

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОНЛАЙН
АДМІНІСТРУВАННЯ СПІВБЕСІД
SOFTWARE DEVELOPMENT FOR ONLINE ADMINISTRATION OF
INTERVIEWS

Виконав(ла): студент(ка) 4 курсу, групи КНТ-220
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

ГЛАМАЗДІН Д.О.

(ПРИЗВИЩЕ та ініціали)

Керівник ЗЕЛЕНЬОВА І.Я.

(ПРИЗВИЩЕ та ініціали)

Рецензент ГОЛУБ Т.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГЛАМАЗДІНА Дмитра Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення для онлайн адміністрування співбесід.

Software development for online administration of interviews

керівник проєкту (роботи) к.т.н., доцент, ЗЕЛЕНЬОВА Ірина Яківна

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 24 ” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз існуючих рішень та визначення вимог до програмного забезпечення. 2. Проєктування системи. 3. Опис вебзастосунку. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЗЕЛЕНЬОВА І.Я., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “ 15 ” квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис)

Дмитро ГЛАМАЗДІН

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

Ірина ЗЕЛЕНЬОВА

(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 66 с., 13 табл., 24 рис., 2 дод., 12 джерел.

ВЕБЗАСТОСУНОК, ОНЛАЙН АДМІНІСТРУВАННЯ, СПІВБЕСІДИ, ASP.NET, ENTITY FRAMEWORK, SQL SERVER, JAVASCRIPT, HTML, CSS.

Об'єкт дослідження – процес планування, організації, проведення та аналізу співбесід..

Предмет дослідження – методи та технології управління співбесідами в корпоративному середовищі за допомогою програмного забезпечення.

Мета роботи – розробка вебзастосунку для ефективного онлайн адміністрування співбесід, що забезпечує зручність використання, безпеку даних та достатню функціональність для організації процесу проведення співбесід.

Матеріали, методи та технічні засоби: мова програмування C# та фреймворки ASP.NET і Entity Framework для серверної частини, HTML і CSS для клієнтської частини, середовище розробки Microsoft Visual Studio 2022, система управління базами даних Microsoft SQL Server, персональний комп'ютер під управлінням операційної системи Microsoft Windows 10.

Результати. Розроблено вебзастосунок для онлайн адміністрування співбесід. Застосунок має інтуїтивно зрозумілий інтерфейс користувача та надійну систему автентифікації й авторизації для безпеки даних.

Висновки. Результати дипломної роботи демонструють успішну реалізацію поставлених завдань та досягнення мети дослідження

Галузь використання – рекрутинг, організація корпоративних заходів.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 66 pages, 13 tables, 24 figures, 2 appendices, 12 sources.

WEB APPLICATION, ONLINE ADMINISTRATION, INTERVIEWS, ASP.NET, ENTITY FRAMEWORK, SQL SERVER, JAVASCRIPT, HTML, CSS.

The object of research is the process of planning, organizing, conducting, and analyzing interviews within a web application.

The subject of research is the methods and technologies for managing interviews in a corporate environment using software.

The aim of the work is to develop a web application for effective online administration of interviews, which provides ease of use, data security, and sufficient functionality for organizing the interview process.

Materials, methods, and technical means: C# programming language and ASP.NET and Entity Framework for the server-side, HTML and CSS for the client-side, Microsoft Visual Studio 2022 development environment, Microsoft SQL Server database management system, a personal computer running Microsoft Windows 10 operating system.

Results. A web application for online administration of interviews has been developed. The application has an intuitive user interface and a reliable authentication and authorization system for data security.

Conclusions. The results of the thesis demonstrate the successful implementation of the set tasks and the achievement of the research goal.

Field of application - recruiting, organization of corporate events.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз існуючих рішень та визначення вимог до програмного забезпечення	10
1.1 Огляд існуючих програмних продуктів.....	10
1.2 Аналіз функціональних можливостей та недоліків.....	13
1.3 Формування вимог до нової системи.....	15
1.4 Висновки до розділу 1	18
2 Проєктування системи.....	20
2.1 Вибір технологій та інструментів розробки.....	20
2.2 Проєктування архітектури системи	22
2.3 Опис середовища розробки.....	24
2.4 Висновок за розділом 2.....	26
3 Опис вебзастосунку	28
3.1 Опис бази даних	28
3.2 Структуру програмної частини	30
3.3 Інтерфейс користувача	32
3.4 Висновки за розділом 3	37
4 Експлуатація, тестування та експериментальне дослідження програми	39
4.1 Функціональне тестування.....	39
4.2 Навантажувальне тестування.....	41
4.3 Висновки за розділом 4	42
Висновки	44
Перелік джерел посилання	46
Додаток А Текст програми.....	46
Додаток Б Слайди презентації	59

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ASP.NET	- Active Server Pages для .NET;
AWS	- Amazon Web Services;
CSHTML	- C# HTML;
CSS	- Cascading Style Sheets;
HTML	- HyperText Markup Language;
IDE	- Integrated Development Environment;
MVC	- Model-View-Controller;
SQL	- Structured Query Language;
SSMS	- SQL Server Management Studio;
IT	- Інформаційні технології.

ВСТУП

У сучасному світі, де швидкість інформаційних потоків і вимоги до оперативності комунікації невідмінно зростають, значення ефективного управління зустрічами та співбесідами стає все більш вагомим. Це особливо актуально для бізнес-середовища, де час є критичним ресурсом, а кожна зустріч може вплинути на розвиток цілого підприємства. Розробка програмного забезпечення для онлайн-адміністрування співбесід відіграє ключову роль у підвищенні ефективності організаційних процесів [1].

Сучасні IT-технології, такі як хмарні рішення, штучний інтелект та машинне навчання, інтенсивно інтегруються в сферу управління співбесідами, надаючи можливості для автоматизації задач, покращення координації між учасниками і забезпечення вищої продуктивності співбесід. Ці технології дозволяють користувачам не тільки планувати зустрічі, але й ефективно керувати ними на основі аналітики поведінки учасників, оптимізації часу та ресурсів [2].

Основною метою розробки вебзастосунку для онлайн-адміністрування співбесід є створення зручного інструменту для ефективного планування, управління та аналізу корпоративних співбесід та співбесід, який дозволяє користувачам віддалено створювати, модифікувати, фільтрувати та відслідковувати події в реальному часі.

Предметом дослідження є методи та технології управління співбесідами в корпоративному середовищі за допомогою програмного забезпечення. Об'єктом є процес планування, організації, проведення та аналізу співбесід в рамках вебзастосунку.

Задачі роботи:

- аналіз існуючих рішень та визначення їх недоліків і обмежень;
- розробка вимог до функціоналу вебзастосунку, що підтримує створення, управління, фільтрацію та моніторинг співбесід;

- проектування архітектури програмного забезпечення з урахуванням масштабованості, безпеки та інтеграційної взаємодії з іншими системами;
- реалізація інтерфейсу користувача, що є інтуїтивно зрозумілим та зручним для користувачів з різним рівнем технічних навичок;
- тестування та оптимізація системи для забезпечення надійності та високої продуктивності в реальних умовах експлуатації.

Ці задачі відображають комплексний підхід до створення програмного забезпечення, яке має стати не лише інструментом для планування, але й значним ресурсом для підвищення ефективності організаційної діяльності.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Огляд існуючих програмних продуктів

У сучасному світі, де швидкість та ефективність інформаційних потоків визначають успіх бізнесу, актуальним є питання оптимізації процесу управління зустрічами та співбесідами. Наявність якісного програмного забезпечення для онлайн-адміністрування співбесід дозволяє підвищити продуктивність роботи, забезпечуючи більш ефективну організацію співбесід та краще використання робочого часу. Зі збільшенням кількості віртуальних команд та поширенням дистанційної роботи, ця потреба стає ще більш вираженою [1].

Проте, не всі існуючі рішення здатні задовольнити потреби сучасних організацій. Багато програм мають обмежений функціонал, не дозволяють інтегрувати внутрішні корпоративні системи або не надають достатньої гнучкості для кастомізації під специфічні потреби користувачів. Також, багато з них не забезпечують адекватного рівня безпеки даних, що є критично важливим для бізнесу.

Розробка нового програмного забезпечення для онлайн-адміністрування співбесід дозволяє вирішити ці проблеми, забезпечуючи платформу, яка буде відповідати всім сучасним вимогам безпеки, інтеграційної сумісності та користувацького досвіду. Така система може надати функціонал для автоматизації процесів планування, моніторингу участі, ведення переговорів та обміну матеріалами, що робить процеси більш ефективними та прозорими.

Однією з головних переваг новітнього програмного забезпечення для співбесід є можливість швидкого адаптування до змінюваних умов роботи та інтеграція з іншими корпоративними системами, такими як електронна пошта, календарі та системи документообігу. Це значно покращує

координацію між учасниками співбесід і забезпечує кращий контроль над корпоративними ресурсами.

На застосунок, актуальність розробки такого програмного забезпечення підкріплюється стрімким розвитком технологій та зростанням вимог до мобільності працівників. Сучасні рішення мають бути доступні з будь-якого пристрою та місця, забезпечуючи здатність до проведення співбесід у віртуальному режимі без втрати продуктивності та ефективності.

Отже, розробка програмного забезпечення для онлайн-адміністрування співбесід є не тільки актуальною, але й необхідною для сучасних компаній, що прагнуть підвищити ефективність своїх операційних процесів. Це створює фундамент для більш ефективної, гнучкої та продуктивної роботи, адаптованої до потреб цифрової ери.

На рисунку 1.1 відображено інтерфейс вебзастосунку для адміністрування співбесід, зокрема для створення та керування співбесідами співбесіди від сервісу work.ua. Основна вікно діалогу пропонує користувачу надіслати кандидатам повідомлення про заплановану зустріч. Користувач може обрати метод надсилання (емейл, SMS, push-повідомлення) та активувати нагадування за 30 хвилин до початку [3].

Функціонал сторінки включає:

- планування співбесід співбесіди та запрошення кандидатів;
- можливість вибору каналу зв'язку для надсилання інформації кандидатам;
- автоматизоване нагадування про майбутню подію;
- збір відповідей від кандидатів на запрошення.

Цей інтерфейс може значно спростити процес адміністрування співбесід, оскільки дозволяє централізовано управляти всіма аспектами організації співбесід, від запрошення учасників до моніторингу їхніх відповідей. Для розробки програмного забезпечення для онлайн-адміністрування співбесід важливо забезпечити інтуїтивно зрозумілий інтерфейс, що підтримує взаємодію з користувачем через різні канали

зв'язку, а також надає можливості для гнучкого керування розкладом і відстеження статусу учасників.

