

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

КУРСОВА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Бази даних»
на тему:
«База даних моніторингу робочого часу працівника»

студента II курсу групи ПЗ-21-4
спеціальності 121 «Інженерія програмного
забезпечення»

Грибана Владислава Валентиновича
(прізвище, ім'я та по-батькові)

Керівник: доцент кафедри КН
Сугоняк І.І

Дата захисту: " ____ " _____ 20__ р.
Національна шкала _____
Кількість балів: _____
Оцінка: ECTS _____

Члени комісії

_____	<u>Ольга Коротун</u>
(підпис)	(прізвище та ініціали)
_____	<u>Світлана Кравченко</u>
(підпис)	(прізвище та ініціали)
_____	<u>Чижмотря Олексій</u>
(підпис)	(прізвище та ініціали)

Житомир – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп'ютерних технологій

Кафедра Інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

В.о зав кафедри ІПЗ

_____ А.В Морозов

“ ____ ” _____ 2023 р.

ЗАВДАННЯ
НА КУРСОВИЙ ПРОЕКТ СТУДЕНТУ

Грибану Владиславу Валентиновичу

- Тема роботи: База даних моніторингу робочого часу працівника.
керівник роботи: доцент кафедри КН Сугоняк І.І.
- Строк подання студентом: “ 29 ” травня 2022 р.
- Вихідні дані до роботи: База даних моніторингу робочого часу працівника.
- Зміст розрахунково-пояснювальної записки (перелік питань, що підлягають розробці)
 - Аналіз інформаційних потоків та особливостей предметної області дослідження
 - Проектування бази даних за напрямком курсової роботи
 - Реалізація підсистеми обробки даних за напрямком курсової роботи
 - Адміністрування баз даних
- Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
 - Презентація до Курсової роботи
 - Посилання на репозиторій: <https://github.com/Fr0stYANN/time-tracker>
- Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3	Сугоняк І.І.	10.02.2023	10.02.2023

- Дата видачі завдання “ 10 ” лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсової роботи	Термін виконання етапів роботи	Примітки
1	Постановка задачі	10.02.2022	Виконано
2	Пошук, огляд та аналіз аналогічних розробок	23.02.2022	Виконано
3	Формулювання технічного завдання	05.03.2022	Виконано
4	Опрацювання літературних джерел	10.03.2022	Виконано
5	Проектування структури	05.04.2022	Виконано
6	Написання програмного коду	30.04.2022	Виконано
7	Відлагодження	14.05.2022	Виконано
8	Написання пояснювальної записки	20.05.2022	Виконано
9	Захист		

Студент

(підпис)

Грибан В.В

(прізвище та ініціали)

Керівник проекту

(підпис)

Сугоняк І.І

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до курсового проекту на тему «База даних моніторингу робочого часу працівника» складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 25 сторінках друкованого тексту.

Пояснювальна записка має 5 сторінок додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 22 малюнка. Загальний обсяг роботи – 35 сторінок.

У додатку представлений лістинг розробленого програмного продукту.

Ключові слова: C#, ASP.NET Core, MS SQL, Dapper, CEPBER, КЛІЄНТ, FRONTEND, BACKEND, SQL, LOCALSTORAGE.

					ДУ «Житомирська політехніка».23.121.08.000						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Грибан В.В			«Розробка бази даних моніторингу робочого часу працівника»			Літ.	Арк.	Аркушів	
Перевір.		Сугоняк І.І								4	40
Керівник								ФІКТ Гр. ІПЗ-21-4[1]			
Н. контр.											
Зав. каф.											

Зміст

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ.....	5
1.1 Аналіз задачі, засобів та методів її вирішення.....	5
1.2 Архітектура та засоби реалізації бази даних.....	5
РОЗДІЛ 2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ МОНІТОРИНГУ РОБОЧОГО ЧАСУ ПРАЦІВНИКА	9
2.1 Аналіз структури інформаційних процесів.....	9
2.2 Проектування бази даних оренди житла.....	10
2.3 Розробка математичної моделі та алгоритмів обробки інформації бази даних моніторингу робочого часу працівників.....	11
Висновки до другого розділу.....	12
РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ.....	13
3.1 Опис роботи додатку.....	13
3.2 Реалізація операцій обробки даних в БД.....	18
Висновки до третього розділу.....	19
РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ	20
4.1. Розробка заходів захисту інформації в БД	20
Висновки до четвертого розділу.....	21
ВИСНОВКИ	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	23
ДОДАТКИ.....	24
Додаток А.....	25

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – База даних

КР – Курсова робота

ПЗ – Програмне забезпечення

СУБД – система управління базою даних

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

У цій курсовій роботі буде наведено процес розробки бази даних моніторингу робочого часу працівників.

Актуальність теми. Бази даних становлять невід’ємну складову діяльності сучасних підприємств та організацій. Невпинне зростання обсягів інформації, що зберігається в базах даних, та розширення кола користувачів висувають до систем керування базами даних (СКБД) нові вимоги.

