

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка сайту для віддаленого ведення кадрового аудиту з використанням C# та .Net»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

(підпис) Микита СТРУК

Виконав: здобувач вищої освіти групи ПД-44

Микита СТРУК

Керівник: Вікторія ЖЕБКА
д.т.н., професор,

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Струку Микиті Юрійовичу

1. Тема кваліфікаційної роботи: «Розробка сайту для віддаленого ведення кадрового аудиту з використанням C# та .Net»

керівник кваліфікаційної роботи д.т.н., проф., зав. кафедри ТЦР Вікторія ЖЕБКА,
затверджені наказом Державного університету інформаційно-комунікаційних
технологій від «27» лютого 2024 р. № 36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи:

3.1.Офіційна документація C#, .NET.

3.2.Офіційна документація Microsoft Visual Studio.

3.3.Науково-технічна література.

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

4.1.Аналіз актуальності та огляд існуючих додатків.

4.2.Аналіз інструментів реалізації та розробка структури ПЗ.

4.3.Реалізація програмної частини додатку.

4.4.Висновки.

5. Перелік графічного матеріалу: *презентація*

5.1 Аналіз аналогів

5.2 Вимоги до ПЗ

5.3 Програмні засоби для розробки

5.4 Діаграма варіантів використання

5.5 Діаграма класів

5.6 Файлова структура проекту

5.7 Мапа сайту

5.8 Екрані форми

5.9 Відео роботи додатку

5.10 Апробація результатів дослідження

6. Дата видачі завдання «28» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	28.02.-06.03.2024	виконано
2	Аналіз та дослідження існуючих аналогів	07.03-13.03.2024	виконано
3	Огляд існуючих мов програмування. вибір платформи, фреймворків.	14.03-20.03.2024	виконано
4	Проектування застосунку для кадрового аудиту, контролю кадрів	21.03-28.03.2024	виконано
5	Програмна реалізація веб-додатку	28.03-10.04.2024	виконано
6	Тестування веб-додатку	11.04-28.04.2024	виконано
7	Оформлення роботи: вступ, висновки, реферат	29.04-05.05.2024	виконано
8	Розробка демонстраційних матеріалів	06.05-12.05.2024	виконано
9	Попередній захист роботи	13.05-31.05.2024	виконано

Здобувач вищої освіти

_____ (підпис)

Микита СТРУК

Керівник

кваліфікаційної роботи

_____ (підпис)

Вікторія ЖЕБКА

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 40 стор., 2 табл., 10 рис., 15 джерел.

Мета роботи – спрощення процесу ведення кадрового аудиту, за рахунок використання програмного забезпечення розроблено з використанням мови C# та .net

Об'єкт дослідження – Процес проведення віддаленого кадрового аудиту..

Предмет дослідження – Програмне забезпечення для віддаленого ведення кадрового аудиту. *Короткий зміст роботи:* Робота включає розробку веб-сайту, який дозволяє полегшити ключові аспекти кадрового аудиту, зокрема збирання, зберігання та аналіз даних про працівників. Впровадження такого рішення забезпечує підвищення ефективності та прозорості процесів, зменшення часу на виконання рутинних завдань та зниження ймовірності помилок, пов'язаних з людським фактором. Описано етапи розробки, технології та інструменти, що використовувалися, а також результати тестування створеної системи у реальних умовах. Розроблено алгоритм роботи застосунку та програмно реалізовані ключові функціональні можливості, зокрема: Create, Update, Delete операції з БД, реєстрація користувачів, авторизація, обмеження доступу та реалізовано графічний UI.

Проведено тестування додатку. В роботі використано мову програмування C# та платформу ASP.NET, бібліотека Entity Framework та БД MSSQL.

Сферою використання застосунку є спрощення кадрового контролю на підприємствах та бізнесах, перенесення його у цифру.

КЛЮЧОВІ СЛОВА: КАДРОВИЙ АУДИТ, КАДРОВИЙ МЕНДЖМЕНТ, УПРАВЛІННЯ КАДРАМИ, C#, MSSQL, Entity Framework.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	11
1.1 Поняття кадрового аудиту, контролю кадрів.....	11
1.2 Аналіз програмних рішень для предметної області.....	12
1.3 Формулювання вимог до ПЗ.....	21
2 ВИБІР ЗАСОБІВ ТА БІБЛІОТЕК ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ.....	23
2.1 Середовище розробки	23
2.2 Мова програмування та платформа	25
2.3 Бібліотеки.....	27
2.4 База даних.....	29
3 ПРОЕКТУВАННЯ, РОЗРОБКА, ТА ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	31
3.1 Архітектура веб-додатку	31
3.2 Архітектура БД додатку	37
3.3 Тестування.....	42
ВИСНОВКИ.....	44
ПЕРЕЛІК ПОСИЛАНЬ.....	47
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	49

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

MSSQL - Microsoft SQL Server

БД – База даних

C# - Мова програмування

CSS – Cascading Style Sheets

ORM – Object-Relational Mapping

EF – Entity Framework

API - Application Programming Interface

LINQ - Language Integrated Query

ВСТУП

Сучасний світ бізнесу та управління кадрами динамічно змінюється під впливом стрімкого розвитку інформаційних технологій. В умовах глобалізації та цифрової трансформації виникає потреба у нових підходах до управління персоналом, що дозволяють підвищити ефективність, гнучкість та прозорість процесів. Одним з таких підходів є використання веб-технологій для віддаленого ведення кадрового аудиту.

Кадровий аудит є невід'ємною частиною управління персоналом, адже він дозволяє виявляти та усувати проблеми у кадровій політиці, оптимізувати роботу з персоналом та забезпечувати відповідність вимогам законодавства. Традиційні методи проведення кадрового аудиту часто потребують значних ресурсів, часу та зусиль, що може уповільнювати діяльність компанії. Тому виникає необхідність у створенні інструментів, що дозволяють спростити цей процес.

У зв'язку з цим, метою даної дипломної роботи є розробка веб-сайту для віддаленого ведення кадрового аудиту з використанням C# та платформи ASP.NET. Такий підхід дозволяє забезпечити доступність, зручність та ефективність процесу кадрового аудиту, зменшити витрати часу та ресурсів на його проведення, а також підвищити точність та надійність отриманих даних.

. По-перше, ці технології забезпечують високу продуктивність та безпеку додатків. По-друге, платформа .NET підтримує інтеграцію з іншими системами та базами даних, що є важливим для забезпечення комплексного підходу до управління кадрами. По-третє, C# є мовою програмування, яка поєднує простоту та потужність, що дозволяє розробляти ефективні та масштабовані рішення.

Особливо актуальним стає питання впровадження цифрових рішень у бізнесі під час війни. В умовах військових конфліктів, підприємства стикаються з низкою додаткових проблем, таких як загроза фізичної втрати паперової документації через бойові дії або евакуацію. Втрата важливих кадрових документів може мати катастрофічні наслідки для бізнесу, включаючи нездатність підтвердити трудовий стаж працівників, втрату даних про кваліфікацію та досвід персоналу, а також

складнощі з відновленням інформації для аудиту та звітності.

Цифровізація кадрових процесів, а конкретно - створення веб-сайту для віддаленого ведення кадрового аудиту, обліку кадрів це дуже важливий фактор для збереження та безпеки інформації. Веб-технології забезпечують можливість надійного зберігання даних у хмарних сервісах, доступ до інформації з будь-якої точки світу та захист від втрати фізичних копій та оригіналів. Крім того, це дозволяє забезпечити безперервність бізнес-процесів навіть у кризових ситуаціях, таких як Війна.

У рамках даної дипломної роботи буде проведено аналіз існуючих методів та інструментів для кадрового аудиту, визначено основні вимоги до розроблюваного веб-сайту, спроектовано його архітектуру та реалізовано функціональні можливості. Особлива увага буде приділена питанням забезпечення безпеки даних, захисту конфіденційної інформації та зручності використання системи для кінцевих користувачів.

Таким чином, розробка веб-сайту для віддаленого ведення кадрового аудиту з використанням C# та .NET на сьогодні є дуже актуальним та важливим завданням, яке відповідає сучасним тенденціям розвитку інформаційних технологій та потребам бізнесу, особливо під час війни. Виконання цієї роботи дозволить зробити вагомий внесок у підвищення ефективності управління персоналом, забезпечення конкурентоспроможності компаній в умовах переходу все у цифру та цифрової економіки, а також збереження та безпеку даних у складних умовах повномасштабної війни.

1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Поняття кадрового аудиту, контролю кадрів

Контроль кадрів та кадровий аудит - це два важливих процеси, які допомагають компаніям організувати та оптимізувати свою кадрову роботу. Обидва процеси мають спільну мету: забезпечити відповідність кадрових процесів, політик та практик законодавству, стратегії розвитку компанії та загальноприйнятим стандартам управління персоналом.

Контроль кадрів[1] - це постійний процес, який включає в себе моніторинг, аналіз та оцінку кадрових процесів, політик та практик. Він дозволяє виявляти проблеми та недоліки на ранніх стадіях та вживати заходів для їх усунення.

Кадровий аудит[2] - це більш глибоке та комплексне дослідження кадрової роботи, яке проводиться періодично або за потреби. Він включає в себе оцінку всіх аспектів кадрової роботи, від набору та звільнення персоналу до навчання та розвитку.

Для проведення контролю кадрів та кадрового аудиту існує декілька методів:

- перевірка відповідності кадрових документів (наказів, положень, інструкцій тощо) чинному законодавству та внутрішнім нормативним актам компанії.
- вивчення даних про плинність кадрів, відвідуваність, продуктивність праці тощо.
- оцінка того, як кадрова служба виконує свої функції.

Особливо важливим кадровий аудит та контроль кадрів стають в умовах кризових ситуацій, таких як військові конфлікти, коли є ризик втрати паперової документації. В таких умовах цифровізація кадрових процесів, зокрема перехід на електронний документообіг та використання хмарних сервісів для зберігання даних, дозволяє забезпечити безперервність роботи та збереження важливої інформації. Що в першу чергу буде корисно впливати на економіку країни.

1.2 Аналіз програмних рішень для предметної області

Існує безліч способів організувати кадровий облік, які можна узагальнити наступними категоріями :

- планування подумки - найпростіший метод, який не потребує жодних інструментів, але може бути незручним для відстеження великої кількості інформації.
- На папері - традиційний спосіб, який включає в себе різноманітні форми, журнали та таблиці.
- Програми та веб-додатки такі як: Google Sheets, Excel та інші програми, які пропонують зручні функції для організації та аналізу даних.

До паперових засобів ведення кадрового обліку, аудиту можна віднести “Особову карту П2, А4”[3], яка спеціально розроблена за спеціальною структурою для ведення кадрового обліку, так включає майже все що треба робітнику діловодства, для того щоб розуміти що це за працівник, освіта, ім’я та інша важлива інформація.

Перейдемо до структури більш ближче:

- 1) Загальні відомості які включають у себе ПІБ, Дату та місце народження, Освіту, спеціальність, професію стать та сімейний стан, а також адресу та іншу потрібну інформацію
- 2) Відомості про прийняття на роботу такі як: Посада, дата прийому, підрозділ, департамент, оклад та інші.
- 3) Відомості про трудову діяльність, а саме: Зміна розміру окладу, надання надбавок, дисциплінарні стягнення, відпустки, командировки та іншу інформацію
- 4) Відомості про військовий облік, які включають у себе категорію військової придатності, військове звання, відомості про військовий квиток
- 5) Відомості про звільнення з роботи: дата звільнення. Підстава

звільнення, наказ про звільнення.

б) Та інші додаткові відомості які потрібні діловодству.