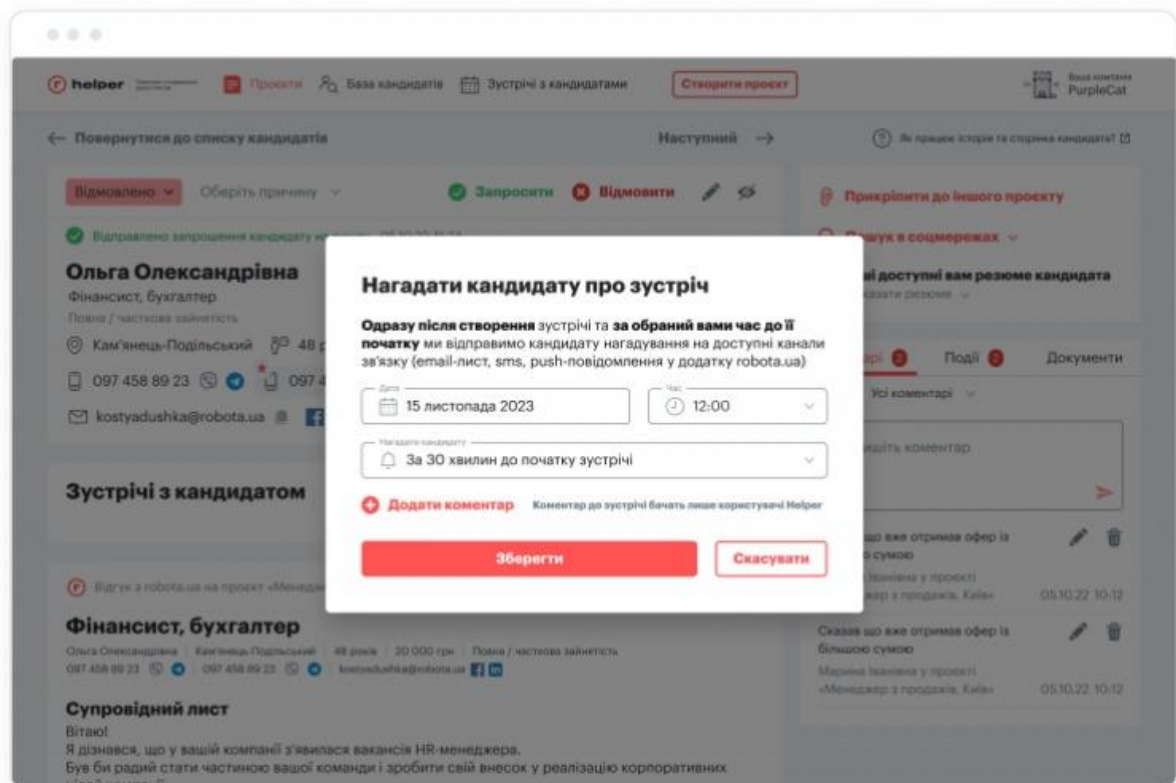


Рисунок 1.1 – Помічник від work.ua [3]

На рисунку 2.1 відображено головну сторінку вебсайту EasyWeek, який спеціалізується на рішеннях для управління бізнесом або організаційним плануванням. Вгорі сторінки розташоване горизонтальне меню з такими опціями: "Можливості", "Рішення", "Ресурси", "Ціни", "Увійти" та "Реєстрація", що надає користувачам швидкий доступ до різних сегментів сайту. Кнопка зеленого кольору у нижньому правому куті може вказувати на можливість швидкого зв'язку або доступу до підтримки [4].

Враховуючи елементи дизайну та функціональні можливості, зазначені на сторінці, можна зробити висновок, що сайт зосереджений на наданні рішень для планування та оптимізації бізнес-процесів, що може бути корисним для компаній, які прагнуть підвищити свою продуктивність і організаційну ефективність.

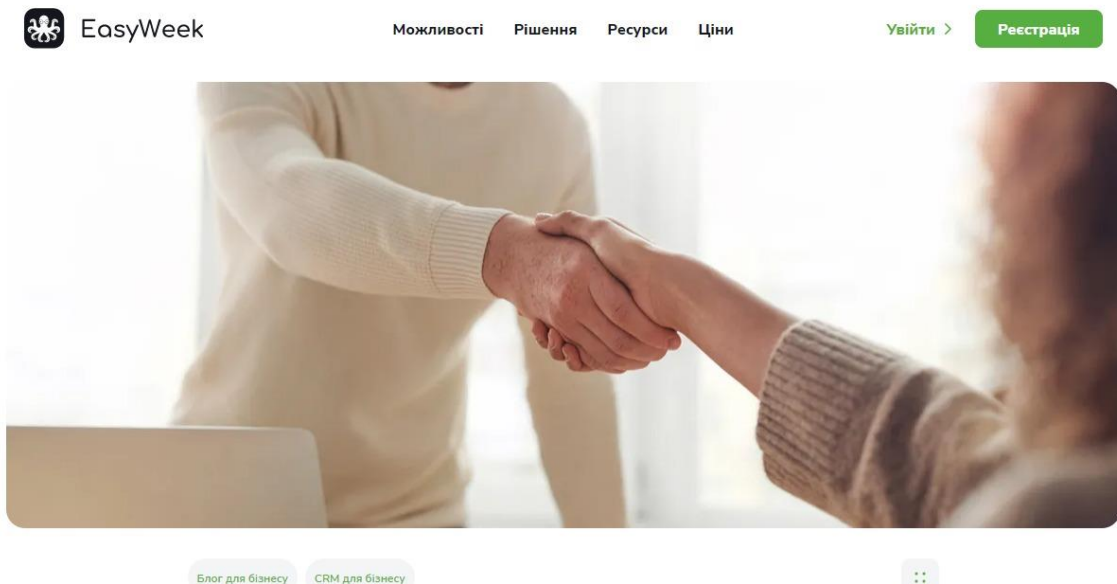


Рисунок 1.2 -EasyWeek [4]

Нарешті, є також альтернативні способи інформування про співбесіди, такі як канали в Telegram, які можуть ефективно досягати широкої аудиторії завдяки постійному використанню месенджера в повсякденному житті. Це дає можливість організаторам безпосередньо залучати увагу потенційних учасників, надсилаючи інформацію про співбесіди безпосередньо в повідомленнях, що зазвичай сприймаються легше і природніше, ніж традиційні методи реклами.

Спершу ми аналізуємо плюси та мінуси кожного з рішень, а потім проведемо їх порівняльний аналіз в рамках категорій, визначених раніше.

1.2 Аналіз функціональних можливостей та недоліків

Таблиця 1.1 порівнює ключові характеристики та особливості трьох обговорених сервісів: Helper, EasyWeek, Telegram-канали. Це допоможе більш чітко зрозуміти переваги та недоліки кожного з них.

Ця таблиця демонструє ключові особливості та відмінності між трьома платформами, допомагаючи користувачам вибрати найбільш підходящий сервіс відповідно до їхніх потреб і сфери діяльності.

Таблиця 1.1 – Порівняння сервісів

Критерій	Helper	EasyWeek	Telegram
Дизайн інтерфейсу	Має чіткий, організований інтерфейс для управління співбесідами.	Простий, сучасний дизайн з інтуїтивно зрозумілим навігаційним меню.	Мінімалістичний інтерфейс, орієнтований на спілкування.
Функціональність	Зосереджена на організації співбесід та співбесід.	Надає рішення для планування та оптимізації бізнес-процесів.	Включає миттєві повідомлення, групові чати, дзвінки, боти.
Оптимізація	Оптимізовано для специфічних завдань з управління співбесідами.	Високий рівень оптимізації для бізнес-задач.	Оптимізовано для швидкої та ефективної комунікації.
Надійність	Висока, з функціями автоматичних нагадувань.	Зазвичай висока, залежить від конкретного використання.	Висока, з надійними серверами та шифруванням даних.
Зручність користування	Інтуїтивно зрозумілий для користувачів HR.	Зручний для бізнес-користувачів, що планують ресурси.	Дуже зручний для повсякденного використання у спілкуванні.
Цільова аудиторія	HR-менеджери та організатори співбесід.	Керівники бізнесу, адміністратори планування.	Широка аудиторія, включаючи звичайних користувачів та бізнеси.

Таблиця 1.2 підсумовує загальні переваги та недоліки сервісів для організації співбесід та співбесід.

Таблиця 1.2 – Переваги та недоліки сучасних сервісів

Переваги/Недоліки	Helper	EasyWeek	Telegram
Переваги			
Зручність і доступність	Організація співбесід з різними каналами комунікації.	Легкість планування та оптимізації бізнес-процесів.	Легкий доступ та використання через мобільні та десктопні пристрої.
Широка аудиторія	Використовується HR-фахівцями в різних індустріях.	Підходить для різних розмірів бізнесів, зокрема середніх та малих.	Доступний та популярний у багатьох країнах світу.

Продовження таблиці 1.2

Переваги/Недоліки	Helper	EasyWeek	Telegram
Функціональність	Специфічні функції для управління співбесідами.	Різноманітність бізнес-інструментів для планування.	Широкий спектр можливостей для комунікації та обміну файлами.
Інтеграція	Можливість інтеграції з іншими HR-платформами.	Інтеграція з іншими бізнес-додатками та системами.	Інтеграція з багатьма сторонніми сервісами та ботами.
Недоліки			
Обмежена функціональність	Фокус лише на управлінні співбесідами, менш універсальний.	Може бути занадто складним для дуже малих підприємств.	Менше функцій для спеціалізованого бізнес-менеджменту.
Проблеми з дизайном та інтерфейсом	Інтерфейс може здатися перевантаженим для нових користувачів.	Дизайн може бути не таким інтуїтивним для окремих користувачів.	Інтерфейс може бути незручним для деяких бізнес-застосувань.
Проблеми з надійністю	Залежність від інтернет-з'єднання для функціональності.	Потребує стабільного з'єднання для управління ресурсами.	Залежність від серверів та потенційні проблеми з конфіденційністю.
Обмеження залучення через формат	Може не підходити для неформальних або невеликих організацій.	Може бути занадто складним або дорогим для дуже малих підприємств.	Інформація може загубитися серед численних повідомлень.

Отже, хоча ці сервіси пропонують багато переваг, що робить їх важливими інструментами для організаторів заходів, їх недоліки потребують уваги і можливих покращень, щоб забезпечити більш надійний та користувачам зручний досвід.

1.3 Формування вимог до нової системи

Розроблюваний вебзастосунок має стати зручним інструментом для адміністрування та планування різноманітних співбесід. Він повинен забезпечувати можливості, що спростять процес організації співбесід для адміністраторів та дозволять ефективно керувати ними.

Ключовими функціями застосунку є створення нових співбесід, редагування їхніх деталей, визначення прав доступу для інших користувачів, а також формування команди та розподіл завдань між її учасниками. Адміністратор події повинен мати швидкий доступ до всієї необхідної інформації про подію та змогу вносити зміни у разі потреби.

Інтерфейс користувача має бути простим та інтуїтивно зрозумілим. Він не повинен містити надлишкових елементів, які можуть відволікати чи заплутувати користувача. Меню навігації по застосунку має бути логічним та забезпечувати швидкий доступ до ключових функцій.

Застосунок повинен містити загальний список доступних співбесід, який можна фільтрувати за датою чи популярністю. Для кожної співбесіди передбачається окрема сторінка з детальною інформацією, яку можуть переглядати інші користувачі. Важливо реалізувати систему автентифікації та авторизації для забезпечення безпеки даних та розмежування прав доступу.

Вебзастосунок має поєднувати функції різних існуючих рішень для адміністрування співбесід та надавати їх у зручній та ефективній формі. Це дозволить адміністраторам заощадити час та зусилля на організацію співбесід, використовуючи єдиний інструмент замість набору розрізнених сервісів.

Розробка даного програмного забезпечення спрямована на спрощення та оптимізацію процесу онлайн-адміністрування співбесід та заходів. Воно має стати надійним помічником для організаторів різноманітних співбесід та забезпечити ефективну комунікацію між усіма залученими сторонами.

Дизайн застосунку має бути простим і зрозумілим. На головній сторінці буде розташований список усіх доступних співбесід, тоді як в лівому боці екрана розміщуватиметься навігаційна панель, що надає доступ до усіх функцій застосунку. Користувальницький інтерфейс має бути інтуїтивно зрозумілим і мінімізувати кількість помилок у процесі використання.

Після детального аналізу предметної області ми вирішили, що до складу вебзастосунку входитимуть наступні ключові функції: перегляд списку співбесід, можливість їх створення та редагування, призначення ролей учасникам співбесід, а також додавання завдань чи інформації залежно від ролі користувача.