Потреба у швидкому редагуванні та отриманні інформації з бази даних постала ще в 1970-роках. Великі обсяги інформації робили неможливим ручне її опрацювання. Саме тоді в дослідницькій лабораторії компанії IBM розробляються мови запитів SQL та QBE, які сьогодні реалізовані практично в кожній системі керування базами даних. Візуальний підхід до формування запитів мовою QBE робить її незамінним інструментом для задоволення інформаційних потреб користувачів, які не володіють навикам програмування. Декларативна мова запитів SQL не вимагає вказання алгоритму отримання даних, а тому з легкістю може бути інтегрована в будь-який програмний код, що дає програмістам широкі можливості по створенню, модифікації баз даних та вибірці інформації з них в режимі реального часу.

Метою курсової роботи є дослідження особливостей проектування та реалізації бази даних за визначеною темою курсової роботи.

Завданням на курсову роботу є:

- аналіз теоретичних засад проектування та реалізації систем на основі баз даних;
- визначення інформаційних потреб предметної області дослідження;
- аналіз напрямку ризиків інформаційних потоків та їх структури;
- проектування бази даних за визначеною предметною областю;
- розробка математичної та алгоритмічної моделі функціонування системи на основі БД;
- реалізація БД та інтерфейсних засобів інформаційної системи.

Об’єктом дослідження є методи та засоби проектування баз даних за визначеними предметними областями.

Предмет дослідження є використання можливостей застосування концепції БД та СУБД для забезпечення інформаційних потреб предметної області.

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		6

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1 Аналіз задачі, засобів та методів її вирішення

В даному курсовому проекті за предметну область взято базу даних, яка зможе зберігати усю необхідну інформацію для моніторингу робочого часу працівників.. Це дає можливість працівникам компанії відслідковувати робочий час.

Для коректного виконання запитів на отримання даних, дані повинні вноситись та змінюватись у БД. За це будуть відповідати деякий перелік операцій:

- 1) Вхід та реєстрація користувача. Щоб кожного клієнта можна було б розрізнити і зберігати їхню конфіденційність
- 2) Створення робочої сесії, її редагування.
- 3) Створення запиту на відпустку, можливість її підтверджувати, або ні.
- 4) Гістограма робочого часу працівника.
- 5) Перегляд робочих сесій, їх редагування, додавання.

1.2 Архітектура та засоби реалізації бази даних.

Аналіз та вибір СУБД проведемо з урахуванням того, що число клієнтських місць становить від 30 до 500, а доступ до даних має бути максимально ефективним. Обчислювальна техніка працюватиме під керівництвом ОС Windows і Linux. В таблицях наведено порівняльні характеристики СУБД.

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		6

Таблиця 1.1.

Характеристика	Microsoft SQL Server	MongoDB	PostgreSQL
Робота під керівництвом ОС	Задовільно	Добре	Добре
Складність	Відмінно	Добре	Відмінно
Доступ до даних	Відмінно	Відмінно	Відмінно
Використання у великих проектах	Відмінно	Добре	Відмінно
Використання у малих проектах	Добре	Відмінно	Відмінно
Схема даних	Відмінно	Відмінно	Добре
Підтримувані мови програмування	Відмінно	Відмінно	Відмінно
Підключення до Web	Відмінно	Відмінно	Відмінно
Графічні інструменти	Відмінно	Відмінно	Відмінно
Тригери	Відмінно	Добре	Відмінно
Паралелізм	Відмінно	Відмінно	Відмінно
Одночасний доступ декількох користувачів	Добре	Добре	Відмінно
Обробка даних мультимедіа	Погано	Добре	Відмінно
Пошук за регулярними виразами	Добре	Відмінно	Відмінно
Засоби аналізу	Відмінно	Відмінно	Відмінно

MongoDB

MongoDB є провідною базою даних документів . Яка побудована на розподіленій масштабованій архітектурі та стала комплексною хмарною платформою для керування та доставки даних у програми. MongoDB обробляє транзакційні, операційні та аналітичні навантаження.

MongoDB зберігає дані як документи у двійковому представленні під назвою BSON (Binary JSON). Поля можуть відрізнятися від документа до документа; немає необхідності оголошувати системі структуру документів – документи описуються самі. Якщо до документа потрібно додати нове поле, це поле можна створити, не впливаючи на всі інші документи в колекції, не оновлюючи центральний системний каталог, оновлюючи ORM і не переводячи систему в автономний режим. За бажанням перевірку схеми можна використовувати для забезпечення контролю керування даними над кожною колекцією.

MongoDB гарантує повну ізоляцію під час оновлення документа. Будь-які помилки ініціюватимуть операцію оновлення до відкату, скасовуючи зміни та гарантуючи, що клієнти отримають узгоджений вигляд документа.

MongoDB також підтримує транзакції бази даних у багатьох документах, тому фрагменти пов'язаних змін можна зафіксувати або відкотити. Завдяки можливості транзакцій з багатьма документами MongoDB є однією з небагатьох баз даних, яка поєднує в собі гарантії ACID традиційних реляційних баз даних зі швидкістю, гнучкістю та потужністю моделі документів.

Транзакції в MongoDB складаються з кількох інструкцій із подібним синтаксисом, тому будь-кому, хто має попередній досвід роботи з транзакціями, легко додати такі конструкції до будь-якої програми.