Особова картка заповнюється на підставі документів які були надані працівником, та зберігається протягом 75 днів після звільнення працівника.

Ця структура є типовою, але деякі підприємства можуть вносити до неї зміни відповідно до потреб які можуть бути надані керівництвом

Типова форма N П-2

ЗАТВЕРДЖЕНО
наказом Держкомстату та Міністерства оборони
України
віа 25 грудня 2009 р. N 495/656

Найменування підприємства (установи, організації)
Код ЄДРПОУ _____

Дата заповнення	Табельний номер	Індивідуальний ідентифікаційний номер	Стать (чоловіча, жіноча)	Вид роботи (основна, за сумісництвом)	
					місце для фотокартки

ОСОБОВА КАРТКА ПРАЦІВНИКА

I. ЗАГАЛЬНІ ВІДОМОСТІ

1. Прізвище _____ Ім'я _____ По батькові _____

2. Дата народження "___" ____ 19__ р. 3. Громадянство _____

4. Освіта (базова загальна середня, повна загальна середня, професійно-технічна, вища загальна, вища спеціальна, вища базова, повна вища)

Назва освітнього закладу	Диплом (свідчення), серія, номер	Рік закінчення
Спеціальність (професія) за дипломом (свідченням)	Кваліфікація за дипломом (свідченням)	Форма навчання (денна, вечірня, заочна)

5. Післядипломна професійна підготовка: навчання в ☐ аспірантури ☐ аспірантури ☐ докторантури (необхідне відмітити х)

Назва освітнього, наукового закладу	Диплом, номер, дата видачі	Рік закінчення	Науковий ступінь, учене звання

6. Останнє місце роботи _____ посада (професія) _____

7. Стаж роботи: ставом на "___" ____ 20__ р. Загальний _____ днів _____ місяців _____ років, що дає право на надбавку за вислугу років _____ днів _____ місяців _____ років

8. Дата та причина звільнення (скорочення штатів; за власним бажанням, за прогул та інші порушення, невідповідність займаній посаді тощо) "___" ____ 20__ р. _____

9. Відомості про отримання пенсій (у разі наявності вказати вид пенсійних виплат згідно з чинним законодавством)

10. Родинний стан _____

Ступінь родинного зв'язку (склад сім'ї)	П. І. Б.	Рік народження

11. Місце фактичного проживання (область, місто, район, вулиця, N будинку, квартири, номер контактного телефону, поштовий індекс) _____

12. Місце проживання за державною реєстрацією _____ Паспорт: серія _____ N _____, ким виданий _____, дата видачі _____

II. ВІДОМОСТІ ПРО ВІЙСЬКОВИЙ ОБЛІК

Група обліку _____ Придатність до військової служби _____

Категорія обліку _____ Назва райвійськкомату за місцем реєстрації _____

Склад _____ Назва райвійськкомату за місцем фактичного проживання _____

Військове звання _____ Перебування на спеціальній обліку _____

Військово-облікова спеціальність N _____

Рис. 1.1 Приклад структури Особової картки П2, у форматі "А4"

Переваги використання паперових документів для контролю кадрів:

- Простота не потребують спеціальних знань чи навичок, доступні для будь-якого роботодавця.

- Зрозумілість: інформація на папері легко читається та розуміється.

- Збереження: паперові документи слугують надійним доказом кадрових рішень протягом тривалого часу.

паперові документи можна зберігати в безпечному місці, обмежуючи доступ до них сторонніх осіб.

не потребують спеціального обладнання чи програмного забезпечення.

дозволяють створювати власні форми та журнали відповідно до потреб підприємства.

Однак, паперові методи мають й недоліки:

- пошук та обробка інформації на папері може бути трудомістким.

- ручне введення даних може призвести до помилок.

- паперові документи можуть бути втрачені або пошкоджені.

паперові методи не дозволяють проводити складний аналіз кадрових даних.

- інформація часто дублюється в різних документах.

- не підходять для великих підприємств з великою кількістю працівників.

В цілому, паперові методи кадрового обліку здатні бути ефективними для малих підприємств з невеликою кількістю працівників. Проте, для більших компаній або тих, що потребують більш детального аналізу даних, всеж таки краще використовувати електронні таблиці або спеціалізовані програмні комплекси.

Таблиці Excel [5] є набагато зручнішим способом організації обліку кадрів та контролю кадрів. Приклад таблиці для ведення кадрового контролю наведено на рисунку 1.2.

	A	B	C	D	E	F	G	H	I	J	K
1	Посада	Оклад									
2	Директор	7000									
3	Бухгалтер	3000									
4	Начальник IT відділу	5000									
5	Інженер	4000									
6											
7											

Рис. 1.2 Приклад таблиці Excel для ведення кадрового аудиту

Переваги електронних таблиць Excel:

- простий інтерфейс взаємодії з додатком, оскільки щоденник початково має вигляд таблиці;
- можливість легко створювати користувацькі шаблони щоденників обліку витрат та доходів та використовувати ці шаблони багатократно;
- можливість автоматизувати розрахунки із використанням простих та складних формул Excel;
- можливість спрощення введення даних в щоденник за допомогою графічних форм;
- можливість організовувати списки підстановки та автоматичні підстановки в комірках таблиць, що спрощує введення даних та зменшує кількість помилок при введенні однакових назв статей витрат чи доходів;
- наявність засобів пошуку, фільтрації та сортування;
- наявність засобів для формування статистичних звітів як в чисельному так і в графічному вигляді (статистичні формули, зведені таблиці, діаграми тощо).

Однак, таблиці Excel мають і суттєві недоліки, які ускладнюють їх використання для задач обліку та контролю кадрів та їх аудиту, серед яких можна відзначити такі:

- створення якісних шаблонів таблиць вимагає певного досвіду від користувача;
- з часом таблиці розростаються і робота з ними ускладнюється;
- для формування даних за новий період необхідно постійно копіювати формули та шаблони;
- для автоматизації формування підсумкових звітних даних необхідно писати макроси або модулі;
- дані в таблиці доводиться вводити уручну, що може призвести до пропуску позицій;
- відсутні засоби структурування списків за категоріями;
- засоби захисту даних в Excel обмежені, тому не можуть виключити сторонній доступ до них.

Більш зручним засобом для обліку та контролю кадрів, кадрового аудиту є он-лайн таблиці. Такі як Google Sheets[6], розглянемо детальніше.

Shirt Sales Information									
				Shirt 1			Shirt 2		
Sale ID	Date	Point of Sale	Amount	Size	Design	Type	Size	Design	Type
1	12/1/2023	Online	4	L	Design 5 - Graffiti	V-neck	L	Design 2 - Sunset	Crewneck
2	12/2/2023	Michaela Ho	1	M	Design 4 - Skyline	V-neck			
3	12/2/2023	Online	5	M	Design 1 - Waves	V-neck	M	Design 2 - Sunset	Long-sleeved
4	12/3/2023	Online	2	M	Design 5 - Graffiti	V-neck	L	Design 3 - Boat	Sweatshirt
5	12/4/2023	Online	3	M	Design 3 - Boat	Long-sleeved	M	Design 5 - Graffiti	V-neck
6	12/6/2023	Natasha Young	3	M	Design 5 - Graffiti	Tank Top	L	Design 6 - Midtown	Long-sleeved
7	12/7/2023	Daniela Rosas	1	L	Design 1 - Waves	V-neck	M	Design 5 - Graffiti	Sweatshirt
8	12/8/2023	Daniela Rosas	4	M	Design 2 - Sunset	V-neck	M	Design 5 - Graffiti	V-neck
9	12/9/2023	Michaela Ho	5	S	Design 2 - Sunset	V-neck	S	Design 1 - Waves	V-neck
10	12/10/2023	Chastity Smith	2	XXL	Design 2 - Sunset	V-neck	S	Design 1 - Waves	Sweatshirt
11	12/10/2023	Gabriel Medina	1	L	Design 1 - Waves	V-neck			
12	12/11/2023	Gabriel Medina	2	M	Design 2 - Sunset	Long-sleeved	L	Design 1 - Waves	Sweatshirt
13	12/12/2023	Stephanie Gilmore	1	M	Design 2 - Sunset	Crewneck			
14	12/12/2023	Online	3	S	Design 2 - Sunset	V-neck	M	Design 4 - Skyline	Tank Top
15	12/13/2023	Online	1	XS	Design 1 - Waves	Crewneck			
16	12/14/2023	Michaela Ho	2	XL	Design 5 - Graffiti	V-neck	XL	Design 4 - Skyline	V-neck
17	12/14/2023	Gabriel Medina	1	XL	Design 1 - Waves	Tank Top			
18	12/16/2023	Michaela Ho	1	L	Design 3 - Boat	Crewneck			

Рис. 1.3 Приклад таблиці Google sheets

Google Sheets пропонує низку переваг, роблячи його зручним інструментом для ведення обліку:

- доступ до таблиць з будь-якого місця та з будь-якого пристрою.
- спільна робота над таблицями з іншими людьми в режимі реального часу.
- не потрібно турбуватися про збереження змін - вони відбуваються автоматично.
- легке поєднання з Gmail, Google Calendar та іншими інструментами Google.
- безкоштовне використання для особистих потреб.

Поміж переваг у “Excel” від гугла є деякі недоліки, вони не дуже великі та проблемні, але все ж таки вони існують і їх треба рахувати, їх перелік нижче

- порівняно з Excel, деякі складні функції та можливості створення макросів відсутні.
- робота з таблицями неможлива без підключення до мережі.
- ваші таблиці зберігаються на серверах Google, що може викликати питання щодо конфіденційності.
- складність автоматизації складних завдань за допомогою макросів.

Google Sheets чудово підходить для спільного ведення обліку, доступу до таблиць з будь-якого місця та базових операцій з даними.

Але є і велика проблема, доступ до даних неможливо буде отримати при відсутності інтернету – що може бути великою проблемою для деяких організацій та діловодств. Через те що деякі компанії суворо забороняють підключати компютери до мережі, для запобігання витоку інформації.

Перейдемо до наступного десктопного додатку, це буде LibreOffice Calc[7] - це безкоштовний та відкритий програмний комплекс для роботи з електронними таблицями, який пропонує широкий спектр функцій для ведення обліку, аналізу

даних та створення звітів. Calc є частиною LibreOffice, офісного пакету, що також включає текстовий процесор Writer, презентаційний редактор Impress, систему керування базами даних Base та інші інструменти що можуть стати у нагоді при проведенні контролю кадрів та кадрового аудиту.