Також було складено перелік завдань для реалізації проєкту:

- вибрати методологію розробки застосунку;
- створити діаграму класів;
- розробити базу даних;
- вибрати необхідні фреймворки та бібліотеки;
- реалізувати програмний комплекс;
- здійснити тестування програми.

Проєкт передбачає створення вебзастосунку, який спрощує організацію різноманітних заходів і може бути доповнений новими функціями у майбутньому (табл. 1.3).

Таблиця 1.3 – Вимоги до проєкту

Аспект	Опис
Функціональність	Застосунок дозволить створювати нові співбесіди, редагувати їх деталі, визначати права доступу для користувачів, формувати команди та розподіляти завдання.
Доступ до інформації	Адміністратори матимуть швидкий доступ до всієї необхідної інформації про співбесіди та зможуть вносити зміни у разі потреби.
Інтерфейс користувача	Інтерфейс буде простим та інтуїтивно зрозумілим, без надлишкових елементів, що можуть відволікати чи заплутувати користувача. Навігаційна панель забезпечить логічний та швидкий доступ до ключових функцій.

Продовження таблиці 1.3

Аспект	Опис
Лістинг співбесід	Головна сторінка міститиме загальний список доступних співбесід, який можна буде фільтрувати за датою чи популярністю. Кожна подія матиме окрему сторінку з детальною інформацією.
Безпека	Реалізація системи автентифікації та авторизації для забезпечення безпеки даних та розмежування прав доступу.
Інтеграція	Застосунок поєднуватиме функції різних існуючих рішень, надаючи їх у зручній та ефективній формі, що дозволить адміністраторам заощадити час та зусилля на організацію співбесід.
Майбутні функції	Вебзастосунок може бути доповнений новими функціями у майбутньому, що робить його адаптивним до змінних потреб і вимог.

Ця таблиця відображає основні елементи розробки вебзастосунку, який має на меті спростити та оптимізувати процес організації співбесід і заходів, забезпечуючи адміністраторам ефективний інструмент для управління цими процесами.

Він повинен бути легким у використанні та швидким у роботі. В процесі розробки можливі зміни та доповнення, що зумовлюються новими ідеями чи змінами у вимогах, тому можливості та переваги застосунку залишаються відкритими для корекцій.

1.4 Висновки до розділу 1

У першому розділі було проведено детальний аналіз предметної області та існуючих рішень для онлайн-адміністрування співбесід і заходів.

Розглянуто сильні та слабкі сторони популярних сервісів, таких як Helper, EasyWeek та Telegram-канали, які дозволяють організаторам створювати події та інформувати про них потенційних учасників.

Аналіз виявив, що ці сервіси мають ряд переваг, зокрема зручність та доступність для користувачів, широке охоплення аудиторії та функціональні можливості. Проте, були виявлені й значні недоліки, такі як проблеми з дизайном та інтерфейсом користувача, обмежена функціональність, ненадійність та незручності у залученні аудиторії.

На основі цього аналізу було визначено вимоги до нової системи для онлайн-адміністрування співбесід. Розроблюваний вебзастосунок має стати зручним та ефективним інструментом для організаторів заходів, що поєднує переваги існуючих рішень та уникає їх недоліків. Ключовими вимогами до системи є:

- широкі функціональні можливості для створення, редагування та управління співбесідами;
- інтуїтивно зрозумілий та мінімалістичний інтерфейс користувача;
- загальний список співбесід з можливістю фільтрації та деталізації інформації;
- надійна система автентифікації та авторизації для безпеки даних;
- інтеграція функцій різних сервісів в єдиному зручному застосунку;
- можливість розширення функціоналу в майбутньому.

Таким чином, в першому розділі закладено основу для розробки нового вебзастосунку, що має на меті оптимізувати та спростити процес онлайн-адміністрування співбесід і заходів, враховуючи потреби організаторів та недоліки існуючих рішень.

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Вибір технологій та інструментів розробки

При проєктуванні системи для онлайн-адміністрування співбесід та співбесід важливо обрати технології та інструменти, які забезпечать не тільки функціональність, але й надійність, масштабованість та високу продуктивність роботи. Основною метою є створення стійкої та гнучкої архітектури, що дозволить ефективно керувати всіма аспектами співбесід, від планування до виконання.

Серед популярних технологій для розробки вебзастосунків можна виділити JavaScript з бібліотекою React для фронтенду, що забезпечує швидке оновлення інтерфейсу без перезавантаження сторінки, а також Node.js для бекенду, що дозволяє використовувати JavaScript і для серверної частини, спрощуючи розробку і підтримку коду. Це також забезпечує кращу інтеграцію між клієнтською та серверною частинами [5].

Для управління даними рекомендується використання бази даних PostgreSQL, що є надійним рішенням з потужними можливостями для складного запитування та обробки великих масивів даних. PostgreSQL підтримує розширені функції SQL і може бути легко масштабований для обробки зростаючих обсягів інформації. [6]

Щодо вебсервера, Nginx або Apache виступають надійними варіантами для розгортання вебзастосунків, пропонуючи високу швидкість обробки запитів, стабільність та гнучке налаштування. Ці технології добре себе зарекомендували в різних операційних середовищах і мають широкую підтримку спільноти, що є важливим фактором для довгострокової підтримки та оновлення застосунку [5].

Розглядаючи системи управління версіями, Git є майже незамінним інструментом в сучасній розробці програмного забезпечення. Це дозволяє командам ефективно співпрацювати, контролюючи зміни в коді та забезпечуючи його консистентність і відкат змін при необхідності.

Таблиця 2.1 підсумовує вибір технологій та інструментів для розробки системи онлайн-адміністрування співбесід та співбесід, виглядає наступним чином.

Таблиця 2.1 – Стек технологій вебзастосунку

Технологія/Інструмент	Опис
JavaScript/React	Використання для фронтенду забезпечує швидке оновлення інтерфейсу без перезавантаження сторінки.
ASP.NET і Entity	Дозволяє використовувати JavaScript на серверній стороні, спрощуючи розробку і підтримку коду, забезпечуючи кращу інтеграцію між клієнтською та серверною частинами.
PostgreSQL	База даних, яка підтримує складні запити та великі обсяги даних, надає надійність та легку масштабованість.
Nginx/Apache	Вебсервери, які пропонують високу швидкість обробки запитів, стабільність і гнучке налаштування, ефективні в різних операційних середовищах.
Git	Система управління версіями, яка дозволяє командам контролювати зміни в коді, забезпечуючи консистентність і можливість відкату при необхідності.
AWS/Google Cloud	Хмарні платформи, що надають масштабовані обчислювальні ресурси та послуги зберігання даних, забезпечують високу доступність та надійність застосунку без значних інвестицій в власну IT-інфраструктуру.

Ця таблиця демонструє зв'язок між вибором технологій і задачами, які вони допомагають вирішити у контексті розробки вебзастосунку для адміністрування співбесід та співбесід. Вона висвітлює важливість вибору відповідних інструментів для забезпечення ефективності, надійності та масштабованості майбутньої системи.

Використання сучасних хмарних платформ, таких як AWS або Google Cloud, надає можливість використовувати масштабовані обчислювальні ресурси та послуги зберігання даних, що може бути істотно для забезпечення високої доступності та надійності застосунку без необхідності значних інвестицій в власну IT-інфраструктуру.

Такий підхід до вибору технологій та інструментів відображає зусилля зі створення ефективного, гнучкого та надійного програмного рішення, яке зможе вирішити сучасні виклики у сфері організації заходів та адміністрування співбесід.

2.2 Проектування архітектури системи

Будь-який вебзастосунок традиційно складається з декількох складових (рис.2.1). Частину, що є видимою для користувача, називають клієнтською. Це інтерфейс, через який користувач взаємодіє з внутрішньою логікою програми.

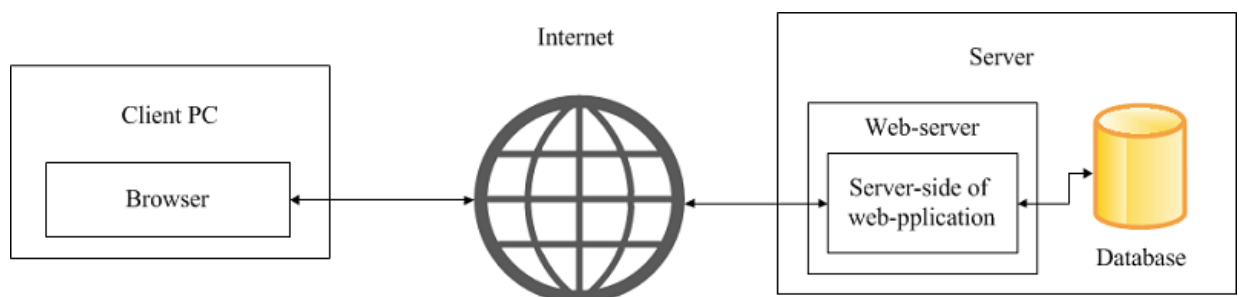


Рисунок 2.1 – Архітектура застосунку

Клієнтська частина містить засоби для візуалізації вебсторінок та первинну обробку запитів користувача, забезпечуючи, щоб кожна «кнопка» виконувала задану логіку, яка обробляється на сервері.

Зазвичай, для створення клієнтської частини використовуються такі технології як HTML для структури сторінок, CSS для оформлення і JavaScript для додавання інтерактивності. JavaScript застосовується для виконання клієнтських скриптів, наприклад, для динамічного зміни вмісту сторінки в залежності від даних, отриманих від сервера. У певних випадках JavaScript може бути замінений на інші мови або фреймворки, наприклад, якщо використовується C# разом з ASP.NET на серверній стороні, можливе використання файлів розмітки CSHTML замість стандартних HTML файлів [2].

Серверна частина, відповідає за основну функціональність вебзастосунку, яку користувачі не бачать. На сервері відбувається обробка даних, взаємодія з базами даних, виконання обчислень та формування даних для фронтенду. Бекенд може бути написаний на багатьох мовах, таких як JavaScript, Python, C#, Ruby.

Детальний опис обраних технологій.

Entity Framework є фреймворком для роботи з базами даних, який дозволяє взаємодіяти з даними як з об'єктами, мінімізуючи потребу в SQL коді [7].

HTML використовується для створення структури вебсторінок, визначаючи їх "скелет". В проєкті HTML код знаходиться у файлах CSHTML, які обробляються ASP.NET фреймворком [8].

CSS використовується для визначення стилю сторінок, створених за допомогою HTML, встановлюючи вигляд елементів на сторінці [2].

SQL використовується для взаємодії з реляційними базами даних, де дані зберігаються у вигляді таблиць, що забезпечує збереження цілісності інформації [10].

2.3 Опис середовища розробки

Розробка будь-якого програмного продукту потребує використання спеціалізованих інструментів - від простих текстових редакторів до потужних інтегрованих середовищ розробки (IDE). Для реалізації даного проєкту було обрано Microsoft Visual Studio 2022 - надзвичайно функціональне середовище, яке підтримує широкий спектр мов програмування або дозволяє розширити їх перелік за допомогою плагінів. Це дає змогу розробляти і налагоджувати код прямо всередині IDE [11].

Visual Studio 2022 має багату екосистему розширень і плагінів, інтеграцію з системою контролю версій git та хмарними сервісами Microsoft Azure. Середовище також оснащено вбудованим "інтелектуальним помічником", який показує перевантаження методів, підказує необхідні простори імен тощо. Дане програмне забезпечення доступне безкоштовно, має гнучкі налаштування зовнішнього вигляду та технічної конфігурації завдяки вищезгаданій системі плагінів.