PostgreSQL

PostgreSQL є прикладом добре керованого проекту з відкритим кодом. PostgreSQL називає себе системою об'єктно-реляційної бази даних з відкритим кодом. Це база даних SQL, яка має певні стратегії для обробки індексування, підвищення паралельності та впровадження оптимізацій і підвищення продуктивності, включаючи розширене індексування, розділення таблиць та інші механізми. Об'єктна частина PostgreSQL пов'язана з багатьма розширеннями, які дозволяють включати інші типи даних, такі як об'єкти даних JSON та XML.

MS SQL Server

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

MS SQL Server - система управління базами даних, яка розробляється корпорацією Microsoft. Як сервер даних виконує головну функцію по збереженню та наданню даних у відповідь на запити інших застосунків, які можуть виконуватися як на тому ж самому сервері, так і у мережі. Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

Отже, після порівняння вказаних вище СУБД, було вирішено використовувати MSSQL.

Висновки до першого розділу :

Було проаналізовано СУБД (MS SQL Server, MongoDB, PostgreSQL), їхні переваги та недоліки. Також під час опрацювання цього розділу, було обрано використовувати СУБД MS SQL на курсовий проект. Було поставлено задачі, які мають бути виконані.

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ МОНІТОРИНГУ РОБОЧОГО ЧАСУ ПРАЦІВНИКА

2.1 Аналіз структури інформаційних процесів

На етапі проектування потрібно скласти загальний алгоритм роботи програми.

На головній сторінці користувач матиме змогу увійти. Всього у користувачів буде три опції – можливість керувати календарем «спеціальних днів», можливість керувати іншими користувачами, та «імперсонеїтати їх» переглядати їх відпустки, лікарняні, робочі сесії, і тд. Керувати робочим часом працівників, видаляти робочі сесії, редагувати їх.. А також у користувача, який працює не повний робочий день, на головній сторінці буде можливість почати робочу сесію і слідкувати за своїм відпрацьованим робочим часом. І у кожного користувача буде можливість створити запит на відпустку, подивитись свої лікарняні.

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		6

2.2 Проектування бази даних оренди житла

Внаслідок проектування до бази даних включено таблиці:

- Calendar
- TimeTracks
- Users
- Vacations
- SickLeaves

Таблиця 2.1

Структура таблиці **Calendar**

Назва поля	Тип даних	Опис поля
Id	Int	ID
Date	date	Дата дня
HoursToWork	int	Кількість годин, які треба працювати в цей день
DayTypeId	int	Тип робочого дня

Таблиця 2.2

Структура таблиці **TimeTracks**

Назва поля	Тип даних	Опис поля
Id	Int	ID
UserId	Int	Id користувача
CreationTypeId	Int	Тип створення тайм треку
StartDate	Datetime	Дата початку
EndDate	Datetime	Дата кінця
TimeTrackTypeId	Int	Тип таймтреку
TotalTime	Nvarchar	Тотальний час

Таблиця 2.3

Структура таблиці **Users**

Назва поля	Тип даних	Опис поля
Id	Int	ID

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

Email	Nvarchar	Емейл
Firstname	Nvarchar	Ім'я
Lastname	Nvarchar	Прізвище
EmploymentDate	Date	Дата прийняття на роботу
WorkTypeId	Int	Тип працевлаштованості(парт, фулл тайм)
IsActivated	Bit	Активований
Password	Nvarchar	пароль

Для збереження відпусток призначена таблиця **Vacations**

Таблиця 2.4

Структура таблиці **Vacations**

Назва поля	Тип даних	Опис поля
Id	Int	ID
StartDate	Date	Початок
EndDate	Date	Кінець
Comment	Nvarchar	Коментар
IsApproved	Bit	Підтверджена
UserId	Int	id

Для збереження відпусток призначена таблиця **SickLeaves**

Таблиця 2.4

Структура таблиці **SickLeaves**

Назва поля	Тип даних	Опис поля
Id	Int	ID
StartDate	Date	Початок
EndDate	Date	Кінець
UserId	Int	id

2.3 Розробка математичної моделі та алгоритмів обробки інформації бази даних моніторингу робочого часу працівників

Для збереження інформації про користувача в базі даних, користувачу з опцією створення користувачів потрібно створити користувача де він повинен ввести Прізвище, Ім'я, пошту та пароль. Після того як він ввів ці дані, він їх відправляє, ці дані оброблюються на бекенді, де користувачу за замовчуванням задається роль користувача, а також перевіряється, чи існує дана пошта користувача в базі даних, пароль шифрується.

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		6

Після створення користувача, на головній сторінці користувач матиме змогу увійти. Всього у користувачів буде три опції – можливість керувати календарем «спеціальних днів», можливість керувати іншими користувачами, та «імперсонеїти їх» переглядати їх відпустки, лікарняні, робочі сесії, і тд. Керувати робочим часом працівників, видаляти робочі сесії, редагувати їх.. А також у користувача, який працює не повний робочий день, на головній сторінці буде можливість почати робочу сесію і слідкувати за своїм відпрацьованим робочим часом. І у кожного користувача буде можливість створити запит на відпустку, подивитись свої лікарняні.