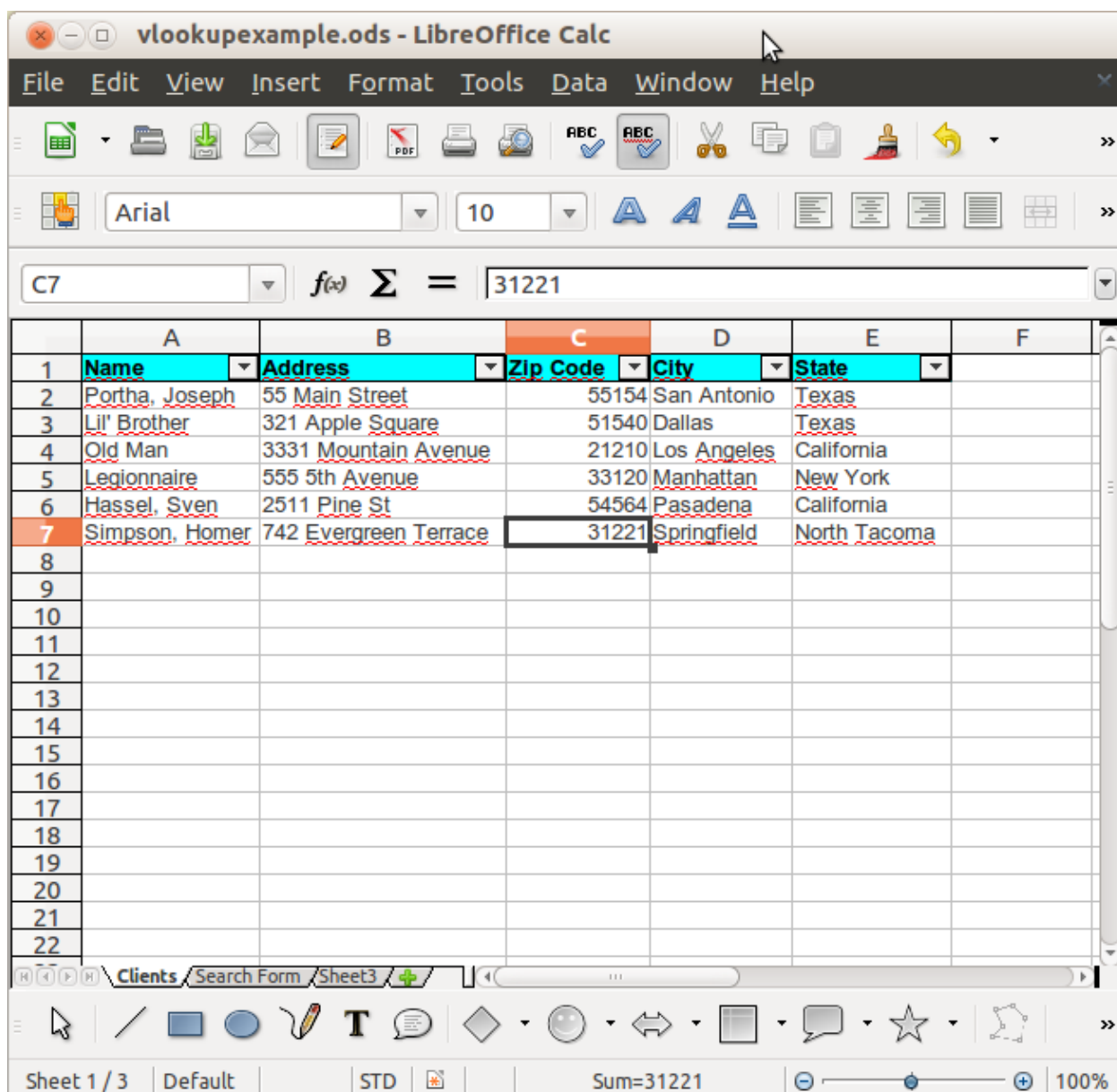


Рис. 1.4 Приклад таблиці LibreOffice calc

Інтерфейс LibreOffice декілько відрізняється від інших програм, але це можна додати до особливостей програми, давайте перейдемо до плюсів та мінусів. З плюсів можна чётко окреслити цілий перелік:

- можна використовувати безкоштовно для будь-яких цілей, а його код доступний для публічного перегляду та модифікації. Це робить його привабливим вибором для користувачів, які шукають доступні та прозорі рішення.
- доступність на різних операційних системах, включаючи Windows, macOS та Linux. Це забезпечує гнучкість та зручність використання для користувачів, які працюють у різних середовищах.
- може відкривати та зберігати файли у форматах Excel, CSV та інших популярних форматах. Це гарантує, що ви зможете легко обмінюватися своїми таблицями з іншими людьми, незалежно від того, яке програмне забезпечення вони використовують.
- Calc пропонує безліч функцій для роботи з даними, включаючи формули, діаграми, таблиці даних та багато іншого. Це робить його потужним інструментом для аналізу даних, створення звітів та візуалізації інформації.
- ПЗ дозволяє писати макроси для автоматизації завдань. Це може значно заощадити час та спростити повторювані дії.
- можна розширити за допомогою плагінів, які додають нові функції та можливості. Це дозволяє користувачам налаштувати Calc відповідно до своїх потреб та вподобань.

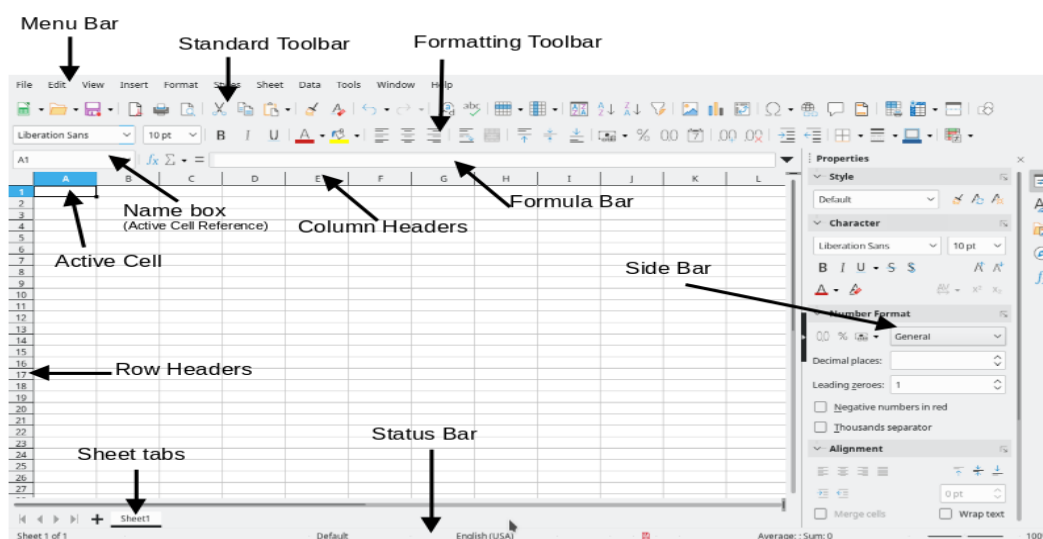


Рис. 1.5 Пояснення до інтерфейсу LibreOffice Calc

Тепер перейдемо до недоліків Calc'a, а саме з основних можна підкреслити:

- Інтерфейс може здатися трохи застарілим та складним для нових користувачів порівняно з Excel та Google Sheets. Це може ускладнити навчання та роботу з програмою.

- може бути трохи повільнішим за Excel, особливо при роботі з великими наборами даних. Це може бути проблемою для користувачів, яким потрібна швидка та чуйна робота з таблицями.

- не пропонує вбудованих функцій для онлайн-співпраці над таблицями, як це роблять Google Sheets. Це може ускладнити спільну роботу над проектами з іншими людьми.

- може не мати деяких функцій, які доступні в Excel, таких як розширені можливості аналізу даних та інструменти для візуалізації. Це може бути недоліком для користувачів, яким потрібна більш просунута функціональність.

З цього всього можна зробити висновок що Calc це потужний та безкоштовний інструмент для роботи з електронними таблицями, який може стати чудовою альтернативою Excel та Google Sheets для багатьох користувачів. Calc пропонує широкий спектр функцій, сумісність з форматами файлів та можливість розширення за допомогою плагінів.

Однак, Calc може здатися трохи складним для нових користувачів, а також поступатися Excel за швидкістю роботи та Google Sheets за можливостями онлайн-співпраці.

Зведені результати аналізу характеристик розглянутих додатків наведено у таблиці 1.1.

Таблиця 1.1

Зведені результати аналізу характеристик додатків для контролю кадрів,
кадрового аудиту

Показник	Microsoft Excel	Google Sheets	Libre Office Calc
Платформи	MacOs, Windows	Android, iOS, Web	Android, iOS, Web
Можливість спільного редагування в режимі реального часу	-	+	-
Зберігання на віддаленому сервері	+	+	+
Offline доступ	+	-	+
Підтримка мобільних пристроїв	-	+	+
Інтеграція з іншими продуктами	+	+	+
Інтеграція з іншими продуктами	+	+	+

1.3 Формулювання вимог до ПЗ

Вивчення процесу контролю кадрів, а також кадрового аудиту та існуючих інструментів для управління та обліку кадрів є важливим етапом у розробці програмного забезпечення, що відповідає сучасним вимогам бізнесу. Кадровий контроль та аудит дозволяють не лише виявляти та усувати недоліки у роботі з персоналом, але й сприяти розвитку ефективних стратегій управління людськими ресурсами. Проведений аналіз існуючих підходів та інструментів допомагає сформулювати чіткі та обґрунтовані вимоги до майбутнього програмного продукту.

ПЗ для контролю кадрів повинно бути адаптовано для використання у

дистанційному режимі, що особливо актуально в умовах сучасного бізнес-середовища, де віддалена робота стає все більш поширеною. Відповідно, вона має забезпечувати високий рівень доступності та зручності користування, дозволяючи менеджерам та співробітникам виконувати свої завдання незалежно від їхнього місцезнаходження.

Основні можливості, які повинна надавати програма для контролю кадрів, це:

- Можливість внесення в систему інформації про людину для ведення віддаленого кадрового аудиту.
- Можливість редагування внесеної інформації про людину для ведення віддаленого кадрового аудиту.
- Можливість видалення інформації про людину з бази даних.
- Можливість перегляду інформації про людину.
- Можливість перевірки відповідності до вимог Українського законодавства даних працівника: віку, наявності військово-облікового документа, відповідності посади освіти (ECTS), відповідності ступеню освіти до наявних ECTS кредитів.
- Система реєстрації та авторизації.

2. ВИБІР ЗАСОБІВ ТА БІБЛІОТЕК ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

2.1 Середовище розробки

Вибір середовища розробки є ключовим етапом у процесі створення програмного забезпечення, оскільки воно визначає ефективність роботи розробника, зручність налагодження та тестування програми, а також забезпечує підтримку необхідних бібліотек та інструментів. Для розробки бакалаврського програмного продукту було обрано Visual Studio – одне з найпотужніших і найпоширеніших середовищ розробки, особливо для роботи з мовами програмування C# та платформою .NET.

Visual Studio пропонує широкий спектр можливостей для розробників. Серед основних переваг цього середовища:

- зручне редагування коду з підсвічуванням синтаксису, автодоповненням та швидким переходом до визначень. Це значно підвищує продуктивність розробників і зменшує кількість помилок у коді.
- надає потужні інструменти для налагодження програм, включаючи можливість покрокового виконання коду, встановлення точок зупину (breakpoints), перегляд змінних та стеку викликів. Це дозволяє ефективно виявляти та виправляти помилки.
- має вбудовані інструменти для створення та виконання модульних тестів, що допомагає забезпечити високу якість програмного продукту. Інтеграція з фреймворками для тестування, такими як MSTest, NUnit та xUnit, дозволяє легко писати та запускати тести безпосередньо з середовища розробки.

Інтеграція з системами контролю версій: Visual Studio підтримує інтеграцію з популярними системами контролю версій, такими як Git та Azure DevOps. Це дозволяє ефективно керувати змінами у коді, відстежувати історію змін та

співпрацювати з іншими розробниками.

- Середовище підтримує велику кількість розширень та плагінів, які додають нові функціональні можливості та покращують зручність роботи. Розширення можна легко встановити через вбудований менеджер розширень.

- дозволяє створювати різноманітні типи проектів, включаючи веб-додатки, мобільні додатки, десктопні програми та багато інших. Це робить середовище універсальним інструментом для розробників, що працюють у різних галузях.

- пропонує інтеграцію з хмарними сервісами Azure, що дозволяє легко розгортати та керувати додатками у хмарі. Це забезпечує масштабованість та високу доступність розроблюваних рішень.