Для роботи з реляційною базою даних було використано інший продукт Microsoft - SQL Server Management Studio (SSMS). Це безкоштовна утиліта, що слугує інтегрованим середовищем для управління SQL інфраструктурою: SQL Server, Azure SQL Database, MySQL. SSMS дозволяє створювати і обслуговувати локальні сервери баз даних, а також виконувати запити до існуючих баз [11].

Програма надає інструменти для налаштування безпеки серверів та баз даних, встановлення додаткових розширень і утиліт, проведення аналізу продуктивності та налагодження. Вона, як і Visual Studio, має відкриту архітектуру, що дозволяє розширювати її функціональність за допомогою плагінів.

Поєднання Visual Studio і SSMS створює зручне середовище для розробки вебзастосунків з базами даних. Зокрема, вони дозволяють швидко

створювати нові бази даних чи модифікувати існуючі структури за допомогою механізму міграцій [11].

Таблиця 2.2 підсумовує використані інструменти та їх характеристики у процесі розробки програмного продукту.

Таблиця 2.2 – Інструменти розробки

Інструмент	Опис
Microsoft Visual Studio 2022	Інтегроване середовище розробки, що підтримує багато мов програмування та дозволяє розширювати їх за допомогою плагінів. Забезпечує розробку та налагодження коду всередині IDE, інтегрується з системою контролю версій git та хмарними сервісами Microsoft Azure. Має багату екосистему розширень та інтелектуальні функції для підвищення продуктивності розробника.
SQL Server Management Studio	Безкоштовна утиліта для управління SQL інфраструктурою, включаючи SQL Server, Azure SQL Database, MySQL. Дозволяє створювати, обслуговувати сервери баз даних та виконувати запити. Надає інструменти для налаштування безпеки, встановлення розширень, проведення аналізу продуктивності. Можливість розширення функціональності через плагіни.
Інтеграція з Microsoft Azure	Обидва середовища інтегровані з хмарною платформою Azure, що дозволяє легко розгортати розроблені додатки в інтернеті. Azure масштабується згідно з потребами вебзастосунку і тарифікує лише фактично спожиті ресурси. Початковий рівень розміщення застосунку з базою даних до 1 ГБ на Azure є безкоштовним, а розширення обсягів сховища може коштувати 5-8€ на місяць. Забезпечує контроль над кодом і базою даних.

Ця таблиця відображає основні інструменти та їх можливості, використані у розробці вебзастосунків, акцентуючи на гнучкості та інтеграції

з хмарними сервісами, що є ключовими для сучасної розробки програмного забезпечення.

Оскільки обидва середовища тісно інтегровані з хмарною платформою Azure від Microsoft, розроблений застосунок можна легко розгорнути в інтернеті, а не лише на локальному сервері. Проте, чим більш потужним і вимогливим до ресурсів є застосунок, тим більше коштів необхідно витратити на його підтримку в хмарі. Azure масштабується відповідно до потреб вебзастосунку і тарифікує лише фактично спожиті ресурси, на відміну від фіксованих тарифних планів, які часто включають надлишкові ресурси [11].

Початковий рівень розміщення невеликого застосунку з базою даних до 1 ГБ на Azure є безкоштовним. Проте розширення обсягів сховища обійдеться приблизно в 5-8€ на місяць. Це забезпечить хостинг застосунку на серверах провайдера з виділеними обчислювальними потужностями і простором. За власне доменне ім'я потрібно доплачувати окремо або одразу купувати готовий пакет послуг за фіксованою місячною ціною. При цьому розробник зберігає повний контроль над кодом застосунку і базою даних, маючи доступ до всіх необхідних інструментів адміністрування.

2.4 Висновок за розділом 2

У другому розділі дипломної роботи було розглянуто ключові аспекти проєктування системи для онлайн-адміністрування співбесід та співбесід. Особлива увага приділялася вибору технологій та інструментів розробки, проєктуванню архітектури системи та опису середовища розробки.

Архітектура системи базується на традиційному поділі вебзастосунку на фронтенд та бекенд. Фронтенд відповідає за візуалізацію даних та взаємодію з користувачем, використовуючи HTML, CSS та JavaScript. Бекенд реалізує основну бізнес-логіку застосунку, обробку даних та взаємодію з базою даних, використовуючи C# та фреймворки ASP.NET і Entity.

Для розробки проєкту було обрано інтегроване середовище Microsoft Visual Studio 2022, яке надає потужні інструменти для розробки, налагодження та розгортання вебзастосунків. Для роботи з базою даних використовується SQL Server Management Studio. Обидва середовища тісно інтегровані з хмарною платформою Azure, що дозволяє легко масштабувати застосунок та керувати ресурсами.

3 ОПИС ВЕБЗАСТОСУНКУ

3.1 Опис бази даних

Основою для будь-якої системи адміністрування онлайн-співбесід є її модель даних. Модель даних визначає сталі правила для зберігання, взаємодії та відображення інформації в системі [12].

Розроблена система повинна дозволяти адміністраторам та користувачам ефективно зчитувати та записувати дані під час співбесід. Для цього вона має включати декілька таблиць і відповідні атрибути, що визначають їх функціональність. Конструкція моделі бази даних орієнтована на користувацькі потреби, визначаючи необхідні таблиці і атрибути для забезпечення робочих процесів та відкидаючи непотрібні.

Користувачі системи адміністрування співбесід повинні мати можливість виконувати наступні дії:

- відкрити систему за адресою у вебпереглядачі.
- зареєструватися, вносячи свої особисті дані, такі як електронна адреса, ім'я, пароль.
- призначати та відвідувати співбесіди.
- створювати та керувати списками кандидатів.
- переглядати звіти та відгуки про проведені співбесіди.

Для забезпечення функціональності використовується реляційна база даних, створена на платформі Microsoft SQL Server. Вона включає одинадцять SQL таблиць, 6 з яких були сформовані автоматично через міграції Entity Framework. Реляційний підхід допомагає уникнути змішування даних і полегшує управління, хоча ускладнює перевірку зв'язків між таблицями, що підвищує витрати на перевірку цілісності даних.

Таблиця "Адміністратори" призначена для зберігання інформації про осіб, що керують процесом проведення співбесід. Вона містить дані для авторизації та ідентифікації адміністраторів (табл. 3.1).

Таблиця 3.1 - Адміністратори

Колонка	Атрибут	Тип даних
AdminID	Унікальний ідентифікатор	uniqueidentifier
AdminName	Ім'я адміністратора	nvarchar(MAX)
AdminEmail	Електронна адреса	nvarchar(256)
LoginName	Логін	nvarchar(256)
LoginNormalized	Нормалізований логін	nvarchar(256)
EmailVerified	Підтвердження електронної пошти	bit
PasswordSecureHash	Хеш паролю	nvarchar(MAX)
SecurityCode	Відбиток безпеки	nvarchar(MAX)
ConcurrencyCode	Відбиток паралельності	nvarchar(MAX)

Таблиця "Співбесіди" зберігає дані про заплановані співбесіди, включаючи інформацію про адміністраторів, які їх організували, та основні деталі проведення (табл. 3.2).

Таблиця 3.2 - Співбесіди

Колонка	Атрибут	Тип даних
InterviewID	Унікальний ідентифікатор	uniqueidentifier
OrganizerID	Ідентифікатор організатора	uniqueidentifier
InterviewTitle	Назва співбесіди	nvarchar(MAX)
InterviewDescription	Опис співбесіди	nvarchar(MAX)
DateScheduled	Запланована дата	datetime2(7)
DateCreated	Дата створення	datetime2(7)

Таблиця "Кандидати" містить профілі людей, які беруть участь у співбесідах, з інформацією про їх статус та зв'язок з конкретними співбесідами (табл. 3.3).

Таблиця 3.3 - Кандидати

Колонка	Атрибут	Тип даних
CandidateID	Унікальний ідентифікатор	uniqueidentifier
RelatedInterviewID	Ідентифікатор співбесіди	uniqueidentifier
CandidateName	Ім'я кандидата	nvarchar(MAX)
CandidateEmail	Електронна адреса	nvarchar(256)
ApplicationStatus	Статус кандидата	nvarchar(MAX)
ApplicationDate	Дата подачі заявки	datetime2(7)
IsReviewed	Переглянуто	bit

Ці таблиці визначають основу для бази даних, забезпечуючи ефективне управління інформацією в рамках системи адміністрування співбесід.

3.2 Структуру програмної частини

Платформа для онлайн-адміністрування співбесід була розроблена з використанням архітектурного шаблону MVC. Таким чином, структура проєкту включає наступні елементи:

- директорія StaticAssets, яка зберігає допоміжні файли, включно з CSS стилями, стандартними зображеннями та іншими;
- директорія DBMigrations, де містяться всі виконані міграції;
- директорія CommandControllers, де розташовані класи контролерів (табл. 3.4);
- директорія DatabaseContexts, що включає файл контексту бази даних;
- директорія EntityModels, де зберігаються моделі даних (табл. 3.5);
- директорія InterfaceViews, що містить файли відображення (табл. 3.6);

- директорія ServiceRepositories, де розміщена власна реалізація сервісу аутентифікації користувачів.

Таблиця 3.4 – Вміст директорії CommandControllers.

Файл	Опис
AdminController.cs	Контролер для адміністраторів, без функціоналу для звичайних користувачів
HomeController.cs	Контролер головної сторінки
InterviewController.cs	Контролер для управління співбесідами
UserAuthenticationController.cs	Контролер для аутентифікації користувачів

Таблиця 3.5 – Вміст директорії EntityModels.

Файл	Опис
UserModel.cs	Модель даних користувача
InterviewModel.cs	Модель даних співбесід
ViewModel.cs	Загальні моделі для відображення даних на сторінках

Таблиця 3.6 – Вміст директорій ServiceRepositories та InterfaceViews.

Директорій	Файл	Опис
InterfaceViews	Index.cshtml	Головна сторінка застосунку
	Login.cshtml	Сторінка для аутентифікації користувачів
	Interview.cshtml	Сторінка для управління співбесідами
ServiceRepositories	UserRepository.cs	Реалізація сервісу для зберігання даних користувачів
	AuthRepository.cs	Реалізація сервісу аутентифікації користувачів

Ця структура забезпечує легке управління і модифікацію системи, оскільки всі ключові компоненти чітко відокремлені та організовані.

3.3 Інтерфейс користувача

Для оптимізації процесу впровадження запланованих функцій платформи онлайн-адміністрування співбесід було розроблено структурний план (рис. 3.1), який демонструє ключові рівні роботи платформи.

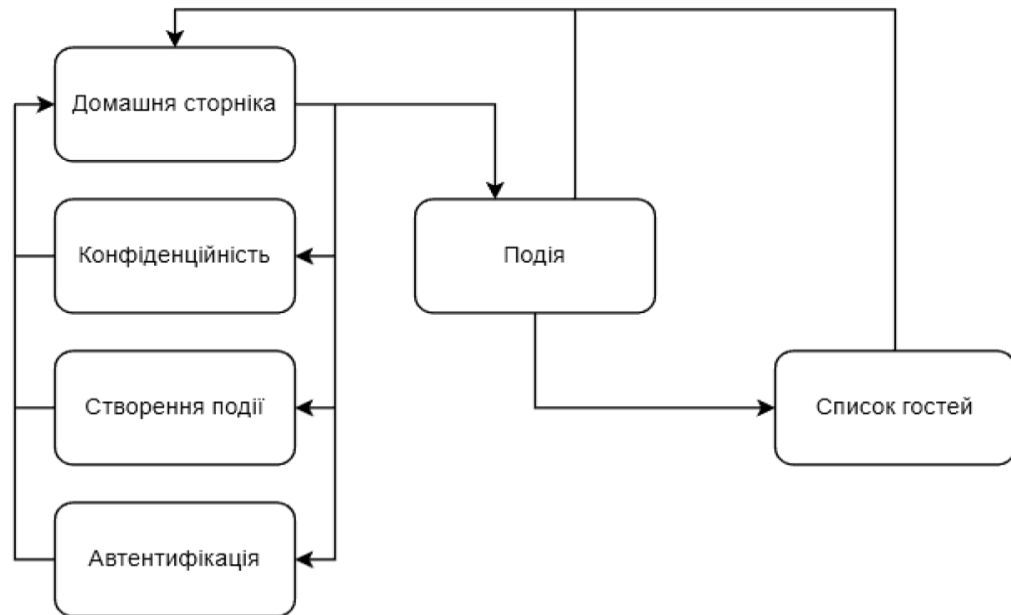


Рисунок 3.1 – Структура платформи

Робота на платформі починається з головної сторінки, яка логічно поділена на дві основні частини.