Висновки до другого розділу

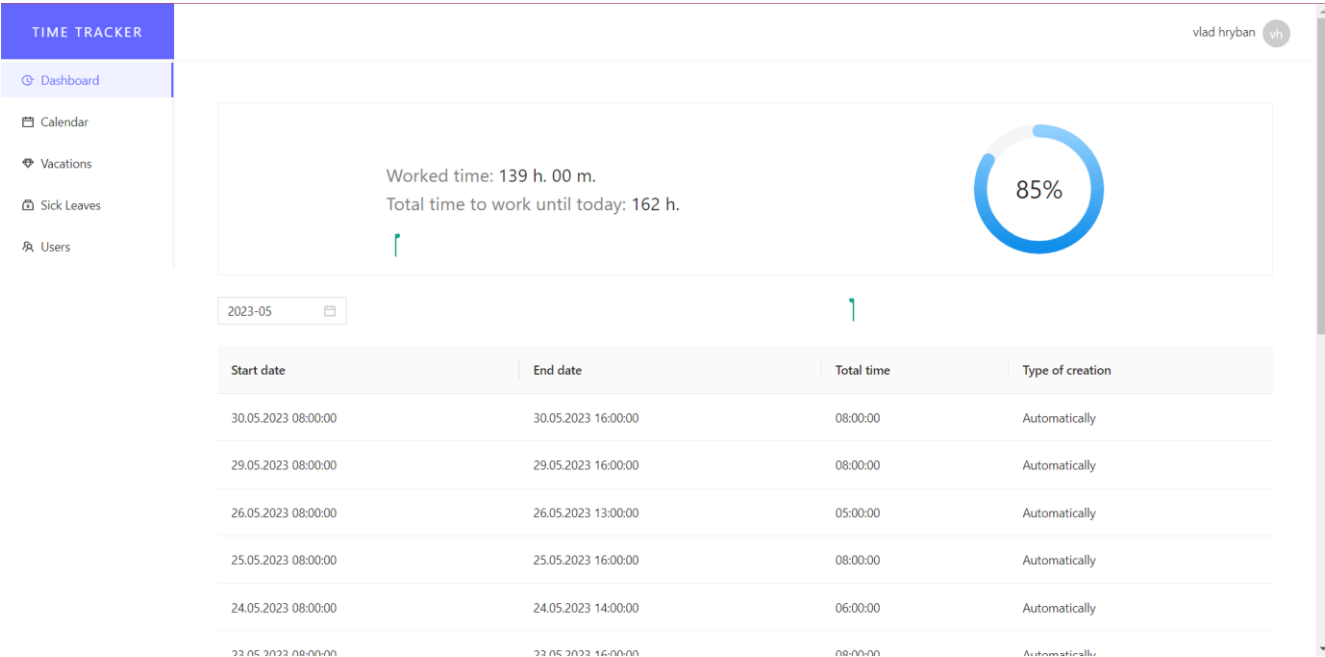
В другому розділі було проаналізовано інформаційні процеси, побудовано структуру бази даних, описано, яку буде мати логіку сайт, для збереження інформації до бази даних. Було проаналізовано, за допомогою чого будуть зберігатись дані і як отримувати інформацію з бази даних.

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

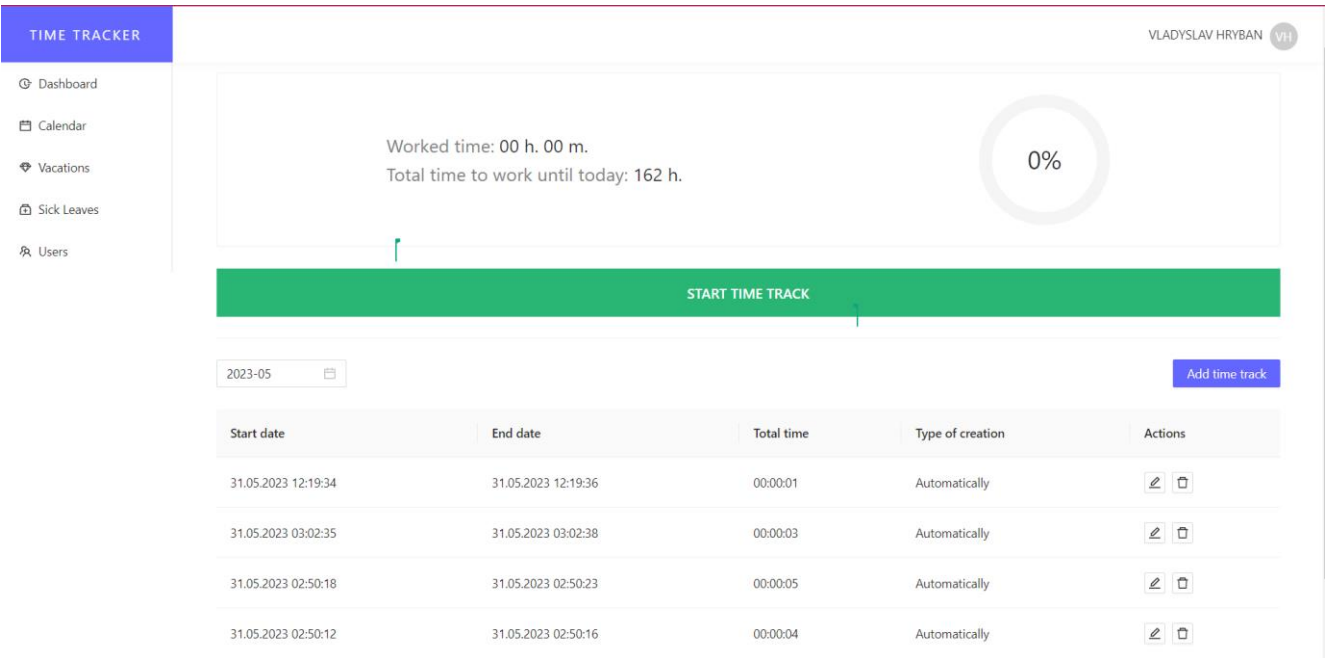
РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ

3.1 Опис роботи додатку

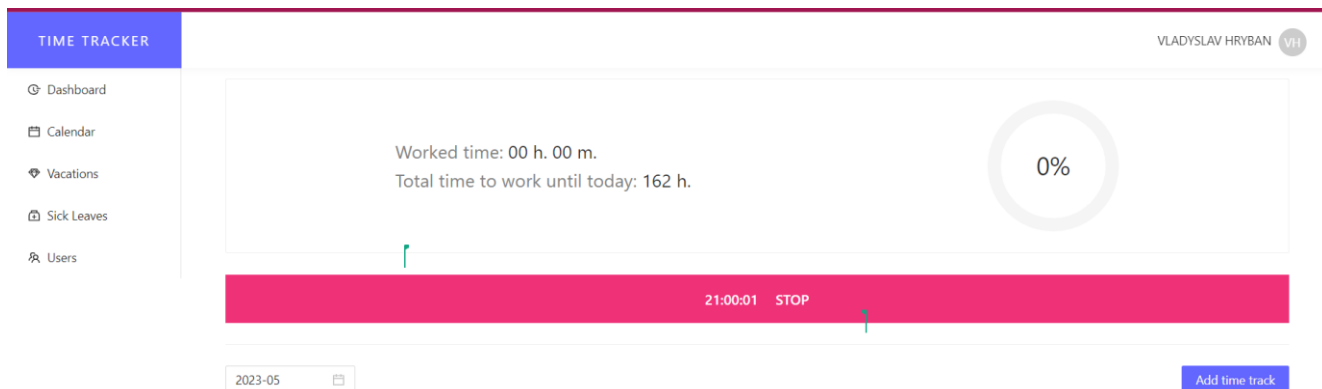
При запуску додатку користувач бачить головну сторінку сайту, де розміщено блок відпрацьованих в місяці годин .



Мал.3.1.Головна сторінка



Мал.3.2.Головна сторінка для парт-таймера



Мал.3.3.Головна сторінка для парт-таймера

Edit time track

Time interval

2023-05-31 15:22:14 → 2023-05-31 15:24:14

Cancel

Save

Мал.3.4. Модальне вікно редагування робочої сесії

Create time track

Time interval

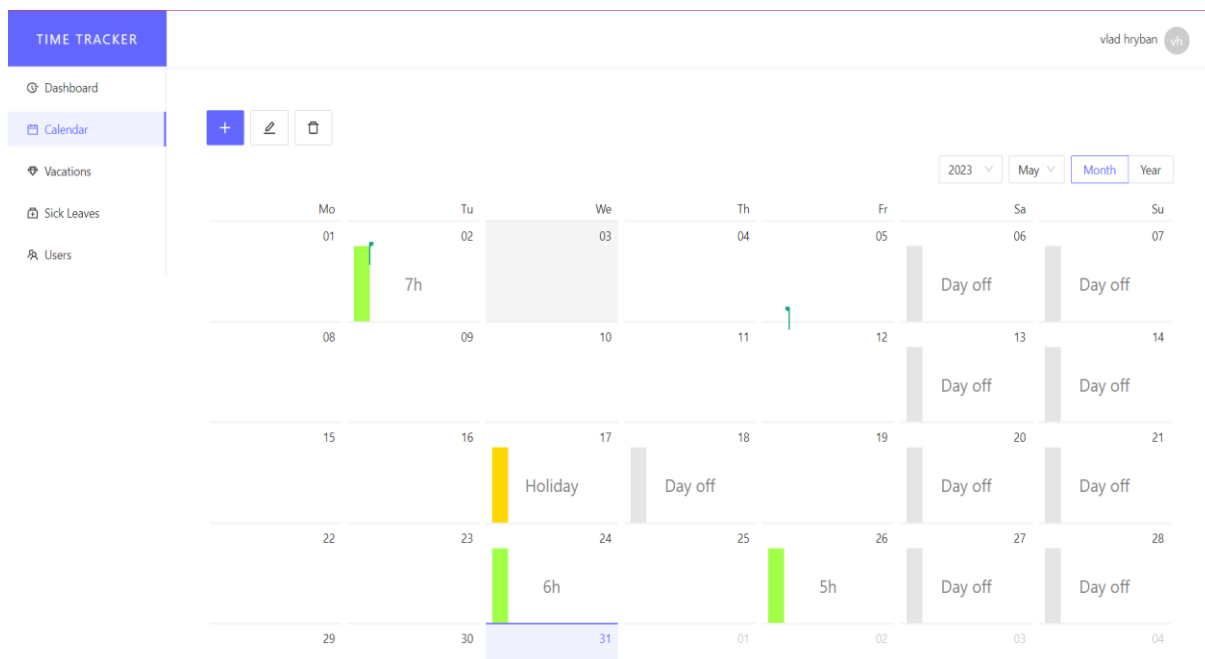
Start date → End date

Cancel

Create

Мал.3.5. Модальне вікно додавання робочої сесії

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		



Мал.3.6.Сторінка з календарем

Мал.3.7.Модальне вікно редагування скороченого дня

EDIT SPECIAL DAY

×

Date of editing day:

2023-05-18

📅

Type of adding day:

Day Off

▼

Works hours: 0

○

○

○

○

○

○

○

○

○

○

1

Cancel

Edit Day

Мал.3.8.Модальне вікно редагування вихідного дня

DELETE SPECIAL DAY

×

Date of deleting day:

Day ▼

2023-05-18

📅

Type of deleting day:

Day Off

▼

Works hours: 0

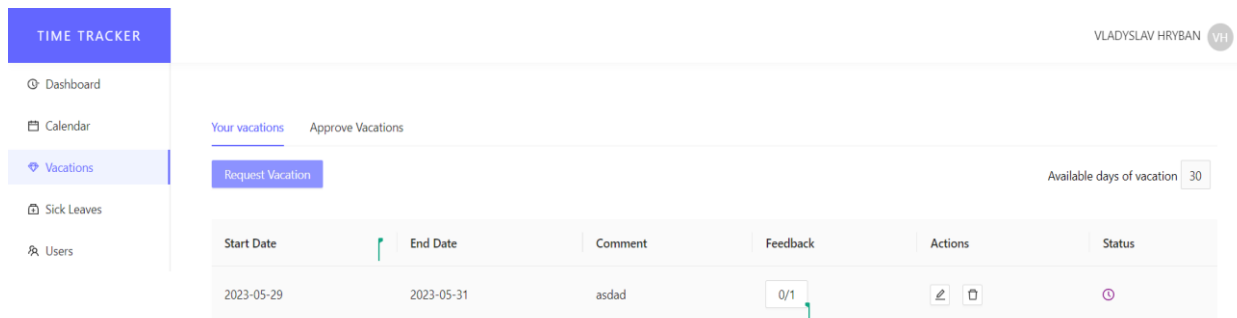
1

Are you want delete this day from calendar?

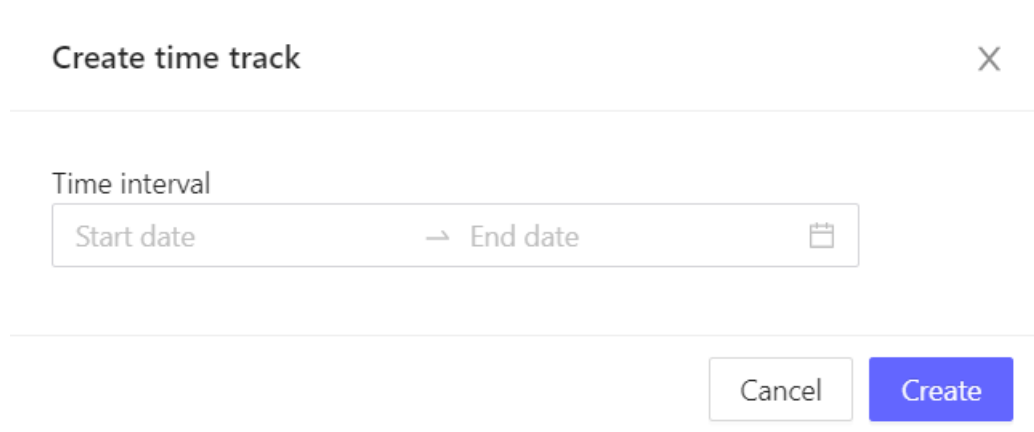
Cancel

Delete

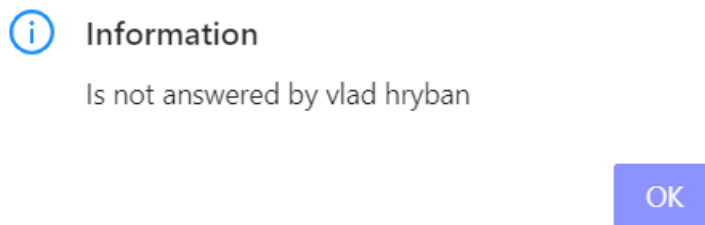
Мал.3.9.Модальне вікно видалення вихідного дня