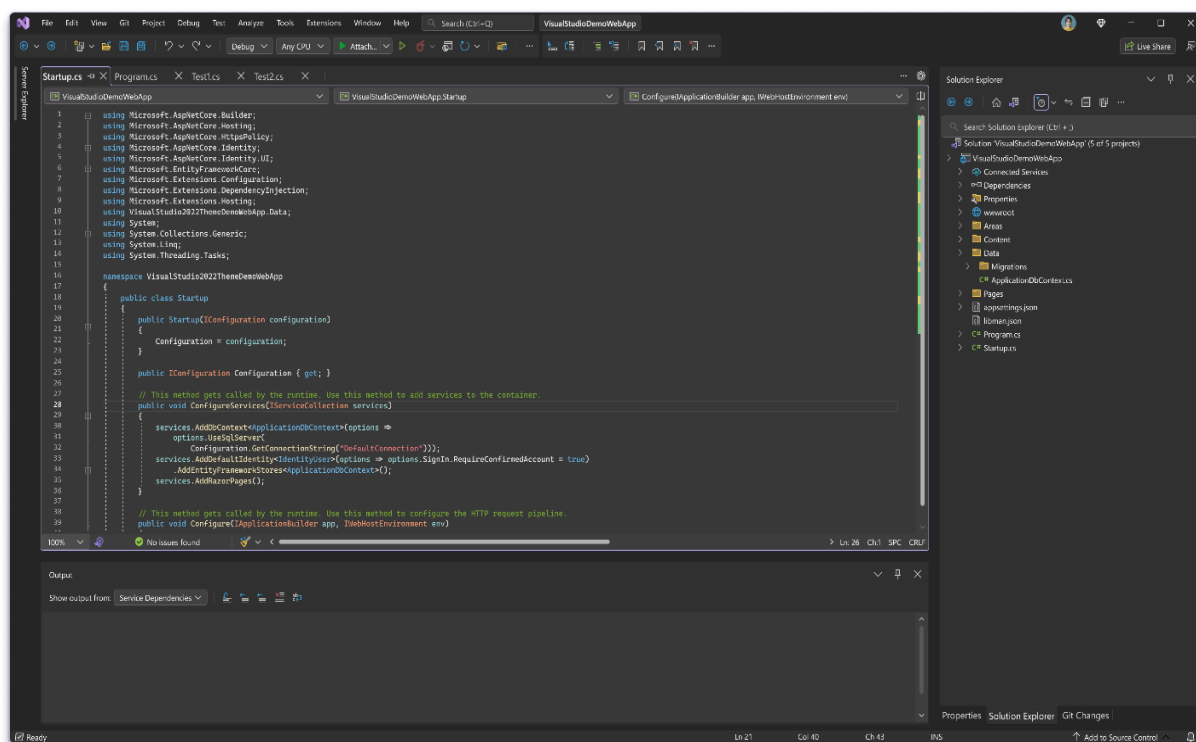


Рис. 2.1 Visual Studio User Interface

Окрім Visual Studio, для розробки нашого програмного продукту також використовуються інші інструменти та бібліотеки, що входять до екосистеми .NET. Це, зокрема, Entity Framework для роботи з базами даних, ASP.NET Core для розробки веб-додатків та Blazor для створення інтерактивних інтерфейсів користувача. Ці технології забезпечують високу продуктивність, надійність та

зручність розробки, що є критично важливими для успішного завершення проекту.

Таким чином, вибір Visual Studio як основного середовища розробки, у поєднанні з іншими інструментами та бібліотеками .NET, дозволяє забезпечити ефективну розробку якісного та масштабованого програмного продукту, що відповідає сучасним вимогам бізнесу та потребам користувачів.

2.2 Мова програмування та платформа

Вибір мови програмування та платформи для розробки програмного продукту є одним з найважливіших етапів у процесі створення програмного забезпечення. Для бакалаврської роботи було обрано мову програмування C# та платформу .NET, зокрема також був використований фреймворк ASP.NET для розробки веб-додатків. Ці технології були обрані через їхні численні переваги, які забезпечують високу продуктивність, надійність та масштабованість розроблюваних рішень.

Мова програмування C#[8] - є однією з найпоширеніших мов програмування, розробленою корпорацією Microsoft. Вона пропонує безліч переваг для розробників:

- Синтаксис C# є зрозумілим та логічним, що робить цю мову легкою для вивчення та використання. Це сприяє швидкому розвитку та зменшенню кількості помилок у коді.

- підтримує принципи ООП, такі як спадковість, інкапсуляція та поліморфізм. Це дозволяє створювати модульний, зрозумілий та легко підтримуваний код.

- пропонує багату стандартну бібліотеку, яка містить численні готові до використання класи та методи. Це дозволяє розробникам швидко вирішувати стандартні завдання та зосереджуватися на специфічних вимогах проекту.

Платформа .NET є неймовірно потужним інструментом для створення різноманітних програмних рішень, включаючи веб-додатки, десктопні програми, мобільні додатки та інші. Вона забезпечує численні переваги:

- Завдяки .NET Core, розробники можуть створювати додатки, що працюють на різних операційних системах, таких як Windows, macOS та Linux. Це значно розширює можливості використання програмного продукту.

- підтримує кілька мов програмування, включаючи C#, F# та Visual Basic. Це дозволяє командам розробників використовувати найбільш зручну та ефективну мову для конкретних завдань.

- забезпечує високий рівень безпеки завдяки вбудованим механізмам управління пам'яттю та захисту від поширених вразливостей. Крім того, платформа оптимізована для високої продуктивності, що дозволяє створювати швидкі та надійні додатки.

ASP.NET є фреймворком для створення веб-додатків на базі платформи .NET. Він надає всі необхідні інструменти для розробки сучасних, масштабованих та безпечних веб-додатків на ASP.NET:

- підтримує модель MVC (Model-View-Controller), яка сприяє розділенню логіки додатку на три частини: модель, уявлення та контролер. Це забезпечує чітку структуру проекту, що полегшує його підтримку та розвиток.

- легко інтегрується з іншими технологіями Microsoft, такими як бібліотека Entity Framework для роботи з базами даних, а також з хмарними сервісами Azure. Це дозволяє створювати комплексні рішення, що включають зберігання даних та хмарні обчислення.

- має вбудовані засоби захисту від поширених веб-атак, таких як SQL-ін'єкції та міжсайтовий скриптинг (XSS). Це забезпечує високий рівень безпеки веб-додатків.

Останнім часом все більше і більше компаній переходять[4] на ASP.NET для розробки своїх продуктів. За результатом усього цього, можна зробити висновок що платформа “ASP.NET”, та мова програмування “C#” підійдуть для написання програмного забезпечення для кадрового аудиту, контролю кадрів.

2.3 Бібліотеки

Для розробки ефективного та функціонального програмного продукту важливо обрати відповідні бібліотеки, які дозволять спростити процес розробки, забезпечити надійну роботу з даними та створити безпечну систему управління користувачами. Тому що як правило 20% зусиль дають 80% результату[9], це твердження яке колись сформував Парето. У бакалаврській роботі було прийнято рішення використовувати дві ключові бібліотеки: Entity Framework та ASP.NET Core Identity. Ці бібліотеки обрані через їхні потужні можливості та інтеграцію з платформою .NET.

Entity Framework (EF) є ORM (Object-Relational Mapping) бібліотекою[10], яка дозволяє розробникам працювати з базами даних, використовуючи об'єктно-орієнтований підхід. Це значно спрощує процес розробки, оскільки дозволяє уникнути написання тон складного SQL-коду. Основні переваги використання Entity Framework це те що:

- EF дозволяє розробникам працювати з базами даних, використовуючи знайомий об'єктно-орієнтований підхід. Це зменшує криву навчання та дозволяє швидко приступити до розробки.
- Завдяки можливості автоматичного створення та оновлення баз даних, розробники можуть зосередитися на бізнес-логіці додатку, залишаючи всі технічні деталі на EF. Це забезпечує швидкість та зручність у роботі.
- EF інтегрується з LINQ (Language Integrated Query), що дозволяє писати запити до баз даних, використовуючи синтаксис C#. Це підвищує читабельність та зручність написання коду.
- EF підтримує міграції баз даних, що дозволяє легко керувати змінами в структурі баз даних у процесі розробки. Це забезпечує безперебійну роботу додатка та зручність оновлення.

ASP.NET Core Identity є бібліотекою для управління ідентифікацією та авторизацією користувачів у додатках на платформі ASP.NET Core. Вона надає

широкий спектр функцій для створення безпечної системи управління користувачами:

- ASP.NET Core Identity надає готові механізми для створення, редагування та видалення облікових записів користувачів. Це спрощує процес розробки та забезпечує надійну роботу з даними користувачів.

- Бібліотека підтримує різні методи аутентифікації, включаючи форми, токени, OAuth та інші. Вона також забезпечує гнучкі механізми управління правами доступу, що дозволяє налаштовувати доступ до різних частин додатка відповідно до ролей та прав користувачів.

- ASP.NET Core Identity включає в себе сучасні засоби захисту, такі як шифрування паролів, захист від brute-force атак, та двофакторну аутентифікацію. Це забезпечує високий рівень безпеки додатка.

- Бібліотека легко інтегрується з іншими сервісами та платформами, що дозволяє створювати комплексні рішення з використанням сторонніх сервісів для аутентифікації, таких як Google, Facebook, та Microsoft.

Комбінація використання Entity Framework та ASP.NET Core Identity забезпечує неймовірно потужну та дуже гнучку основу для розробки веб-додатку. Entity Framework забезпечує зручний та неймовірно ефективний спосіб роботи з базами даних, тоді як ASP.NET Core Identity надає всі необхідні інструменти для управління користувачами та їхньою аутентифікацією. Це дозволяє зосередитися на створенні функціональних можливостей додатка, не витрачаючи багато часу на розробку базових функцій.

Окрім вибору бібліотек для бекенду, важливо приділити увагу зовнішньому вигляду та зручності користування веб-додатком. Для цього ми використовуємо CSS (Cascading Style Sheets) та фреймворк Bootstrap. CSS дозволяє визначати стилі елементів на веб-сторінці, забезпечуючи гнучкість у налаштуванні дизайну та зручність користування.

Bootstrap, розроблений компанією Twitter, є одним з найпопулярніших CSS-фреймворків, що значно спрощує розробку адаптивних веб-додатків. Він надає

готові компоненти та стилі, що дозволяють швидко створювати сучасні та зручні інтерфейси:

- Bootstrap забезпечує адаптивність веб-сторінок, що дозволяє додатку коректно відображатися на різних пристроях, включаючи смартфони, планшети та десктопи.
- включає широкий набір готових компонентів, таких як кнопки, форми, модальні вікна, навігаційні панелі та інші елементи, що значно скорочує час на розробку інтерфейсу.
- Завдяки простоті використання та гарній документації, розробники можуть швидко інтегрувати Bootstrap у свій проект та створювати привабливі та функціональні інтерфейси.

Використання CSS та Bootstrap у поєднанні з Entity Framework та ASP.NET Core Identity забезпечує створення потужного, надійного та привабливого програмного продукту, що відповідає всім сучасним вимогам та очікуванням користувачів.

2.4 База даних

Одним із ключових аспектів розробки програмного продукту є вибір бази даних, яка забезпечить надійне зберігання та управління даними. Для бакалаврського проекту було обрано Microsoft SQL Server[11] (MSSQL), оскільки він надає неймовірно потужні можливості для роботи з даними та легко інтегрується з іншими компонентами екосистеми Microsoft, такими як Azure. Використання MSSQL у поєднанні з середовищем розробки Visual Studio створює дуже гарну синергію та надає зручний інструментарій для розробників.