Перша частина – це "шапка" (рис. 3.2), видима з усіх сторінок платформи, що містить кнопки для переходу на такі розділи як: домашня сторінка, конфіденційність, планування співбесіди, вихід з акаунту. Для неавтентифікованих користувачів замість кнопки планування співбесіди відображається кнопка для входу.



Рисунок 3.2 – Заголовок вебсайту

Відповідно до адаптивного дизайну, на мобільних пристроях та пристроях з маленькими дисплеями шапка адаптується і зображується як випадаюче меню.

На рисунку 3.3 показано стартову сторінку, де відображаються всі співбесіди.

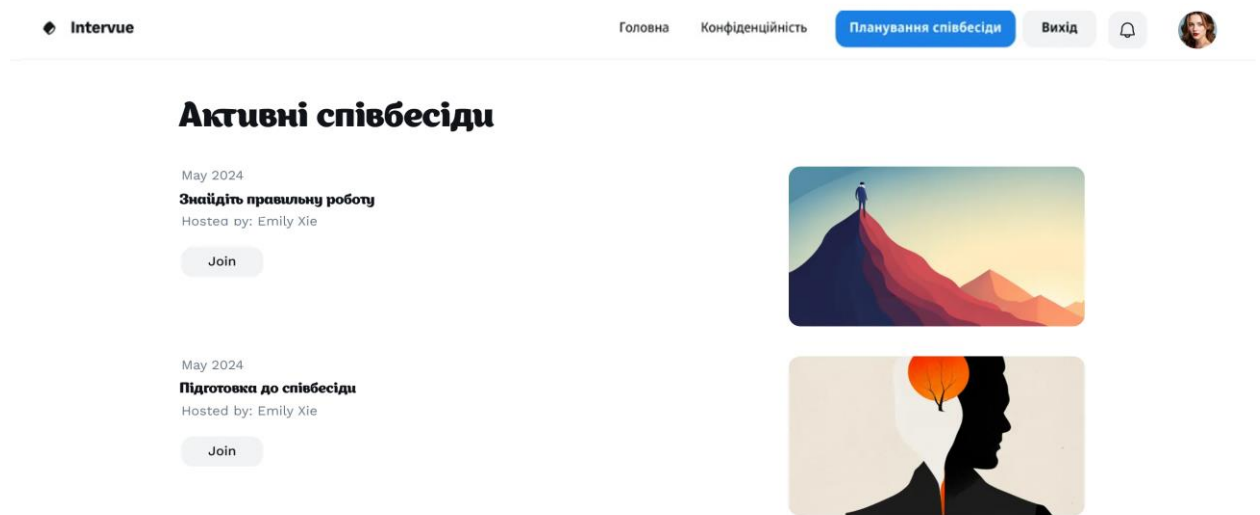


Рисунок 3.3 – Головна сторінка

Друга частина головної сторінки – це основний контент. Кожна сторінка платформи спільно використовує шапку, але має унікальний основний контент. На домашній сторінці основний контент включає поле пошуку за іменем учасника співбесіди та набір карток запланованих співбесід з основною інформацією про кожну з них.

Платформа взаємодіє з SQL базою даних, яка знаходиться на локальному сервері. При старті платформи формується рядок підключення, що визначає шлях до бази даних.

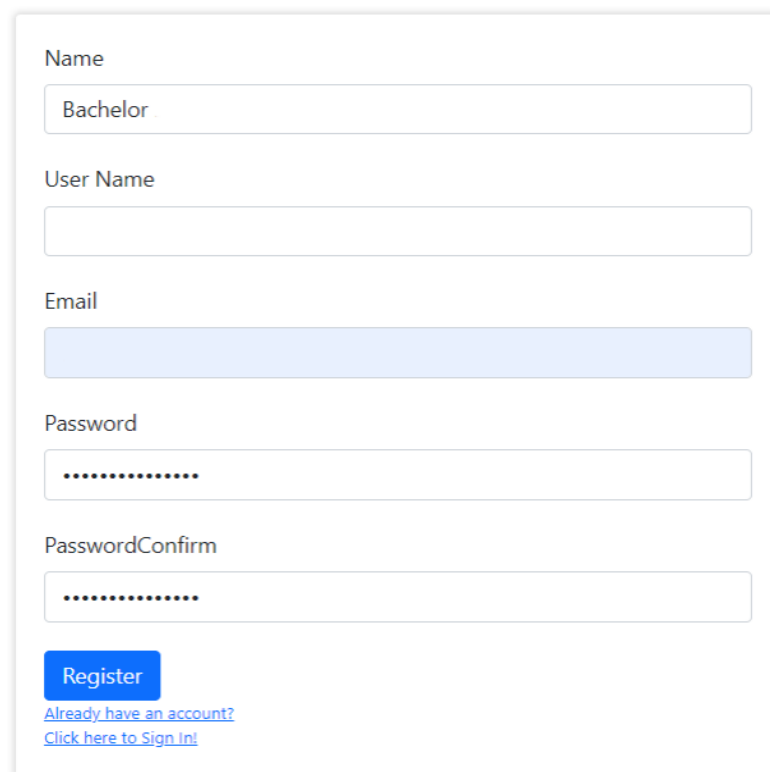
База даних зберігає інформацію про співбесіди, дані користувачів та гостей заходів. За допомогою вбудованих сервісів платформи ASP.NET, платформа забезпечує підключення до бази даних через рядок підключення.

Після успішного підключення до бази даних користувач потрапляє на вже описану домашню сторінку, з якої можна перейти на інші сторінки

платформи. Доступ до деяких сторінок обмежений для користувачів без аутентифікації.

Коли користувач вперше відвідує вебсайт платформи, він потрапляє на головну сторінку. Оскільки це новий користувач, на цьому етапі він може тільки переглядати анонси запланованих співбесід, створених іншими учасниками системи, не маючи можливості перегляду детальної інформації про них, оскільки доступ до деталей обмежений для неавторизованих користувачів. Також користувач може відвідати сторінку "Конфіденційність" та сторінку аутентифікації. Для неавторизованих користувачів кнопка для планування співбесід прихована, і якщо такий користувач спробує отримати доступ до сторінки планування співбесіди, він буде перенаправлений на сторінку аутентифікації.

Перейдемо до процесу аутентифікації користувача. Натиснемо на кнопку "Log In" у шапці платформи. Платформа перенаправить на відповідну сторінку з формою для входу (рис. 3.4).



The image shows a registration form with the following elements:

- Name:** A text input field containing the word "Bachelor".
- User Name:** An empty text input field.
- Email:** An empty text input field with a light blue background.
- Password:** A text input field with masked characters (dots).
- PasswordConfirm:** A text input field with masked characters (dots).
- Register:** A blue button with white text.
- Links:** Two blue links below the button: "[Already have an account?](#)" and "[Click here to Sign In!](#)".

Рисунок 3.4 – Форма реєстрації

Форма включає механізми валідації даних і відображають помилки залежно від того, чи коректно заповнені поля. Валідація перевіряє дані, надані користувачем, на відповідність заданим критеріям, наприклад, довжині паролю або наявності певних символів. Якщо дані не відповідають вимогам, з'являється відповідне повідомлення поруч з кнопкою "Log In" у формі входу і під полями вводу в формі реєстрації.

Після успішної аутентифікації платформа надає користувачу доступ до повної інформації про співбесіди. Користувач тепер може переглядати деталі співбесіди, натиснувши на кнопку "View" , яка тепер доступна на кожній оголошенні співбесіди, а також отримує доступ до кнопки запланування нової співбесіди, розміщеної в шапці. Ця кнопка веде до сторінки запланування співбесіди (рис. 3.15), яка містить поля для введення деталей співбесіди. Заповнивши ці поля і підтвердивши дію, користувач може організувати співбесіду.

Планування співбесіди

Дата

Оберіть дату

Час

Оберіть час

Учасники

Додайте учасників

Тема

Введіть тему співбесіди

Підтвердити

Рисунок 3.5 – Створення та планування співбесіди

Після створення співбесіди користувач може повернутися на головну сторінку або організувати ще одну співбесіду. Інформація про співбесіду буде відображатися у вигляді картки на фоні сторінки. Висота картки змінюється в залежності від об'єму тексту, який вона містить. На картці також є кнопки "Вподобати" і "Приєднатися", які для організатора співбесіди замінюються на "гості". Натискання на кнопку "Вподобати" збільшує індикатор інтересу до співбесіди, який відображається праворуч від кнопки. Натискання на кнопку "Приєднатися" додає унікальний ідентифікатор користувача до списку учасників співбесіди, який організатор може переглянути пізніше (рис. 3.6).

Організатор співбесіди також має можливість створювати завдання для співбесіди на сторінці зі списком учасників.

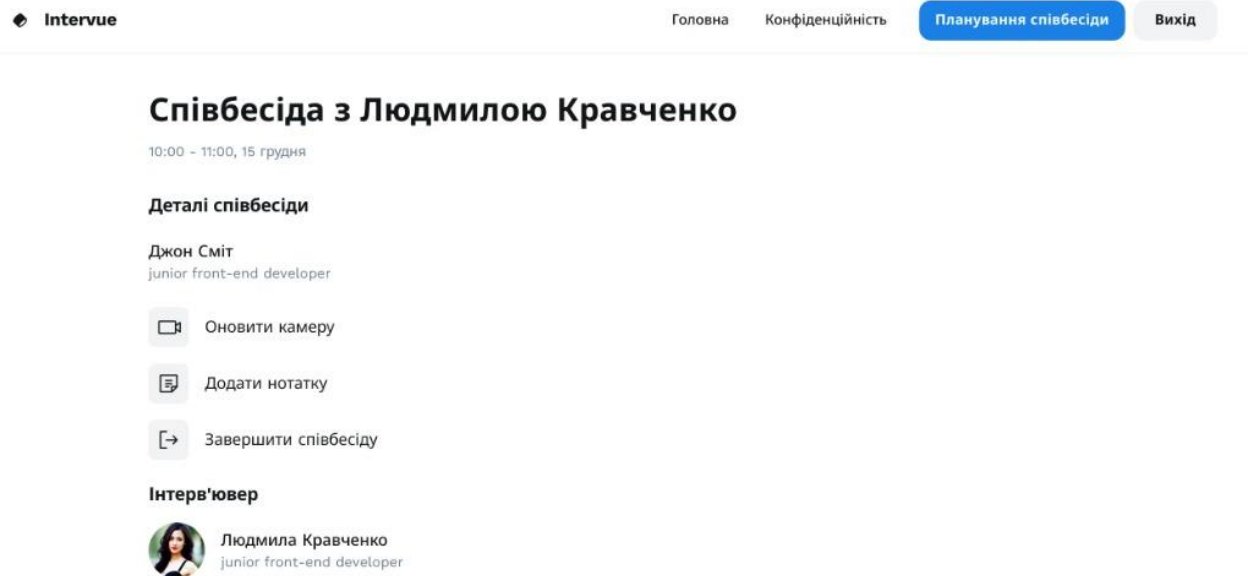


Рисунок 3.6 – Деталі співбесіди

3.4 Висновки за розділом 3

У третьому розділі дипломної роботи було детально описано розробку вебзатосунку для онлайн-адміністрування співбесід. Було розглянуто ключові аспекти реалізації системи, зокрема опис бази даних, структуру програмної частини та інтерфейс користувача.

