручну взаємодію з адміністративними послугами навіть у найскладніші часи.
Незалежно від того, чи потрібна вам консультація, чи доступ до офіційних документів, наша
платформа готова забезпечити вам необхідну підтримку.</p>
<a href="#category" class="btn">Перейти до категорій</a>
</div>
</section>
</div>
<!-- home section ends -->
<!-- category section starts -->
<sectoin class="category" id="name">
<h1 class="heading">Категорії осіб</h1>
</sectoin>
<!-- military-category-section-starts -->
```

```

<div class="about-military">
<section>
<h1 class="heading-military">
<i id="category" class="fas fa-person-military-rifle"></i>
<span>Військовослужбовці</span>
</h1>
<div class="row-military">
<div class="image-military">

</div>
<div class="content-military">
<h3>what makes special?</h3>
<a href="#" class="btn">Перелік документів</a>
</div>
</div>
</section>
</div>
<!-- military-category-section-ends -->
<!-- immigrants-category-section-starts -->
<div class="about-immigrants">
<section>
<h1 class="heading-immigrants">
<i class="fa-solid fa-people-roof"></i>
<span>Внутрішньо переміщені особи</span>
</h1>
<div class="row-immigrants">
<div class="content-immigrants">
<h3>what makes special?</h3>
<a href="#" class="btn">Перелік документів</a>
</div>
<div class="image-immigrants">

</div>
</div>
</section>
</div>
<!--immigrants-category-section-ends-->
<!-- family-category-section-starts -->
<div class="about-family">
<section>
<h1 class="heading-family">
<i class="fas fa-people-group"></i>
<span>Сім'ї Військовослужбовців</span>
</h1>
<div class="row-family">
<div class="content-family">
<h3>what makes special?</h3>
<a href="#" class="btn">Перелік документів</a>
</div>
<div class="image-family">

</div>

```



```

flex-direction: column;
display: flex;
justify-content: center;
align-items: center;
width: 60%;
height: 150px;
border: 1px solid var(--black);
margin: auto;
margin-top: 20px;
text-align: center;
border-radius: 5px;
box-shadow: 2px 2px 10px black;
}
.zavantazhuty .pruclad{
    padding: 5px;
    transition: .5s ease;
}
.zavantazhuty .pruclad:hover{
    box-shadow: 2px 2px 10px black;
    border: 1px solid black;
}
/* section zayava ends */
/* footer styles starts */
.footer{
    background: var(--black);
    text-align: center;
}
.footer .share{
    padding: 1rem 0;
}
.footer .share a{
    height: 5rem;
    width: 5rem;
    line-height: 5rem;
    font-size: 2rem;
    color: #fff;
    border: var(--border);
    margin: .3rem;
    border-radius: 50%;
}
.footer .share a:hover{
    background-color: var(--main-color);
}
.footer .links{
    display: flex;
    justify-content: center;
    flex-wrap: wrap;
    padding: 2rem 0;
    gap: 1rem;
}
.footer .links a{
    padding: .7rem 2rem;

```

```
color: #fff;
border: var(--border);
font-size: 2rem;
}
.footer .links a:hover{
background: var(--main-color);
}
.footer .credit{
font-size: 2rem;
color: #fff;
font-weight: lighter;
padding: 1.5rem;
}
.footer .credit span{
color: var(--main-color);
}
/* footer styles ends */
@media (min-width: 991px) and (max-width: 1200px){
.news-section{
width: 60%;
height: auto;
}
header-news a{
margin-left: 10px;
}
}
```

Додаток Б

Тест-кейси

Таблиця Б.1 – Тест-кейс інтерфейсу системи

Система	Chrome	Safari	Коментарі
Наявність елементів сторінок			
Перевірка правильності відображення сторінок відповідно до зверстаних файлів	Пройдено	Пройдено	
Перевірка наявності логотипу	Пройдено	Пройдено	
Перевірка наявності логотипу	Пройдено	Пройдено	
Відображення елементів			
Перевірка відображення шрифтів	Пройдено	Пройдено	
Перевірка відображення кнопок, меню, блоків, ітд.	Пройдено	Пройдено	
Перевірка відображення застосованих кольорів	Пройдено	Пройдено	
Активні елементи			
Перевірка реакції активних елементів після наведення на них курсора	Не пройдено	Не пройдено	Після відкриття сторінки з переліком послуг, при наведенні курсором на вибрані елементи, реакції наведення не відбулося
Перевірка підказок на активних елементах	Пройдено	Пройдено	
Зміст сторінок			
Перевірка орфографії / граматики вмісту системи	Пройдено	Пройдено	
Перевірка відображення вмісту системи на різних пристроях	Не пройдено	Не пройдено	При переході на посилання системи на пристрої з розширенням екрану 394*667px розміри вмісту набагато більші за саме розширення пристрою
Перевірка відображення верхнього колонтитула	Пройдено	Пройдено	
Перевірка відображення нижнього колонтитула	Пройдено	Пройдено	
Перевірка відображення вмісту усіх сторінок	Не пройдено	Не пройдено	Після відкриття сторінки новини, прокрутивши в низ сторінки, видно що статті виходять за свої межі та виходять на межі нижнього колонтитула

Таблиця Б.2 – Тест-кейс функціоналу системи

Система	Chrome	Safari	Коментарі
Форма реєстрації та авторизації			
Перевірка на правильність заповнення форми email відбувається?	Пройдено	Пройдено	
Перевірка на правильність введених даних відбувається?	Пройдено	Пройдено	
Перевірка на відображення повідомлення, якщо користувач ввів неправильні дані	Пройдено	Пройдено	
Форма зворотного зв'язку			
Перевірка відображення шрифтів	Пройдено	Пройдено	
Перевірка відображення кнопок, меню, блоків, ітд.	Пройдено	Пройдено	
Перевірка відображення застосованих кольорів	Пройдено	Пройдено	
Активні елементи			
Перевірка реакції активних елементів після наведення на них курсора	Не пройдено	Не пройдено	Після відкриття сторінки з переліком послуг, при наведенні курсором на вибрані елементи, реакції наведення не відбулося
Перевірка підказок на активних елементах	Пройдено	Пройдено	
Зміст сторінок			
Перевірка орфографії / граматики вмісту системи	Пройдено	Пройдено	
Перевірка відображення вмісту системи на різних пристроях	Не пройдено	Не пройдено	При переході на посилання системи на пристрої з розширенням екрану 394*667px розміри вмісту набагато більші за саме розширення пристрою
Перевірка відображення верхнього колонтитула	Пройдено	Пройдено	
Перевірка відображення нижнього колонтитула	Пройдено	Пройдено	
Перевірка відображення вмісту усіх сторінок	Не пройдено	Не пройдено	Після відкриття сторінки новини, прокрутивши в низ сторінки, видно що статті виходять за свої межі та виходять на межі нижнього колонтитула

Додаток В

Довідка про використання результатів

Додаток Г

Копія опублікованих результатів