Microsoft SQL Server (MSSQL) є одним з найпопулярніших реляційних систем управління базами даних (RDBMS), який пропонує широкий набір функцій для управління даними, високої продуктивності та безпеки. Основні переваги використання MSSQL:

- забезпечує високу продуктивність та надійність роботи з великими

обсягами даних. Це дозволяє ефективно обробляти запити, забезпечуючи швидку відповідь та високу доступність даних.

- надає потужні засоби для захисту даних, включаючи шифрування, управління правами доступу та засоби для виявлення та запобігання загрозам безпеці.

- має відмінну інтеграцію з платформою .NET, що спрощує розробку додатків на базі технологій Microsoft. Це дозволяє легко використовувати MSSQL у проектах, створених за допомогою Visual Studio та інших інструментів .NET.

- підтримує різні рівні масштабування, що дозволяє адаптувати базу даних під зростаючі потреби бізнесу. Це забезпечує гнучкість у використанні ресурсу та зростанні додатку.

Однією з важливих переваг використання MSSQL є його висока інтеграція з хмарними сервісами, що забезпечує легке розгортання та масштабування додатків. Після виходу продукту на ринок, його можна буде задеплоїти на будь-якому хмарному сховищі, такому як Azure, AWS або Google Cloud, завдяки гарній інтеграції MSSQL з цими платформами. Це забезпечує наступні переваги:

- Розробники можуть вибирати будь-яку хмарну платформу для розгортання додатку, що дозволяє оптимізувати витрати та підвищити ефективність.

- Хмарні платформи забезпечують автоматичне масштабування ресурсів, що дозволяє додатку адаптуватися до змінних навантажень та забезпечує високу доступність.

- Використання хмарних сховищ дозволяє забезпечити безперервну роботу додатку, завдяки високій надійності та відмовостійкості хмарних сервісів.

З цього можна зробити висновок, що вибір MSSQL у поєднанні з Visual Studio забезпечує дуже гарну основу для розробки якісного та надійного програмного продукту. Завдяки високій інтеграції з хмарними сервісами, веб-додаток буде готовий до масштабування та розгортання на будь-якій популярній хмарній платформі, або на локальному сервері, що дозволить забезпечити його стабільну та ефективну роботу після виходу на ринок.

3 ПРОЕКТУВАННЯ, РОЗРОБКА, ТА ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Архітектура веб-додатку

Так як у нас на меті стоїть розробити додаток для контролю кадрів, кадрового аудиту треба проаналізувати які типи додатків загалом роблять [12].

- Веб-сайт - Це найпростіший тип веб-додатку, який складається з статичних сторінок HTML, CSS та JavaScript. Вони використовуються для публікації інформації, презентації продуктів або послуг, а також для зв'язку з користувачами;

- Веб-портал - Це складніші веб-додатки, які пропонують широкий спектр функцій, таких як персоналізація, автентифікація, авторизація, управління контентом, форуми, чати та багато іншого. Вони використовуються для надання користувачам доступу до персоналізованої інформації та послуг.

- Веб-додатки на одній сторінці (SPA) - Ці веб-додатки завантажують лише одну HTML-сторінку, а потім динамічно оновлюють її вміст за допомогою JavaScript. Це робить їх більш швидкими та чутливими, ніж традиційні веб-додатки з декількома сторінками.

- Веб-служби - Це програмні інтерфейси (API), які дозволяють іншим веб-додаткам отримувати доступ до даних та функціональності. Вони використовуються для інтеграції веб-додатків та створення складних розподілених систем;

- Прогресивні веб-додатки - Це веб-додатки, які поєднують в собі функції веб-сайтів та нативних мобільних додатків. Вони можуть працювати в автономному режимі, надсилати push-повідомлення та встановлюватися на головний екран мобільного пристрою.

У результаті розгляду усього цього, було прийнято рішення що бакалаврська робота буду – веб-додатком.

При проектуванні веб-додатку надзвичайно важливо правильно обрати архітектурний підхід, адже від цього залежить ефективність, масштабованість та зручність підтримки програмного продукту. У бакалаврському проекті ми використовуємо модель розподіленої архітектури, яка включає фронтенд та бекенд компоненти. Основні технології, які ми застосовуємо, це Razor Pages для фронтенду та ASP.NET для бекенду.

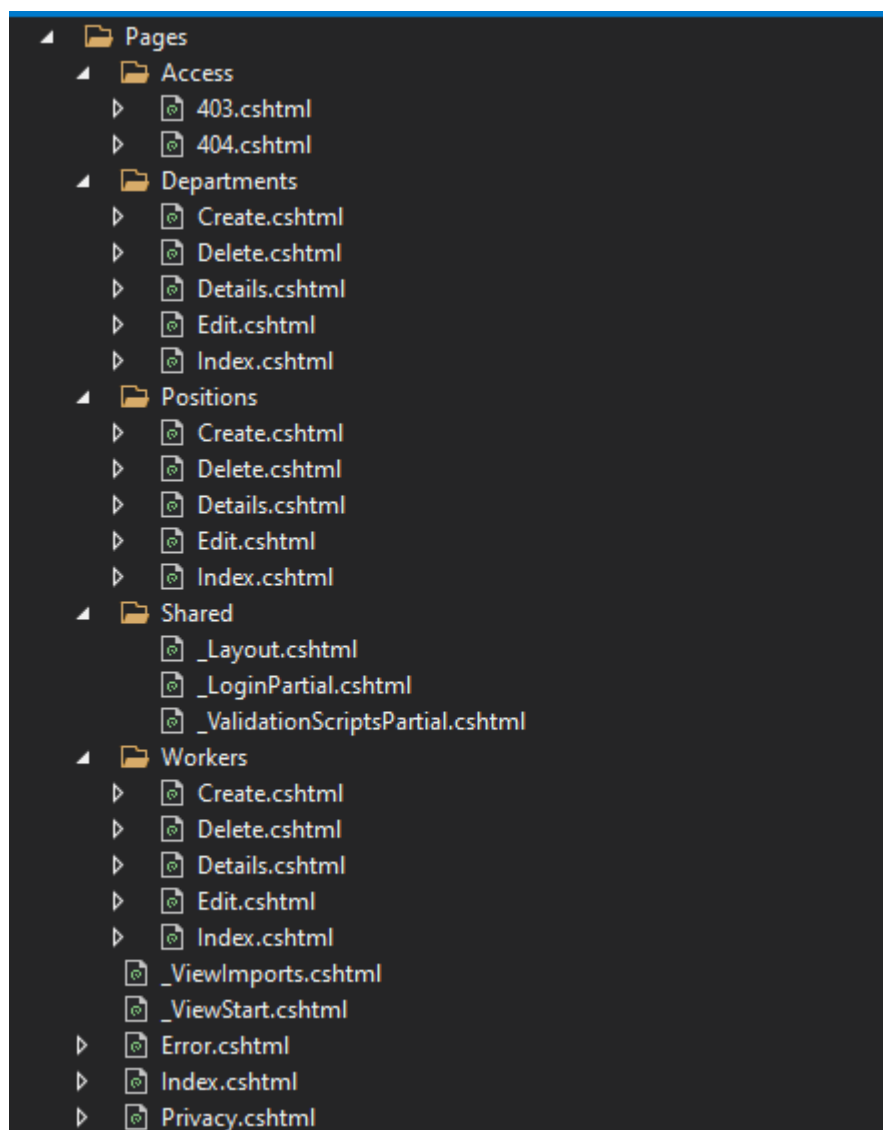


Рис. 3.1 Front-End архітектура на базі Razor-Pages

Ось опис кожної папки на Рис. 3.1:

Access: Ця папка містить сторінки Razor Pages для доступу до помилок, таких як 403 (Немає доступу) та 404 (Не знайдено).

Departments: Ця папка містить сторінки Razor Pages для управління відділами.

Positions: Ця папка містить сторінки Razor Pages для управління посадами.

Shared: Ця папка містить загальні сторінки Razor Pages, які використовуються в усьому проекті.

Workers: Ця папка містить сторінки Razor Pages для управління працівниками.

Razor Pages [13] є частиною ASP.NET Core, яка дозволяє створювати динамічні веб-сторінки з використанням простого та інтуїтивно зрозумілого синтаксису. Основні переваги Razor Pages:

- Razor Pages спрощують структуру додатку, об'єднуючи модель, контролер та подання в одному файлі. Це робить код більш зрозумілим та зручним для підтримки. Завдяки використанню синтаксису Razor, можна легко впроваджувати логіку відображення даних без необхідності писати великий обсяг коду. Це зменшує кількість помилок та підвищує ефективність розробки. Також Razor Pages інтегруються з ASP.NET Core, що дозволяє легко використовувати всі можливості платформи, включаючи засоби безпеки, управління станом та інші. Окрім всього що було описано вище, Razor Pages дозволяють створювати часткові подання (partial views) та компоненти, які можна повторно використовувати в різних частинах додатку, що спрощує підтримку та розвиток проекту. Це дозволяє зосередитись на написанні унікальної логіки, використовуючи вже готові частини для часто використовуваних елементів.

За допомогою візуального моделювання можна зробити висновки що буде доступно користувачам.

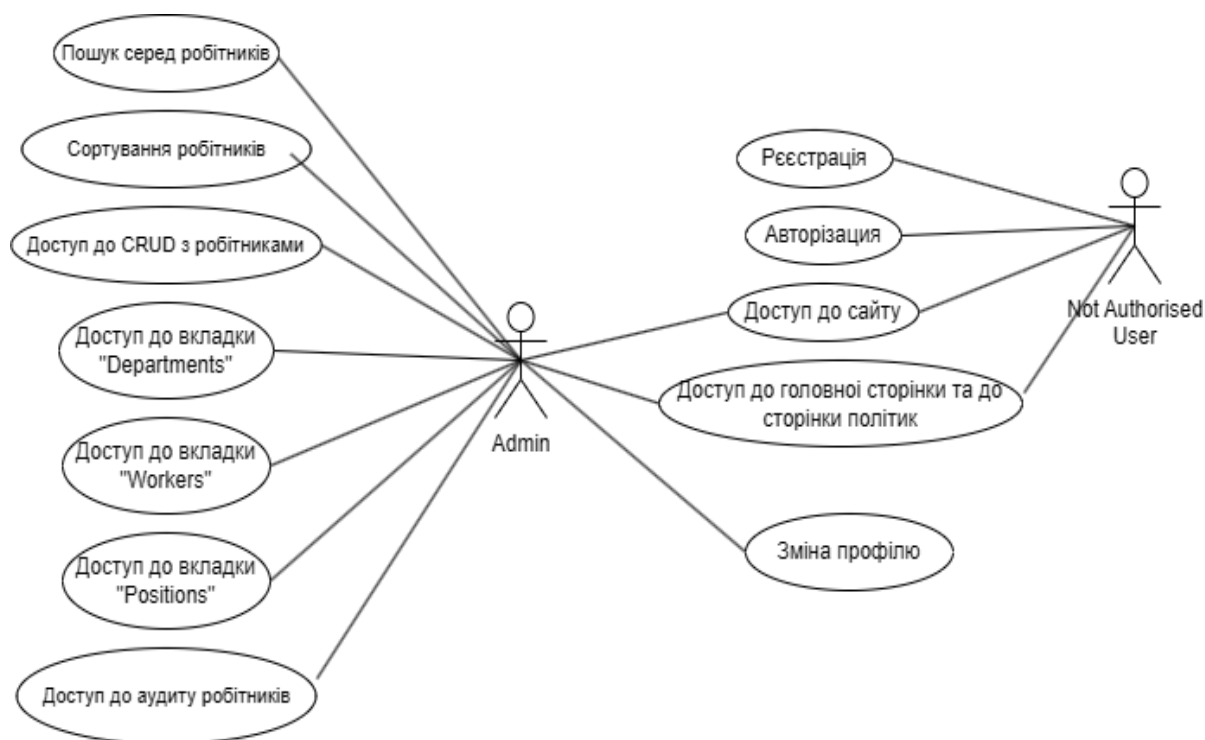


Рис. 3.3 Діаграма варіантів використання

Діаграма варіантів використання (Use Case Diagram) - це візуальний інструмент моделювання, що описує взаємодію між акторами (користувачами або зовнішніми системами) та системою. Вона наочно демонструє функціональні можливості системи з точки зору її користувачів, описуючи доступні їм варіанти використання. Діаграму варіантів використання для бакалаврської роботи можна побачити на Рисунку 3.3.