База даних системи побудована з використанням реляційного підходу на платформі Microsoft SQL Server. Вона складається з трьох таблиць, які зберігають інформацію про адміністраторів, співбесіди та кандидатів. Реляційна модель дозволяє уникнути змішування даних і забезпечує ефективне управління інформацією.

Програмна частина застосунку реалізована з використанням архітектурного шаблону MVC та включає директорії для контролерів, моделей даних, представлень та сервісів. Такий підхід забезпечує чіткий поділ відповідальності та полегшує подальшу модифікацію і розширення системи.

Інтерфейс користувача розроблено з урахуванням вимог зручності та інтуїтивної зрозумілості. Він включає головну сторінку зі списком співбесід, сторінки для аутентифікації користувачів та планування нових співбесід. Використано адаптивний дизайн, який забезпечує коректне відображення елементів на різних пристроях.

Функціональність застосунку включає можливості для перегляду, створення, редагування співбесід, управління списком учасників та призначення їм завдань. Реалізовано механізми аутентифікації та авторизації для забезпечення безпеки даних.

Таким чином, розроблений вебзастосунок є повноцінним рішенням для онлайн-адміністрування співбесід, що забезпечує зручність використання, безпеку даних та достатню функціональність для ефективної організації процесу проведення співбесід. Обрана архітектура та технології дозволяють легко розширювати та модифікувати систему відповідно до майбутніх потреб.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Функціональне тестування

У процесі розробки платформи для онлайн-адміністрування співбесід кожна функція проходила тестування на працездатність. Таке тестування здійснювалося вручну під час використання платформи, без використання unit-тестів чи інших форм автоматизованого тестування.

Тестування включало перевірку таких функцій, як реєстрація користувачів (рис. 4.1), аутентифікація, авторизація, валідація введених даних, створення та управління співбесідами, перегляд детальної інформації про співбесіди, перегляд списку учасників (табл. 4.1).

Таблиця 4.1 – Зведене тестування функціоналу

Ім'я тесту	Вхідні дані	Виконувані дії	Очікуваний результат
Валідація форми реєстрації	Особисті дані: ім'я, email, пароль	Користувач заповнює поля форми, не відповідаючи встановленим правилам валідації	Неуспішна реєстрація з помилками валідації
Реєстрація користувача	Особисті дані: ім'я, email, пароль	Користувач заповнює поля форми згідно з встановленими правилами валідації	Успішна реєстрація
Аутентифікація користувача	Email, пароль	Користувач заповнює поля форми згідно з правилами валідації	Успішний вхід до системи
Авторизація користувача	Спроба доступу до закритих розділів	Неавторизований користувач намагається отримати доступ до сторінки створення співбесіди	Спрямування до сторінки аутентифікації

Продовження таблиці 4.1

Ім'я тесту	Вхідні дані	Виконувані дії	Очікуваний результат
Створення співбесіди	Деталі співбесіди	Авторизований користувач вводить всі необхідні дані і підтверджує створення	Співбесіда створена і доступна для перегляду
Приєднання до співбесіди	Вибір співбесіди	Користувач клікає на кнопку «Приєднатися» і система намагається знайти відповідну сторінку	Неуспішне приєднання через помилку знаходження сторінки

Ця таблиця покриває основні аспекти тестування функціоналу платформи, забезпечуючи зрозуміле і логічне представлення результатів для оцінки ефективності і надійності перед її запуском.

The image shows a registration form with the following fields and validation messages:

- Name:** A text input field containing "Validation check".
- User Name:** A text input field containing "Serj".
- Email:** A text input field containing "fdsfsdf". Below it, a red error message reads: "The Email field is not a valid e-mail address."
- Password:** A password input field with masked characters ".....". Below it, a red error message reads: "Minimum length 6 and must contain 1 Uppercase,1 lowercase, 1 special character and 1 digit".
- PasswordConfirm:** An empty text input field. Below it, a red error message reads: "Doesn't match with password".
- Buttons and Links:** At the bottom, there is a blue "Register" button, a blue link "Already have an account?", and a blue link "Click here to Sign In!".

Рисунок 4.1 – Валідація полів форми

4.2 Навантажувальне тестування

Для забезпечення стабільної роботи платформи онлайн-адміністрування співбесід в умовах реального використання було проведено навантажувальне тестування. Метою цього тестування було визначення здатності системи витримувати високі навантаження та великі обсяги даних без втрати продуктивності чи відмов у обслуговуванні.

Навантажувальне тестування проводилося з використанням спеціалізованого інструменту Apache JMeter. Цей інструмент дозволяє моделювати поведінку великої кількості одночасних користувачів, які взаємодіють з платформою, та вимірювати показники продуктивності, такі як час відповіді, використання ресурсів сервера та кількість помилок.

Тестові сценарії були розроблені для імітації реальних моделей використання платформи. Вони включали реєстрацію нових користувачів, аутентифікацію, створення та управління співбесідами, перегляд деталей співбесід та інші типові операції. Навантаження поступово збільшувалося, починаючи від кількох одночасних користувачів і досягаючи піків у сотні активних сесій.

Під час тестування відстежувалися ключові показники, такі як середній та максимальний час відповіді сервера, кількість запитів в секунду, використання процесора та пам'яті сервера. Ці показники дозволили оцінити здатність платформи обробляти високі навантаження та ідентифікувати потенційні вузькі місця чи проблеми з продуктивністю (табл. 4.2).

Результати навантажувального тестування показали, що платформа здатна витримувати значні навантаження без істотного погіршення продуктивності. Час відповіді залишався в прийнятних межах навіть при великій кількості одночасних користувачів, а використання ресурсів сервера було оптимальним.

Таблиця 4.2 – Показники навантаженого тестування

Кількість одночасних користувачів	Середній час відповіді (мс)	Максимальний час відповіді (мс)	Запити в секунду	Використан ня процесора (%)	Викорис тання пам'яті (МБ)
10	220	480	45	10%	180
50	320	780	190	25%	380
100	450	1150	350	45%	550
200	750	1600	580	65%	750
500	1100	2400	950	80%	1150

Однак, були виявлені деякі потенційні області для оптимізації. Зокрема, спостерігалися піки використання процесора під час інтенсивних операцій, таких як створення співбесід з великою кількістю учасників. Ці результати дозволили визначити напрямки для подальшого вдосконалення архітектури та оптимізації коду платформи.

Загалом, навантажувальне тестування підтвердило, що розроблена платформа здатна ефективно обробляти високі навантаження та забезпечувати стабільну роботу в умовах інтенсивного використання. Отримані результати дають впевненість у надійності та масштабованості системи, а також надають цінну інформацію для подальшої оптимізації та вдосконалення платформи онлайн-адміністрування співбесід.

4.3 Висновки за розділом 4

В четвертому розділі дипломної роботи було проведено експлуатаційне тестування та експериментальне дослідження розробленого вебзастосунку для онлайн-адміністрування співбесід. Метою цього етапу було перевірити відповідність системи висунутим вимогам, оцінити її надійність, продуктивність та здатність витримувати високі навантаження.

Функціональне тестування, яке проводилося вручну, охопило основні функції системи, такі як реєстрація користувачів, аутентифікація, авторизація, валідація даних, створення та управління співбесідами, перегляд детальної інформації та списку учасників. Результати тестування підтвердили коректну роботу цих функцій та їх відповідність встановленим вимогам.

Для оцінки здатності платформи витримувати високі навантаження було проведено навантажувальне тестування з використанням інструменту Apache JMeter. Тестові сценарії імітували реальні моделі використання системи з поступовим збільшенням кількості одночасних користувачів. Результати показали, що платформа здатна ефективно обробляти значні навантаження без істотного погіршення продуктивності, однак були виявлені деякі потенційні області для оптимізації.

Отримані результати тестування та експериментального дослідження дають впевненість у надійності, продуктивності та масштабованості розробленого вебзастосунку. Вони також надають цінну інформацію для подальшого вдосконалення та оптимізації системи.

У процесі експлуатації та тестування не було виявлено критичних помилок чи збоїв, що свідчить про високу якість розробки та відповідність системи встановленим вимогам. Однак, для забезпечення стабільної роботи в реальних умовах, рекомендується продовжувати моніторинг продуктивності та збір відгуків від користувачів для виявлення потенційних проблем та можливостей для покращення.

ВИСНОВКИ

У рамках роботи було успішно спроектовано та розроблено вебзастосунок для онлайн-адміністрування співбесід, який відповідає сучасним вимогам до ефективності, зручності використання та безпеки даних.

У процесі дослідження було проведено ґрунтовний аналіз предметної області та існуючих рішень для онлайн-адміністрування співбесід. Це дозволило визначити ключові недоліки та обмеження наявних систем, а також сформулювати вимоги до нового застосунку, який би поєднував переваги існуючих рішень та усував їх недоліки.

При проектуванні системи особлива увага приділялася вибору оптимальних технологій та інструментів розробки, які забезпечують надійність, масштабованість та високу продуктивність роботи. Архітектура системи побудована на основі шаблону MVC з використанням сучасних фреймворків та технологій, таких як ASP.NET, Entity Framework та HTML/CSS.

Розроблений вебзастосунок забезпечує широкий спектр функціональних можливостей для ефективного адміністрування співбесід, включаючи створення, редагування та управління співбесідами, призначення ролей учасникам, додавання завдань та інформації. Інтерфейс користувача є інтуїтивно зрозумілим та зручним у використанні, а система автентифікації та авторизації гарантує безпеку даних та розмежування прав доступу.

Проведене експлуатаційне тестування та експериментальне дослідження підтвердили відповідність розробленого застосунку встановленим вимогам, його надійність та здатність ефективно обробляти високі навантаження. Результати тестування також дозволили виявити потенційні області для оптимізації та вдосконалення системи.

Таким чином, розроблений вебзастосунок для онлайн-адміністрування співбесід є повноцінним та ефективним рішенням, яке може значно

спростити та оптимізувати процеси організації та проведення співбесід у сучасному бізнес-середовищі. Обрана архітектура та технології забезпечують можливість подальшого розширення функціоналу та адаптації системи до мінливих потреб користувачів.

У перспективі, розроблений застосунок може бути доповнений новими функціями, такими як інтеграція з системами календарів, автоматизація розсилки повідомлень та нагадувань, генерація звітів та аналітики. Це дозволить ще більше підвищити ефективність та зручність використання системи для організаторів та учасників співбесід.