Бекенд частина додатку реалізована на платформі ASP.NET, яка надає потужні інструменти для створення надійних та масштабованих веб-додатків. Основні компоненти ASP.NET, які використовуються в нашому проєкті:

- ASP.NET Web API дозволяє створювати HTTP-сервіси, які можуть бути доступні з різних клієнтських додатків, таких як браузер, мобільні додатки або

інші сервери. Це забезпечує гнучкість у роботі з даними та дозволяє реалізувати архітектуру RESTful API[14]. Використання Web API дозволяє відокремити логіку обробки даних від їх відображення, що підвищує зручність підтримки та розширюваність системи.

- ASP.NET Core побудований за принципом модульності, що дозволяє додавати або видаляти компоненти в залежності від потреб проекту. Це підвищує продуктивність та зменшує розмір додатку. Модульна архітектура також полегшує інтеграцію нових функцій та сервісів, також надає вбудовані засоби для забезпечення безпеки додатків, включаючи аутентифікацію, авторизацію та захист від загроз. Використовуючи ці інструменти, можна легко впроваджувати захист даних та контролювати доступ до ресурсів додатку. Це особливо важливо для додатків, що обробляють конфіденційні дані або мають високі вимоги до безпеки. Також ASP.NET Core підтримує різні методи тестування, включаючи юніт-тести та інтеграційні тести. Це дозволяє забезпечити високу якість коду та стабільність додатку. Можливість автоматизованого тестування допомагає швидко виявляти та виправляти помилки, що знижує ризик появи критичних багів на продакшені.

Взаємодія між фронтенд та бекенд компонентами здійснюється за допомогою HTTP-запитів до Web API, реалізованого на ASP.NET. Razor Pages обробляють користувацькі запити, взаємодіють з API для отримання та відправлення даних, та відображають результати на веб-сторінках. Такий підхід забезпечує розподілення обов'язків між фронтенд та бекенд частинами, що підвищує гнучкість та масштабованість додатку.

- Фронтенд надсилає HTTP-запити до бекенду, який обробляє ці запити та повертає відповідь у форматі JSON або XML. Це дозволяє розділити логіку відображення та обробки даних, забезпечуючи чистоту та організованість коду. Використання асинхронних запитів зменшує навантаження на сервер та підвищує швидкість відповіді додатку.

- Використання асинхронних методів дозволяє підвищити продуктивність додатку, зменшуючи затримки при обробці запитів та покращуючи користувацький досвід. Це дозволяє обробляти більше запитів одночасно та забезпечує стабільну

роботу додатку під високими навантаженнями.

Сумуючі все що було досліджено - використання Razor Pages для фронтенду та ASP.NET для бекенду забезпечує ефективну, масштабовану та легко підтримувану архітектуру веб-додатку. Такий підхід до розробки дозволяє швидко реагувати на зміни вимог, забезпечує високу продуктивність та надійність роботи веб-додатку, що є критично важливим для успішної розробки продукту та задоволення потреб користувачів. Відокремлення логіки обробки даних від їх відображення дозволяє забезпечити чистий та організований код, який легко підтримувати та розширювати у майбутньому.

3.2 Архітектура БД додатку

Архітектура бази даних (БД) є невід'ємною частиною будь-якого програмного продукту з базою даних, оскільки від неї залежить ефективність роботи з даними, їх збереження та доступ. У роботі бакалавра було прийнято рішення використовувати підхід Code First для створення БД за допомогою Entity Framework. Цей підхід дозволяє спочатку створювати моделі класів у коді, а потім автоматично генерувати схему бази даних на основі цих моделей.

Підхід Code First надає наступні переваги:

Спочатку створюються класи моделей, які відображають структуру даних, що зберігаються в БД. Кожен клас відповідає таблиці в базі даних, а його властивості – стовпцям цієї таблиці.

ЕФ автоматично генерує міграції для відстеження змін у моделях класів і відповідного оновлення схеми БД. Це дозволяє легко вносити зміни в структуру бази даних без втрати даних. Завдяки використанню ORM розробники можуть працювати з БД на рівні об'єктів, що значно спрощує код і підвищує продуктивність розробки.

Для нашого додатку були створені наступні класи моделей Рисунок 3.4:

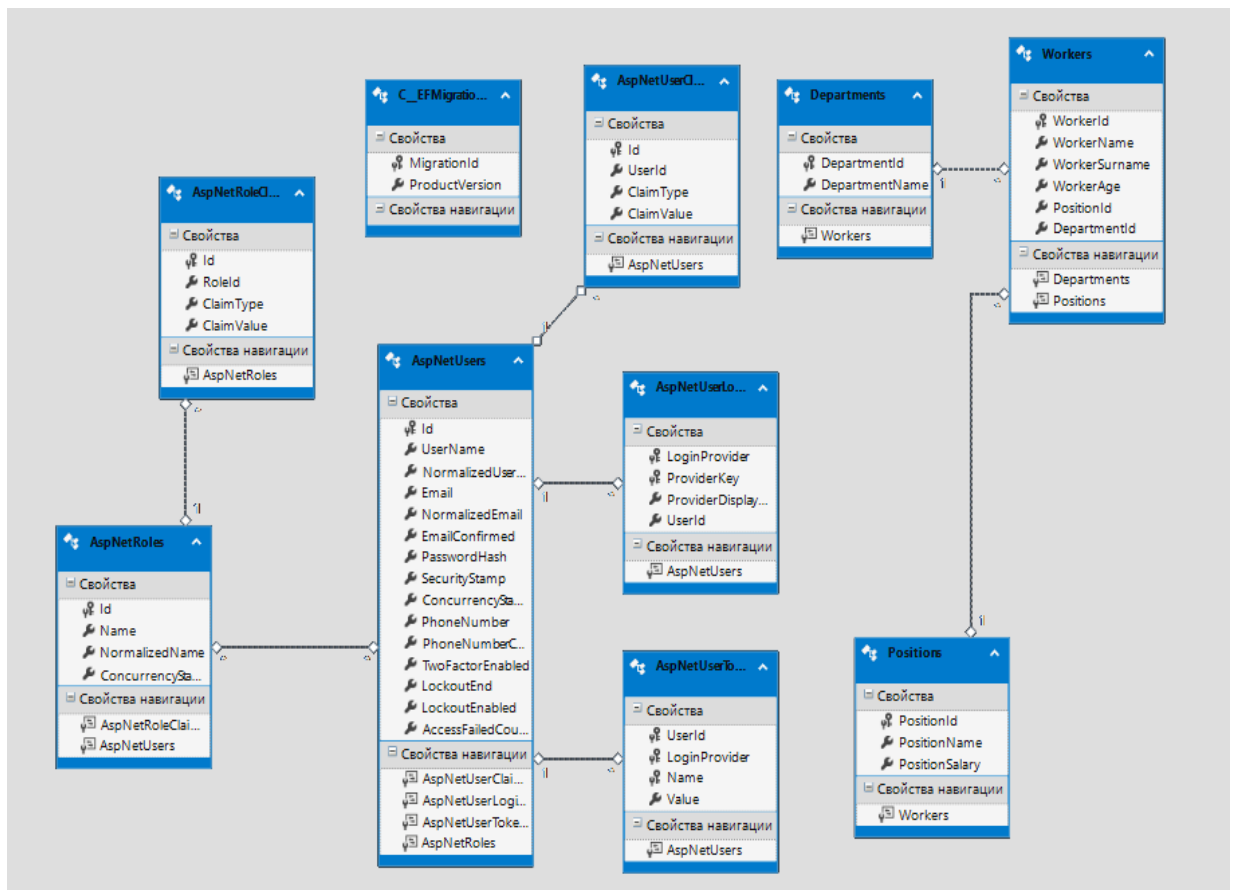


Рис. 3.4 Діаграма класів додатку

Діаграма класів бази даних демонструє структуру даних, їх взаємозв'язки та відображення на таблиці бази даних. На діаграмі представлені наступні класи моделей та їх властивості:

Клас **AspNetRoles**:

Id: Унікальний ідентифікатор ролі.

Name: Назва ролі.

ConcurrencyStamp: Відмітка часу для контролю конкурентного доступу.

Клас **AspNetRoleClaims**:

Id: Унікальний ідентифікатор клейму ролі.

RoleId: Зв'язок з таблицею **AspNetRoles**.

ClaimType: Тип клейму.

ClaimValue: Значення клейму.

Клас **AspNetUsers**:

Id: Унікальний ідентифікатор користувача.

UserName: Ім'я користувача.

NormalizedUserName: Нормалізоване ім'я користувача.

Email: Електронна пошта користувача.

EmailConfirmed: Підтвердження електронної пошти.

PasswordHash: Хеш пароля.

SecurityStamp: Відмітка безпеки.

ConcurrencyStamp: Відмітка часу для контролю конкурентного доступу.

PhoneNumber: Номер телефону.

PhoneNumberConfirmed: Підтвердження номера телефону.

TwoFactorEnabled: Увімкнення двофакторної аутентифікації.

LockoutEnd: Кінець блокування.

LockoutEnabled: Увімкнення блокування.

AccessFailedCount: Лічильник невдалих спроб входу.

Клас AspNetUserClaims:

Id: Унікальний ідентифікатор клейму користувача.

UserId: Зв'язок з таблицею AspNetUsers.

ClaimType: Тип клейму.

ClaimValue: Значення клейму.

Клас AspNetUserLogins:

LoginProvider: Провайдер логіну.

ProviderKey: Ключ провайдера.

ProviderDisplayName: Відображуване ім'я провайдера.

UserId: Зв'язок з таблицею AspNetUsers.

Клас AspNetUserTokens:

UserId: Зв'язок з таблицею AspNetUsers.

LoginProvider: Провайдер логіну.

Name: Ім'я токена.

Value: Значення токена.

Клас C_EFMigrationsHistory:

MigrationId: Ідентифікатор міграції.

ProductVersion: Версія продукту.

Departments

DepartmentId: Унікальний ідентифікатор департаменту.

DepartmentName: Назва департаменту.

Клас Positions:

PositionId: Унікальний ідентифікатор посади.

PositionName: Назва посади.

PositionSalary: Заробітна плата посади.

Клас Workers:

WorkerId: Унікальний ідентифікатор працівника.

WorkerName: Ім'я працівника.

WorkerSurname: Прізвище працівника.

WorkerAge: Вік працівника.

PositionId: Зв'язок з таблицею Positions.

DepartmentId: Зв'язок з таблицею Departments.

Взаємозв'язки між класами

AspNetRoles і AspNetRoleClaims: Один до багатьох (одна роль може мати багато клейм).

AspNetUsers і AspNetUserClaims: Один до багатьох (один користувач може мати багато клейм).

AspNetUsers і AspNetUserLogins: Один до багатьох (один користувач може мати багато логінів).

AspNetUsers і AspNetUserTokens: Один до багатьох (один користувач може мати багато токенів).

Departments і Workers: Один до багатьох (один департамент може мати багато працівників).

Positions і Workers: Один до багатьох (одна посада може мати багато працівників).