Отже, результати дипломної роботи демонструють успішну реалізацію поставлених завдань та досягнення мети дослідження - створення ефективного та зручного вебзастосунку для онлайн-адміністрування співбесід, який відповідає сучасним вимогам та викликам у сфері управління співбесідами та організації бізнес-процесів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дронов В. А. HTML 5, CSS 3 та Web 2.0. Розробка сучасних Web-сайтів/В. А. Дронов. – К.:Друк, 2021. - 414 с.
2. Duckett J. HTML та CSS: Design and Build Websites / J. Duckett. - Indianapolis, IN: John Wiley & Sons, 2011. - 490 p.
3. Work.ua [Електронний ресурс]. Режим доступу: <https://www.work.ua/>
4. EasyWeek [Електронний ресурс]. Режим доступу : <https://www.easyweek.ua/>
5. Фрімен А. ASP.NET Core MVC із прикладами на С# для професіоналів / А. Фрімен. – Львів: ЛДВ, 2022. - 992 с.
6. Smith S. ASP.NET Core in Action / S. Smith. - Shelter Island, NY: Manning Publications, 2021. - 712 p.
7. Мак-Дональд М. WPF: Windows Presentation Foundation у .NET 4.5 із прикладами на С# 5.0 для професіоналів / М. Мак-Дональд. - Львів: ЛДВ, 2023. - 1024 с.
8. Nathan A. WPF 4.5 Unleashed / A. Nathan. - Indianapolis, IN: Sams Publishing, 2020. - 828 p.
9. Троєлсен Е. Мова програмування С# 7 і платформи .NET і .NET Core / Е. Троєлсен, Ф. Джепікс. - Львів: ЛДВ, 2021. - 1328 с.
10. Albahari J. C# 7.0 в Nutshell: Definitive Reference / J. Albahari, B. Albahari. - Sebastopol, CA: O'Reilly Media, 2019. - 1088 p.
11. Шілдт Г. С # 4.0: повне керівництво / Г. Шілдт. - К.:Друк, 2021. - 1056 с.
12. Skeet J. C# in Depth / J. Skeet. - Shelter Island, NY: Manning Publications, 2019. - 528 p.

ДОДАТОК А
Текст програми

```

Admin.cs
using System;
using System.ComponentModel.DataAnnotations;

namespace YourNamespace.EntityModels
{
    public class Admin
    {
        [Key]
        public Guid AdminID { get; set; }

        [Required]
        [MaxLength(256)]
        public string AdminName { get; set; }

        [Required]
        [EmailAddress]
        [MaxLength(256)]
        public string AdminEmail { get; set; }

        [Required]
        [MaxLength(256)]
        public string LoginName { get; set; }

        [Required]
        [MaxLength(256)]
        public string LoginNormalized { get; set; }
    }

    public bool EmailVerified { get; set; }

    [Required]
    public string PasswordSecureHash { get;
set; }

    [Required]
    public string SecurityCode { get; set; }

    [Required]
    public string ConcurrencyCode { get; set; }
}
}
}

```

Interview.cs

```

using System;
using System.ComponentModel.DataAnnotations;

namespace YourNamespace.EntityModels
{
    public class Interview
    {
        [Key]
        public Guid InterviewID { get; set; }

        [Required]
        public Guid OrganizerID { get; set; }

        [Required]
        [MaxLength(256)]
        public string InterviewTitle { get; set; }

        [MaxLength(1024)]
        public string InterviewDescription { get;
set; }

        [Required]
        public DateTime DateScheduled { get; set; }
    }

    public DateTime DateCreated { get; set; }
    = DateTime.UtcNow;
}

Candidate.cs
using System;
using System.ComponentModel.DataAnnotations;

namespace YourNamespace.EntityModels
{
    public class Candidate
    {
        [Key]
        public Guid CandidateID { get; set; }

        [Required]
        public Guid RelatedInterviewID { get; set; }
    }
}

```

```

        [Required]
        [MaxLength(256)]
        public string CandidateName { get; set; }

        [Required]
        [EmailAddress]
        [MaxLength(256)]
        public string CandidateEmail { get; set; }

        [MaxLength(256)]
        public string ApplicationStatus { get;
set; }

        public DateTime ApplicationDate { get;
set; }

        public bool IsReviewed { get; set; }
    }
}

```

```

AdminController.cs
using Microsoft.AspNetCore.Mvc;
using YourNamespace.Data;
using YourNamespace.EntityModels;
using System;
using System.Linq;
using System.Threading.Tasks;

namespace YourNamespace.CommandControllers
{
    public class AdminController : Controller
    {
        private readonly ApplicationDbContext
_context;

        public
AdminController(ApplicationDbContext context)
        {
            _context = context;
        }

        // GET: Admins
        public async Task<IActionResult> Index()
        {

```

```

        return View(await
_context.Admins.ToListAsync());
    }

    // GET: Admins/Details/5
    public async Task<IActionResult>
Details(Guid? id)
    {
        if (id == null)
        {
            return NotFound();
        }

        var admin = await _context.Admins
            .FirstOrDefaultAsync(m
m.AdminID == id);
        if (admin == null)
        {
            return NotFound();
        }

        return View(admin);
    }

    // GET: Admins/Create
    public IActionResult Create()
    {
        return View();
    }

    // POST: Admins/Create
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult>
Create([Bind("AdminID, AdminName, AdminEmail, LoginName, Lo
ginNormalized, EmailVerified, PasswordSecureHash, Security
Code, ConcurrencyCode")] Admin admin)
    {
        if (ModelState.IsValid)
        {
            admin.AdminID = Guid.NewGuid();
            _context.Add(admin);
            await _context.SaveChangesAsync();
            return
RedirectToAction(nameof(Index));

```

```

        }
        return View(admin);
    }

    // GET: Admins/Edit/5
    public async Task<IActionResult>
Edit(Guid? id)
    {
        if (id == null)
        {
            return NotFound();
        }

        var admin = await
_context.Admins.FindAsync(id);
        if (admin == null)
        {
            return NotFound();
        }
        return View(admin);
    }

    // POST: Admins/Edit/5
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Edit(Guid
id,
[Bind("AdminID, AdminName, AdminEmail, LoginName, LoginNorm
alized, EmailVerified, PasswordSecureHash, SecurityCode, Co
ncurrencyCode")] Admin admin)
    {
        if (id != admin.AdminID)
        {
            return NotFound();
        }

        if (ModelState.IsValid)
        {
            try
            {
                _context.Update(admin);
                await
_context.SaveChangesAsync();
            }

```

```

        catch
        (DbUpdateConcurrencyException)
        {
            if
            (!AdminExists(admin.AdminID))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return
        RedirectToAction(nameof(Index));
    }
    return View(admin);
}

// GET: Admins/Delete/5
public async Task<IActionResult>
Delete(Guid? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var admin = await _context.Admins
        .FirstOrDefaultAsync(m
        m.AdminID == id);
    if (admin == null)
    {
        return NotFound();
    }

    return View(admin);
}

// POST: Admins/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult>
DeleteConfirmed(Guid id)
{

```

```

        var admin = await
_context.Admins.FindAsync(id);
        _context.Admins.Remove(admin);
        await _context.SaveChangesAsync();
        return
RedirectToAction(nameof(Index));
    }

    private bool AdminExists(Guid id)
    {
        return _context.Admins.Any(e =>
e.AdminID == id);
    }
}

using Microsoft.AspNetCore.Mvc;
using YourNamespace.Data;
using YourNamespace.EntityModels;
using System;
using System.Linq;
using System.Threading.Tasks;

namespace YourNamespace.CommandControllers
{
    public class InterviewController : Controller
    {
        private readonly ApplicationDbContext
_context;

        public
InterviewController(ApplicationDbContext context)
        {
            _context = context;
        }

        // GET: Interviews
        public async Task<IActionResult> Index()
        {
            return View(await
_context.Interviews.ToListAsync());
        }

        // GET: Interviews/Details/5
        public async Task<IActionResult>
Details(Guid? id)

```

```

        {
            if (id == null)
            {
                return NotFound();
            }

            var interview = await
                _context.Interviews
                    .FirstOrDefaultAsync(m
                        =>
                        m.InterviewID == id);
            if (interview == null)
            {
                return NotFound();
            }

            return View(interview);
        }

        // GET: Interviews/Create
        public IActionResult Create()
        {
            return View();
        }

        // POST: Interviews/Create
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult>
            Create([Bind("InterviewID, OrganizerID, InterviewTitle, In
                terviewDescription, DateScheduled, DateCreated")]
                Interview interview)
        {
            if (ModelState.IsValid)
            {
                interview.InterviewID =
                    Guid.NewGuid();
                _context.Add(interview);
                await _context.SaveChangesAsync();
                return
                    RedirectToAction(nameof(Index));
            }
            return View(interview);
        }

        // GET: Interviews/Edit/5

```

```

        public async Task<IActionResult>
Edit(Guid? id)
    {
        if (id == null)
        {
            return NotFound();
        }

        var interview = await
_context.Interviews.FindAsync(id);
        if (interview == null)
        {
            return NotFound();
        }
        return View(interview);
    }

    // POST: Interviews/Edit/5
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Edit(Guid
id,
[Bind("InterviewID, OrganizerID, InterviewTitle, Interview
Description, DateScheduled, DateCreated")] Interview
interview)
    {
        if (id != interview.InterviewID)
        {
            return NotFound();
        }

        if (ModelState.IsValid)
        {
            try
            {
                _context.Update(interview);
                await
_context.SaveChangesAsync();
            }
            catch
(DbUpdateConcurrencyException)
            {
                if
(!InterviewExists(interview.InterviewID))
                {

```

```

        return NotFound();
    }
    else
    {
        throw;
    }
}
return
RedirectToAction(nameof(Index));
}
return View(interview);
}

// GET: Interviews/Delete/5
public async Task<IActionResult>
Delete(Guid? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var interview = await
_context.Interviews
.FirstOrDefaultAsync(m
=>
m.InterviewID == id);
    if (interview == null)
    {
        return NotFound();
    }

    return View(interview);
}

// POST: Interviews/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult>
DeleteConfirmed(Guid id)
{
    var interview = await
_context.Interviews.FindAsync(id);
    _context.Interviews.Remove(interview);
    await _context.SaveChangesAsync();
}

```

```
        return  
RedirectToAction (nameof (Index)) ;  
    }  
  
    private bool InterviewExists (Guid id)  
    {  
        return    _context.Interviews.Any (e    =>  
e.InterviewID == id) ;  
    }  
    }  
}
```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Розробка програмного забезпечення для онлайн адміністрування співбесід.

ВИКОНАВ СТУДЕНТ ГРУПИ КНТ-220
КЕРІВНИК К.Т.Н. ДОЦЕНТ

ДМИТРО ГЛАМАЗДІН
ІРИНА ЗЕЛЕНЬОВА

Рисунок Б.1 – Слайд 1

2

Мета та задачі

- ▶ Основною метою розробки вебзастосунку для онлайн адміністрування співбесід є створення зручного інструменту для ефективного планування, управління та аналізу корпоративних співбесід та подій, який дозволяє користувачам віддалено створювати, модифікувати, фільтрувати та відслідковувати події в реальному часі.
- ▶ Задачі роботи:
 - аналіз існуючих рішень та визначення їх недоліків і обмежень;
 - розробка вимог до функціоналу вебзастосунку, що підтримує створення, управління, фільтрацію та моніторинг подій;
 - проєктування архітектури програмного забезпечення з урахуванням масштабованості, безпеки та інтеграційної взаємодії з іншими системами;
 - реалізація інтерфейсу користувача, що є інтуїтивно зрозумілим та зручним для користувачів з різним рівнем технічних навичок;
 - тестування та оптимізація системи для забезпечення надійності та високої продуктивності в реальних умовах експлуатації.