Опис взаємозв'язків

AspNetRoles - AspNetRoleClaims: Ролі можуть мати пов'язані клейми, які

визначають специфічні права та доступи для кожної ролі.

AspNetUsers - AspNetUserClaims: Користувачі можуть мати клейми, що додають додаткові атрибути або права користувача.

AspNetUsers - AspNetUserLogins: Користувачі можуть мати різні способи логіну, включаючи логіни через різні провайдери.

AspNetUsers - AspNetUserTokens: Токени використовуються для автентифікації користувачів і можуть бути збережені для різних цілей, таких як скидання пароля або двофакторна автентифікація.

Departments - Workers: Кожен департамент має багато працівників, що дозволяє організувати працівників за відділами.

Positions - Workers: Кожна посада може бути зайнята багатьма працівниками, що дозволяє визначити структуру організації та роль кожного працівника.

Ця діаграма класів надає чітке уявлення про структуру даних у базі даних додатку, взаємозв'язки між різними об'єктами та основні атрибути, що використовуються для зберігання інформації. Вона допомагає розробникам та іншим учасникам проекту зрозуміти, як організовані дані та як вони взаємодіють один з одним.

Першим кроком було створення класів моделей у проекті. Кожен клас відображав таблицю в базі даних.

Було налаштовано контекст бази даних (DbContext), який визначає, як моделі відображаються на таблиці БД. Це включає в себе налаштування зв'язків, обмежень та інших характеристик. Використовуючи команди Entity Framework, були створені міграції для ініціалізації бази даних та подальших змін у її схемі. Міграції були застосовані до бази даних, що призвело до створення таблиць та налаштування відповідних зв'язків між ними.

Завдяки використанню Code First з Entity Framework [15] наша база даних легко піддається розширенню та модифікаціям. Будь-які зміни в структурах даних можна легко внести через зміну моделей класів і застосування нових міграцій. Це забезпечує гнучкість у розробці та підтримці додатку, дозволяючи швидко

адаптуватися до нових вимог бізнесу.

3.3 Тестування

Тестування програмного забезпечення — це процес визначення відповідності функцій та характеристик застосунку вимогам, що до нього пред'являються. Метою цього процесу є надання максимально об'єктивної оцінки програмного продукту, включаючи всі його переваги та недоліки.

Тестування можна розділити на два основні етапи: верифікація та валідація.

Верифікація включає перевірку окремих компонентів програми або системи, щоб упевнитися в їхній відповідності поставленим цілям і задачам виробничого процесу. Це дозволяє виявити, чи правильно реалізовані технічні рішення та чи відповідають вони технічним вимогам.

Валідація полягає у визначенні, наскільки програма відповідає потребам користувачів та сформованим вимогам до системи. Це забезпечує впевненість у тому, що програмне забезпечення виконує всі необхідні функції та відповідає очікуванням кінцевих користувачів.

Для забезпечення систематичності та повноти тестування було складено таблицю, що базується на тест-кейсах. Таблиця 3.1 містить перелік користувацьких очікувань, передбачені способи вирішення цих проблем згідно з вимогами, а також результати проведених тестів.

Усі тести були виконані успішно, що підтверджує готовність системи до використання в реальних умовах. Це означає, що програмне забезпечення може стабільно працювати з реальними користувачами, забезпечуючи необхідний рівень якості та надійності.

Таблиця 3.1

Результати тестування веб-додатку

№ тесту	Тест-кейс	Очікуваний результат	Отриманий результат
1	Зайти на розділ сайту “Workers”, не маючи авторизації	Відкриватиметься вікно з помилкою 403 “Access Denied”	Passed
2	Авторизуватися не валідним аккаунтом	Вилізе нотифай що аккаунту не існує	Passed
3	Пройти авторизацію та отримати доступ до вкладки “Positions”	Відкриватиметься вікно з вкладкою “Positions”	Passed
4	Пройти авторизацію та Видалити усі співробітників з вкладки “Workers”	Має бути пуста сторінка з можливістю додавання нових працівників	Passed
5	Пройти авторизацію та спробувати змінити назву департаменту у вкладці “Departments”	Нова назва відобразиться на сторінці, а також вона зміниться на зв’язаних сторінках	Passed
6	Вийти натиснувши на кнопку Logout та спробувати внести зміни до вкладки “Workers”	Відкриватиметься вікно з помилкою 403 “Access Denied”	Passed
7	Авторизуватися та спробувати змінити пароль від аккаунту	Пароль буде успішно змінено	Passed

ВИСНОВКИ

За результатами аналізу процесів кадрового аудиту, кадрового контролю було визначено, що Контроль кадрів та кадровий аудит - це важливі інструменти для оптимізації роботи з персоналом. Існує безліч способів організувати кадровий облік, від паперового до спеціалізованих програмних комплексів.

Паперові методи прості та доступні, але незручні для відстеження великої кількості інформації, та можуть бути втрачені в наслідок війни, та природних катаклізмів. Електронні таблиці Excel та Google Sheets зручніші, але мають обмеження. LibreOffice Calc - безкоштовний та потужний інструмент, але може бути складним для нових користувачів. Спеціалізовані програмні комплекси пропонують найширший спектр функцій, але й коштують дуже дорого.

Ефективна, проста система контролю кадрів та кадрового аудиту у виді веб-додатку допоможе компаніям підвищити ефективність роботи з персоналом, забути про проблеми “Папірців” , а також зберегти конкурентні переваги на ринку. Сформовано те що програмне забезпечення для контролю кадрів повинно бути пристосоване для використання в дистанційному режимі, що особливо актуально в сучасному бізнес-середовищі, де віддалена робота стає все більш поширеною. Таким чином, воно має забезпечувати високий рівень доступності та зручності користування, дозволяючи менеджерам і співробітникам виконувати свої завдання незалежно від їхнього місцезнаходження. Основні можливості, які повинно надавати програмне забезпечення для контролю кадрів,це:

- Можливість внесення в систему інформації про людину
 - Можливість редагування внесеної інформації про людину
 - Можливість видалення людини з бази.
 - Система реєстрації та авторизації
 - Заборона доступу до деяких розділів сайту без авторизації
 - Можливість перевірки відповідності до вимог Українського законодавства даних працівника: віку, наявності військово-облікового документа,

відповідність посади освіти (ECTS), відповідність ступеню освіти до наявних ECTS кредитів.

Вибір середовища розробки, мови програмування, бібліотек, бази даних та інших інструментів був обґрунтований та аргументований.

Середовище розробки Visual Studio було обрано завдяки його зручності, наявності потужних інструментів для налагодження та тестування коду, а також підтримці інтеграції з системами контролю версій.

Мова програмування C# та платформа .NET були обрані через їхню поширеність, простоту використання, високу продуктивність, надійність та масштабованість. Для розробки веб-додатків використовувався фреймворк ASP.NET.

Бібліотеки Entity Framework та ASP.NET Core Identity забезпечують зручну роботу з базами даних та управління користувачами відповідно.

База даних Microsoft SQL Server (MSSQL) пропонує високу продуктивність, надійність, безпеку та інтеграцію з платформою .NET.

CSS та фреймворк Bootstrap використовуються для створення зручного та адаптивного інтерфейсу користувача.

Вибір цих інструментів та технологій дозволив створити якісний, надійний та масштабований веб-додаток, який відповідає сучасним вимогам та очікуванням користувачів.

Додаток відповідає сучасним вимогам та очікуванням користувачів, має зручний інтерфейс, забезпечує надійне зберігання даних та пройшов всебічне тестування.

Використання цього програмного продукту може суттєво покращити роботу кадрової служби компанії. Dodatok спрощує багато завдань, економить час та ресурси, допомагає виявляти та усувати помилки, а також надає керівництву інформацію для прийняття обґрунтованих кадрових рішень.

Впевнений, що цей веб-додаток стане цінним інструментом для багатьох компаній, які прагнуть покращити свою кадрову роботу.

Робота пройшла апробацію на Всеукраїнській науково-технічна конференція «Застосування програмного забезпечення в Інформаційно-комунікаційних технологіях», 24 квітня 2024 року, Київ, Державний Університет Інформаційно-Комунікаційних Технологій. За результатами апробації опубліковано теза доповіді: РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВЕДЕННЯ КАДРОВОГО АУДИТУ МОВОЮ C# ПА ПЛАТФОРМИ .NET. [16]

ПЕРЕЛІК ПОСИЛАНЬ

1. КРИВОРУЧКО, О. М.; КОВАЛЬОВА, О. П. Теоретичні положення кадрового забезпечення корпоративного менеджменту. *Вісник економіки транспорту і промисловості*, 2023, 81-82: 281-289.
2. Що таке кадровий аудит? [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://profpressa.com/consulting/audit#>
3. Особова картка — основний документ з обліку особового складу в організації [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: https://buh.ligazakon.net/analitycs/199257_osobova-karta-pratsvnika-poryadok-zapovnennya-ta-stroki-zbergannya#
4. Чому компанії переходять на ASP.NET Core для розробки веб-додатків для вирішення своїх бізнес завдань? [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://tqm.com.ua/ua/likbez/ua-articles/chomu-asp-net>
5. What is Excel? [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.techtarget.com/searchenterprisedesktop/definition/Excel>
6. Google sheets [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: https://www.google.com/intl/ru_ua/sheets/about/
7. LibreOffice Calc [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.libreoffice.org/discover/calc/>
8. Price M. J. C# 8.0 and. NET Core 3.0—Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP. NET Core, and ML. NET using Visual Studio Code. – Packt Publishing Ltd, 2019.
9. Dunford R., Su Q., Tamang E. The pareto principle. – 2014.
10. Framework E. Entity framework //Architecture [edit]. – 2021. – Т. 21. – С. 22.
11. Mistry R., Misner S. Introducing Microsoft SQL Server 2014. – Microsoft Press, 2014.

- 12.ЩО ТАКЕ ВЕБ ДОДАТОК? РІЗНИЦЯ МІЖ САЙТОМ, ВЕБ-ДОДАТКОМ, SPA І PWA [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://webcase.com.ua/uk/blog/cho-takoe-web-prilozhenie-vse-vidy/>
- 13.Strauss D., Strauss D. Working with Razor Pages //Creating ASP. NET Core Web Applications: Proven Approaches to Application Design and Development. – 2021. – С. 149-182.
- 14.Biehl M. RESTful Api Design. – API-University Press, 2016. – Т. 3.
- 15.Barskiy S. Code-First Development with Entity Framework. – Packt Publishing, 2015.
16. Струк М.Ю., Жебка В.В. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВЕДЕННЯ КАДРОВОГО АУДИТУ МОВОЮ С# ПА ПЛАТФОРМІ .NET. Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях», 24 квітня 2024 р., Київ, Державний Університет Інформаційно-Телекомунікаційних Технологій. Збірник тез. К.: ДУІКТ, 2024. С. 70-71.

ДОДАТОК А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка сайту для віддаленого ведення кадрового аудиту з використанням C# та .Net

Виконав студент 4 курсу
групи ПД-44
Струк Микита Юрійович
Керівник роботи

д.т.н., проф., зав. кафедри ТЦР Жебка Вікторія Вікторівна
Київ – 2024

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи:** спрощення процесу ведення кадрового аудиту, за рахунок використання програмного забезпечення розроблено з використанням мови C# та .Net
- **Об'єкт дослідження:** процес проведення віддаленого кадрового аудиту.
- **Предмет дослідження:** програмне забезпечення для віддаленого ведення кадрового аудиту.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз галузі кадрового аудиту та особливостей ведення віддаленого кадрового аудиту.
2. Провести аналіз готових рішень для галузі віддаленого кадрового аудиту.
3. Розробка та формування вимог до програмного забезпечення для ведення віддаленого кадрового аудиту.
4. Провести аналіз підходів до розробки програмного забезпечення з використанням ASP.NET, C#, обрати бібліотеки, базу даних та інші компоненти для реалізації програмного забезпечення.
5. Спроекувати веб-додаток для системи ведення віддаленого кадрового аудиту.
6. Реалізувати веб-додаток для системи ведення віддаленого кадрового аудиту.
7. Провести тестування програмного забезпечення для ведення віддаленого кадрового аудиту.