Рисунок Б.2 – Слайд 2

Аналіз функціональних можливостей та недоліків сучасних засобів

3

Критерій	Helpert	EasyWeek	Telegram
Дизайн інтерфейсу	Має чіткий, організований інтерфейс для управління співбесідами.	Простий, сучасний дизайн з інтуїтивно зрозумілим навігаційним меню.	Мінімалістичний інтерфейс, орієнтований на спілкування.
Функціональність	Зосереджена на організації співбесід та подій.	Надає рішення для планування та оптимізації бізнес-процесів.	Включає миттєві повідомлення, групові чати, дзвінки, боти.
Оптимізація	Оптимізовано для специфічних завдань з управління співбесідами.	Високий рівень оптимізації для бізнес-задач.	Оптимізовано для швидкої та ефективної комунікації.
Надійність	Висока, з функціями автоматичних нагадувань.	Зазвичай висока, залежить від конкретного використання.	Висока, з надійними серверами та шифруванням даних.
Зручність користування	Інтуїтивно зрозумілий для користувачів HR.	Зручний для бізнес-користувачів, що планують ресурси.	Дуже зручний для повсякденного використання у спілкуванні.
Цільова аудиторія	HR-менеджери та організатори подій.	Керівники бізнесу, адміністратори планування.	Широка аудиторія, включаючи звичайних користувачів та бізнеси.

Рисунок Б.3 – Слайд 3

Вимоги до проєкту

4

Аспект	Опис
Функціональність	Застосунок дозволить створювати нові співбесіди, редагувати їх деталі, визначати права доступу для користувачів, формувати команди та розподіляти завдання.
Доступ до інформації	Адміністратори матимуть швидкий доступ до всієї необхідної інформації про співбесіди та зможуть вносити зміни у разі потреби.
Інтерфейс користувача	Інтерфейс буде простим та інтуїтивно зрозумілим, без надлишкових елементів, що можуть відволікати чи заплутувати користувача. Навігаційна панель забезпечить логічний та швидкий доступ до ключових функцій.
Лістинг подій	Головна сторінка міститиме загальний список доступних співбесід, який можна буде фільтрувати за датою чи популярністю. Кожна подія матиме окрему сторінку з детальною інформацією.
Безпека	Реалізація системи автентифікації та авторизації для забезпечення безпеки даних та розмежування прав доступу.
Інтеграція	Застосунок поєднуватиме функції різних існуючих рішень, надаючи їх у зручній та ефективній формі, що дозволить адміністраторам заощадити час та зусилля на організацію подій.
Майбутні функції	Вебзастосунок може бути доповнений новими функціями у майбутньому, що робить його адаптивним до змінних потреб і вимог.

Рисунок Б.4 – Слайд 4

5

Стек технологій вебзастосунку

Технологія/Інструмент	Опис
JavaScript/React	Використання для фронтенду забезпечує швидке оновлення інтерфейсу без перезавантаження сторінки.
ASP.NET i Entity	Дозволяє використовувати JavaScript на серверній стороні, спрощуючи розробку і підтримку коду, забезпечуючи кращу інтеграцію між клієнтською та серверною частинами.
PostgreSQL	База даних, яка підтримує складні запити та великі обсяги даних, надає надійність та легку масштабованість.
Nginx/Apache	Вебсервери, які пропонують високу швидкість обробки запитів, стабільність і гнучке налаштування, ефективні в різних операційних середовищах.
Git	Система управління версіями, яка дозволяє командам контролювати зміни в коді, забезпечуючи консистентність і можливість відкату при необхідності.
AWS/Google Cloud	Хмарні платформи, що надають масштабовані обчислювальні ресурси та послуги зберігання даних, забезпечують високу доступність та надійність застосунку без значних інвестицій в власну IT-інфраструктуру.

Рисунок Б.5 – Слайд 5

6

Архітектура застосунку

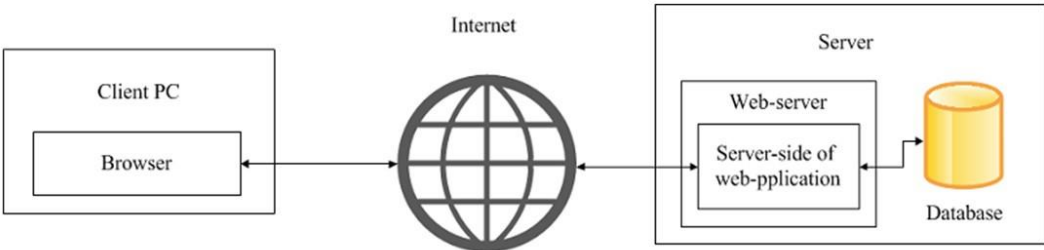


Рисунок Б.6 – Слайд 6

Опис бази даних

Колонка	Атрибут	Тип даних
AdminID	Унікальний ідентифікатор	uniqueidentifier
AdminName	Ім'я адміністратора	nvarchar(MAX)
AdminEmail	Електронна адреса	nvarchar(256)
LoginName	Логін	nvarchar(256)
LoginNormalized	Нормалізований логін	nvarchar(256)
EmailVerified	Підтвердження електронної пошти	bit
PasswordSecureHash	Хеш паролю	nvarchar(MAX)
SecurityCode	Відбиток безпеки	nvarchar(MAX)
ConcurrencyCode	Відбиток паралельності	nvarchar(MAX)

Колонка	Атрибут	Тип даних
InterviewID	Унікальний ідентифікатор	uniqueidentifier
OrganizerID	Ідентифікатор організатора	uniqueidentifier
InterviewTitle	Назва співбесіди	nvarchar(MAX)
InterviewDescription	Опис співбесіди	nvarchar(MAX)
Datescheduled	Запланована дата	datetime2(7)
DateCreated	Дата створення	datetime2(7)

Колонка	Атрибут	Тип даних
CandidateID	Унікальний ідентифікатор	uniqueidentifier
RelatedInterviewID	Ідентифікатор співбесіди	uniqueidentifier
CandidateName	Ім'я кандидата	nvarchar(MAX)
CandidateEmail	Електронна адреса	nvarchar(256)
ApplicationStatus	Статус кандидата	nvarchar(MAX)
ApplicationDate	Дата подачі заявки	datetime2(7)
IsReviewed	Переглянуто	bit

Рисунок Б.7 – Слайд 7

Склад проекту

Вміст директорії CommandControllers.

Файл	Опис
AdminController.cs	Контролер для адміністраторів, без функціоналу для звичайних користувачів
HomeController.cs	Контролер головної сторінки
InterviewController.cs	Контролер для управління співбесідами
UserAuthenticationController.cs	Контролер для аутентифікації користувачів

Вміст директорії EntityModels.

Файл	Опис
UserModel.cs	Модель даних користувача
InterviewModel.cs	Модель даних співбесіди
ViewModel.cs	Загальні моделі для відображення даних на сторінках

Вміст директорії ServiceRepositories та InterfaceViews.

Директорії	Файл	Опис
InterfaceViews	Index.cshtml	Головна сторінка застосунку
	Login.cshtml	Сторінка для аутентифікації користувачів
	Interview.cshtml	Сторінка для управління співбесідами
ServiceRepositories	UserRepository.cs	Реалізація сервісу для зберігання даних користувачів
	AuthRepository.cs	Реалізація сервісу аутентифікації користувачів

Рисунок Б.8 – Слайд 8

Структура платформи

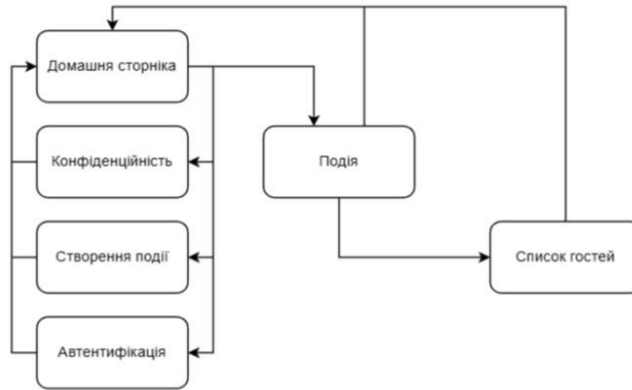


Рисунок Б.9 – Слайд 9

Інтерфейс користувача

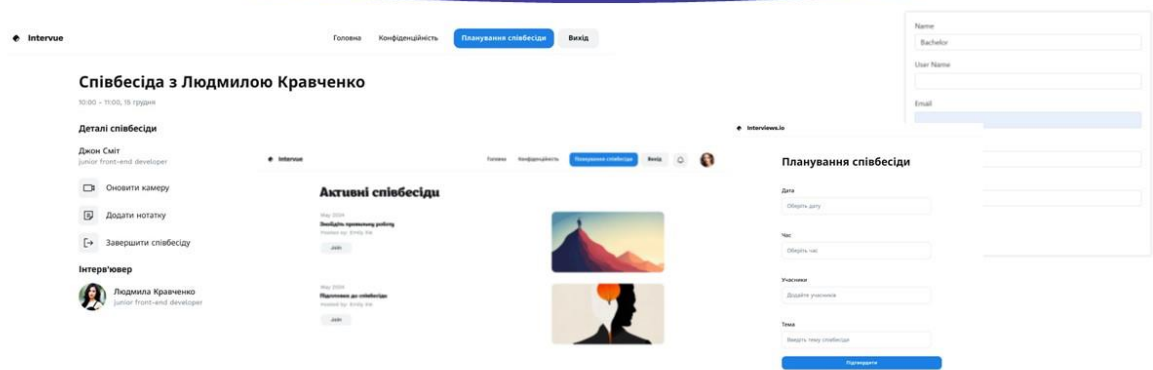


Рисунок Б.10 – Слайд 10

Функціональне тестування

Ім'я тесту	Вхідні дані	Виконувані дії	Очікуваний результат
Валідація форми реєстрації	Особисті дані: ім'я, email, пароль	Користувач заповнює поля форми, не відповідаючи встановленим правилам валідації	Неуспішна реєстрація з помилками валідації
Реєстрація користувача	Особисті дані: ім'я, email, пароль	Користувач заповнює поля форми згідно з встановленими правилами валідації	Успішна реєстрація
Аутентифікація користувача	Email, пароль	Користувач заповнює поля форми згідно з правилами валідації	Успішний вхід до системи
Авторизація користувача	Спроба доступу до закритих розділів	Неавторизований користувач намагається отримати доступ до сторінки створення співбесіди	Спрямовання до сторінки аутентифікації
Створення співбесіди	Деталі співбесіди	Авторизований користувач вводить всі необхідні дані і підтверджує створення	Співбесіда створена і доступна для перегляду
Приєднання до співбесіди	Вибір співбесіди	Користувач клікає на кнопку «Приєднатися» і система намагається знайти відповідну сторінку	Неуспішне приєднання через помилку знаходження сторінки

Рисунок Б.11 – Слайд 11

Показники навантаженого тестування

Кількість одночасних користувачів	Середній час відповіді (мс)	Максимальний час відповіді (мс)	Запити в секунду	Використання процесора (%)	Використання пам'яті (МБ)
10	220	480	45	10%	180
50	320	780	190	25%	380
100	450	1150	350	45%	550
200	750	1600	580	65%	750
500	1100	2400	950	80%	1150

Рисунок Б.12 – Слайд 12

Висновки

У рамках дипломної роботи було успішно спроектовано та розроблено вебзастосунок для онлайн адміністрування співбесід, який відповідає сучасним вимогам до ефективності, зручності використання та безпеки даних. Проведений аналіз предметної області та існуючих рішень дозволив визначити ключові недоліки наявних систем і сформулювати вимоги до нового застосунку. Архітектура системи побудована на основі шаблону MVC з використанням ASP.NET, Entity Framework та HTML/CSS. Розроблений застосунок забезпечує широкий спектр функціональних можливостей для адміністрування співбесід, має інтуїтивно зрозумілий інтерфейс та надійну систему автентифікації й авторизації. Тестування підтвердило відповідність застосунку вимогам, його надійність та здатність ефективно обробляти високі навантаження. Обрана архітектура та технології забезпечують можливість подальшого розширення функціоналу та адаптації системи до мінливих потреб користувачів. Результати роботи демонструють успішну реалізацію поставлених завдань та досягнення мети дослідження.

Рисунок Б.13 – Слайд 13