3

АНАЛІЗ АНАЛОГІВ

Показник	Microsoft Excel	Google Sheets	Libre Office Calc	J solutions	Кадри + Україна	1С:Оцінка персоналу	WEB HR DEPARTMENT
Платформи	MacOs, Windows	Android, iOS, Web	<u>Android, iOS, Web</u>	Web	Windows	Windows	<u>Web</u>
Можливість спільного редагування в режимі реального часу	-	+	-	-	-	-	+
Зберігання на віддаленому сервері	+	+	+	+	-	+	+
Offline доступ	+	-	+	-	+	+	+
Підтримка мобільних пристроїв	-	+	+	-	-	-	+
Інтеграція з іншими продуктами	+	+	+	+	-	+	+

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функціональні вимоги:

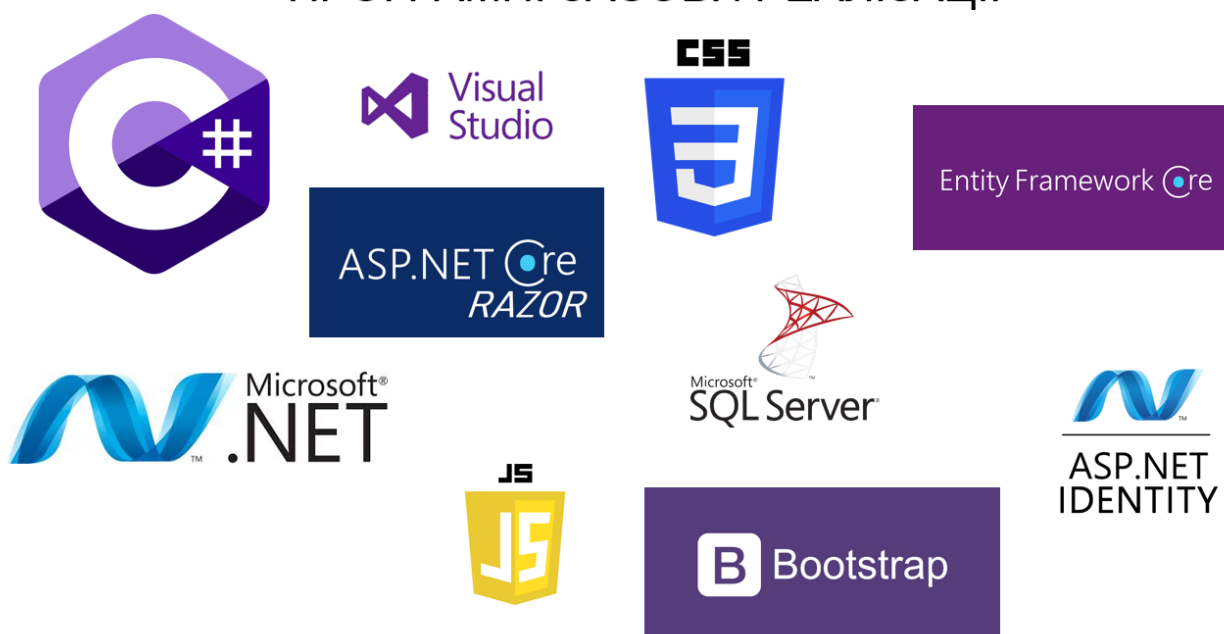
1. Можливість внесення в систему інформації про людину для ведення віддаленого кадрового аудиту.
2. Можливість редагування внесеної інформації про людину для ведення віддаленого кадрового аудиту.
3. Можливість видалення інформації про людину з бази даних.
4. Можливість перегляду інформації про людину.
5. Можливість перевірки відповідності до вимог Українського законодавства даних працівника: віку, наявність військово-облікового документа, відповідність посади освіти (ECTS), відповідність ступеню освіти до наявних ECTS кредитів.
6. Система реєстрації та авторизації.

Нефункціональні вимоги:

7. Англomовна локалізація.
8. Заборона доступу до розділів сайту без авторизації: Departments, Positions, Workers.
9. Розгалуження доступу для користувачів: Admin, Non-Authorised User.
10. Кросбраузерність, робота в браузерях на базі Chromium.

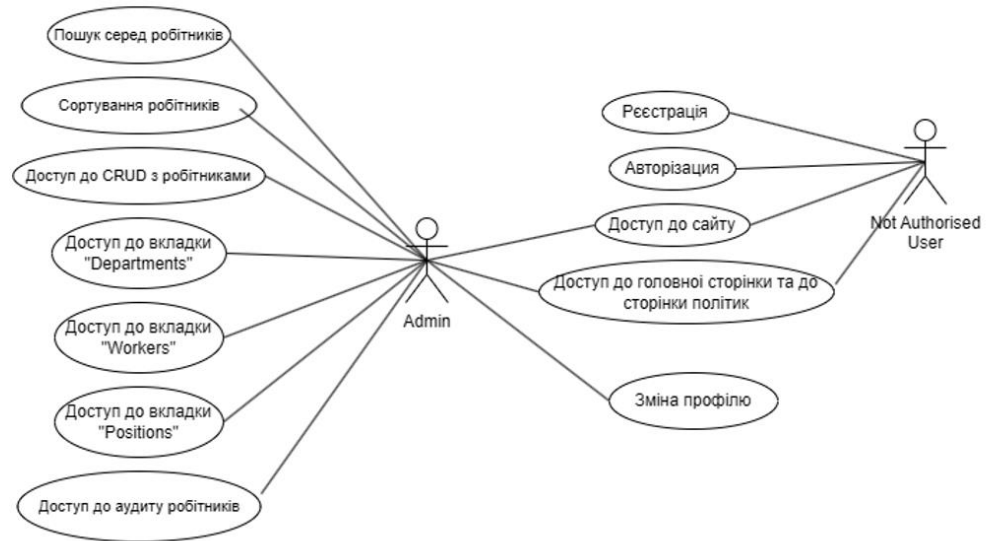
5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



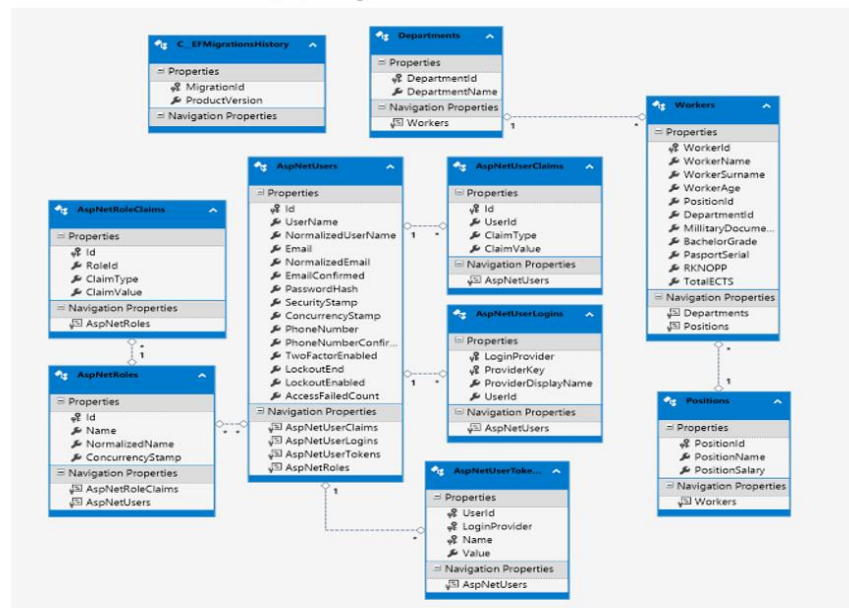
6

Діаграма варіантів використання



7

Діаграма класів



8

Екранні форми

HR Department

HomePrivacyDepartmentsPositionsWorkers

Hello admin123@gmail.com!Logout

Workers

Create New

Click to Audit Workers

Find by Name/Surname:

Search

Back to full List

WorkerName	WorkerSurname	WorkerAge	MilitaryDocument	RKNOPP	PasportSerial	BachelorGrade	TotalECTS	Position	Department	
Biba	Andropov	23	☐	558019254	596583190	☒	120	Software Engineer	IT-Department	EditDeleteDetails
Boba	Sechenov	26	☐	291736997	369931674	☐	0	LEAD QA	IT-Department	EditDeleteDetails
Boris	Animal	32	☐	404241828	802600469	☒	240	Analyst	IT-Department	EditDeleteDetails
Andrey	Sulikorve	21	☐	486272095	737064835	☒	240	Analyst	Teaching Department	EditDelete

сторінка для дій над інформацією про робітників

11

Екранні форми

Krevenko

WorkerAge

18

MilitaryDocument

☒

RKNOPP

23123123

PasportSerial

32463434

BachelorGrade

☒

TotalECTS

240

PositionId

DiplomPosition

DepartmentId

DUIKT-DIPLOMA-DEP

Save

Back to List

сторінка зміни інформації про робітника

12

Екранні форми

Сторінка авторизації

Спроба доступу без авторизації

13

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Струк М.Ю, Жебка В.В. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВЕДЕННЯ КАДРОВОГО АУДИТУ МОВОЮ C# ПА ПЛАТФОРМІ .NET. // Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». Збірник тез. 24 квітня 2024 р., ДУІКТ, м. Київ – К: ДУІКТ, 2024, 70 с.

15

ВИСНОВКИ

1. Проведено аналіз галузі кадрового аудиту, визначено основні вимоги та нормативи, що регулюють кадровий аудит. Виявлено, що віддалений кадровий аудит потребує високого рівня цифрової компетентності, відповідного програмного забезпечення.
2. Проведено аналіз існуючих програмних рішень для віддаленого кадрового аудиту, включаючи популярні платформи та сервіси. Виявлено, що більшість рішень не надають можливість спільного редагування в реальному часі або вимагають значних налаштувань для цього.
3. Розроблено та сформовано функціональні та нефункціональні вимоги до програмного забезпечення, програмне забезпечення повинно працювати в дистанційному режимі(web), та повинно мати графічний інтерфейс. Основні функції включають внесення, редагування та видалення інформації про співробітників, огляд та аудит інформації про працівників, систему реєстрації та авторизації, а також обмеження доступу без авторизації.
4. Проведено аналіз підходів до розробки програмного забезпечення з використанням ASP.NET, C#. Обрано бібліотеки та компоненти, зокрема, Entity Framework для роботи з базою даних, ASP.NET Core Identity для керування користувачами, а також Microsoft SQL Server як основну базу даних. Використано CSS та Bootstrap для дизайну інтерфейсу користувача.
5. Спроектовано веб-додаток для системи ведення віддаленого кадрового аудиту, враховуючи функціональні та нефункціональні вимоги. Визначено архітектуру додатку як веб-додаток, включаючи структуру бази даних яка спроектована CodeFirst методом.
6. Реалізовано веб-додаток для системи ведення віддаленого кадрового аудиту, використовуючи Visual Studio, C#, .NET, ASP.NET, Entity Framework, ASP.NET Core Identity, Microsoft SQL Server, CSS та Bootstrap. Створено веб-додаток, який відповідає раніше сформованим вимогам.
7. Проведено тестування веб-додатку для ведення віддаленого кадрового аудиту. Тестування не виявило помилок в роботі програмного забезпечення.