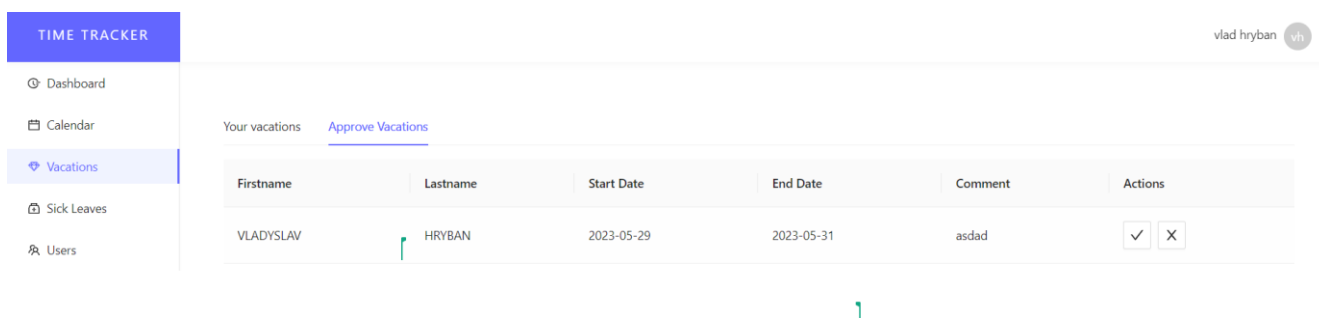
Мал.3.10. Сторінка з відпустками



Мал.3.11. Модальне вікно створення запиту на відпустку



Мал.3.12. Модальне вікно інформації про відгуки на відпустку



Мал.3.13. Сторінка підтвердження відпусток

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

Confirm approve



Are you sure you want to approve vacation request for VLADYSLAV HRYBAN from 2023-05-29 to 2023-05-31?

comment (if needed)

Cancel

OK

Мал.3.14. Модальне вікно підтвердження відпустки

TIME TRACKER	vlad hryban					
Dashboard	Download	Click to sort ascending	Add User			
Calendar	Firstname	Lastname	Email	Employment Date	Work type	Actions
Vacations	vlad	hryban	vl@gmail.com	2021-05-05	full-time	...
Sick Leaves	VLADYSLAV	HRYPAN	vlad.gryban73@gmail.com	2023-02-07	part-time	...
Users	VLADYSLAV	HRYPAN	vlad.gryban74@gmail.com	2023-05-29	part-time	...
	VLADYSLAV	HRYPAN	vlad.gryban75@gmail.com	2023-05-29	part-time	...

Мал.3.15. Сторінка з користувачами

3.2 Реалізація операцій обробки даних в БД

За допомогою Dapper реалізовані операції обробки даних в БД. Наші дані перетворюються з сутностей БД в об'єкти в мові програмування с# в репозиторіях.

Лістинг методу отримання всіх календарних днів з бази даних:

```
public List<CalendarModel> GetAll()
{
    string query = @"
        SELECT
            C.Id,
            C.Date,
            C.HoursToWork,
            C.DayTypeId,
            D.Id,
            D.Name
        FROM Calendar C
        LEFT JOIN DayTypes D ON D.Id = C.DayTypeId";

    using var connection = new SqlConnection(connectionString);
    return connection.Query<CalendarModel, DayTypeModel, CalendarModel>(query,
        (calendar, dayType) =>
        {
            calendar.DayType = dayType;
        }
    );
}
```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return calendar;
    }, splitOn: "Id").ToList();
}

```

Лістинг методу отримання календарного дня за датою:

```

public CalendarModel? GetByDate(DateTime date)
{
    string query = @"
        SELECT
            C.Id,
            C.Date,
            C.HoursToWork,
            C.DayTypeId,
            D.Id,
            D.Name
        FROM Calendar C
        LEFT JOIN DayTypes D ON D.Id = C.DayTypeId
        WHERE C.Date = @Date";

    using var connection = new SqlConnection(connectionString);
    return connection.Query<CalendarModel, DayTypeModel, CalendarModel>(query,
(calendar, dayType) =>
    {
        calendar.DayType = dayType;

        return calendar;
    }, splitOn: "Id", param: new {Date = date}).FirstOrDefault();
}

```

Висновки до третього розділу

Було детально розписано взаємодію користувача з БД. Було створено інтерфейс сайту моніторингу робочого часу працівників. Було розроблені моделі календаря, таймтреку, користувачів та відпусток. Були створені сторінки з можливість перегляду даних, виконання різних запитів на фільтрацію даних, виконання запитів на отримання чи підвантаження даних з БД.

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		6

РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ

4.1. Розробка заходів захисту інформації в БД

Отже, на сайті орендування житла існує тільки 2 головні ролі: адміністратор та менеджер, 3 роль – люди, що орендують. Найбільші права доступу до даних має адміністратор. Він може переглядати будь-які дані сайт. На перетині рядків і стовпців наведено дії, які користувач може виконувати з певною роллю: 1 – перегляд даних, 2 – додавання, 3 – редагування, 4 – видалення, 5 – повний доступ

Таблиця 4.1

	Users	Vacations	TimeTracks	SickLeaves
Адміністратор	5	5	5	5

Також існує користувач з роллю менеджер, він може робити ті самі дії, що й Адміністратор, окрім доступу до Users.

Таблиця 4.2

	Users	Vacations	TimeTracks	SickLeaves
Менеджер	5	5	5	5

Користувачі, які орендують житло мають відповідні права, які наведені нижче

Таблиця 4.3

	Users	Vacations	TimeTracks	SickLeaves
Користувач	0	5	5	0

У **SQL Server** існує три типи ролей :

- заздалегідь визначені ролі;
- ролі, що визначені користувачем ;
- неявні ролі.

Існують два типи ролей рівня бази даних, що визначаються **користувачем**:

- стандартна роль;
- роль рівня додатку.

В нашому випадку доцільно задати стандартні ролі таким користувачам як співробітник та адміністратор

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		6

Для цього використаємо такі **інструкції**:

```
CREATE LOGIN Worker WITH PASSWORD='1';  
USE Timetracker;  
CREATE USER Worker FOR LOGIN Client;  
EXECUTE sp_addrolemember db_datareader, Worker;
```

Для інших користувачів визначимо права доступу використовуючи ролі визначені користувачем та інструкцію **GRANT**:

```
Create role Manager AUTHORIZATION admindb  
go  
create login Manager with password='man'  
go  
CREATE USER Manager FOR LOGIN Manager;  
go  
grant select on timetracks to manager;  
go  
grant select on calendar to manager;  
go  
grant select, insert, update on timetracks to manager;  
go  
grant select, insert, update on calendar to manager;  
  
grant select, insert, update on vacations to manager;
```

Висновки до четвертого розділу

У цьому розділі було розподілено права доступу користувача, а саме контроль за допомогою призначення користувачам певної ролі. Також у цьому розділі було описано, як відбувалося підключення додатку до БД, створення та налаштування підключення до бази даних MSSQL, було наведено віддалені та локальні бази даних MSSQL

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час написання даної курсової роботи було отримано навички роботи з віддаленими та локальними базами даних та перегляд, додавання, редагування, видалення даних з бази даних.

В першому розділі курсової роботи, було проаналізовано три СУБД, визначено їхні переваги та недоліки та обрано одну із них для реалізації додатку курсової роботи.

В другому розділі курсової роботи, було проаналізовано інформаційні процеси, розроблено структуру бази даних, описано логіку роботи з базою даних клієнтом, пояснено як оброблюються дані і зберігаються. Описано, як відбувалась побудова та генерація статичної інформації.

В третьому розділі курсової роботи було написано додаток для моніторингу робочого часу працівника з БД та спроектовано інтерфейс сайту моніторингу робочого часу працівників.

В четвертому розділі курсової роботи було описано надання прав доступу користувача до бази даних, а саме контроль за допомогою ролей та деяких функцій.

В результаті виконання курсової роботи було створено сайт моніторингу робочого часу працівників, який відповідає запланованому функціоналу, умовам та вимогам, поставлених на початку проектування.

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Флэнаган Д. JavaScript. Подробное руководство, 6-е издание – Пер. с англ. – СПб: Символ-Плюс, 2012. – 1080 с.
2. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. – СПб.: Питер, 2012. – 304 с.
3. Справочник по HTML і CSS: <https://metanit.com/web/html5/>
4. Руководство по JS: <https://metanit.com/web/javascript/>
5. Руководство по ASP NET: <https://metanit.com/web/aspnet/>
6. Документация по Fetch API: <https://learn.javascript.ru/fetch>
7. Документация по Axios API: <https://ru.vuejs.org/v2/cookbook/using-axios-to-consume-apis.html>
8. Руководство по Node JS: <https://metanit.com/web/nodejs/>
9. Документація по MSSQL: <https://metanit.com/sql/sqlserver/1.1.php>

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		6

Лістинг коду CalendarRepository:

```

public class CalendarRepository : ICalendarRepository
{
    private readonly string connectionString;
    private readonly IMapper mapper;

    public CalendarRepository(IConfiguration configuration)
    {
        connectionString = configuration.GetConnectionString("MsSql");
        mapper = new MapperSetup().Mapper;
    }

    public void CreateRange(List<CalendarModel> calendarModels)
    {
        string query = @"INSERT INTO Calendar
                        (Date, DayTypeId, HoursToWork)
                        VALUES (@Date, @DayTypeId, @HoursToWork)";

        var calendarDbModels = mapper.Map<List<CalendarDbModel>>(calendarModels);

        var connection = new SqlConnection(connectionString);
        connection.Open();

        var transaction = connection.BeginTransaction();

        try
        {
            transaction.Execute(query, calendarDbModels);
            transaction.Commit();
        }
        catch
        {
            transaction.Rollback();
            throw;
        }
    }

    public int CreateSingle(CalendarModel calendar)
    {
        string query = @"INSERT INTO Calendar
                        (Date, DayTypeId, HoursToWork)
                        OUTPUT Inserted.Id
                        VALUES (@Date, @DayTypeId, @HoursToWork)";

        var calendarDbModel = mapper.Map<CalendarDbModel>(calendar);

        using var connection = new SqlConnection(connectionString);
        return connection.ExecuteScalar<int>(query, calendarDbModel);
    }

    public void DeleteById(int id)
    {
        string query = @"DELETE FROM Calendar
                        WHERE Id = @Id";

        using var connection = new SqlConnection(connectionString);
        connection.Execute(query, new {Id = id});
    }

    public void DeleteByDate(DateTime date)
    {

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        string query = @"DELETE FROM Calendar
                           WHERE Date = @Date";

        using var connection = new SqlConnection(connectionString);
        connection.Execute(query, new {Date = date});
    }

    public void Update(CalendarModel calendarModel)
    {
        string query = @"UPDATE Calendar
                           SET Date = @Date, DayTypeId = @DayTypeId, HoursToWork =
@HoursToWork
                           WHERE Id = @Id";

        var calendarDbModel = mapper.Map<CalendarDbModel>(calendarModel);

        using var connection = new SqlConnection(connectionString);
        connection.Execute(query, calendarDbModel);
    }

    public List<CalendarModel> GetAll()
    {
        string query = @"
                        SELECT
                        C.Id,
                        C.Date,
                        C.HoursToWork,
                        C.DayTypeId,
                        D.Id,
                        D.Name
                        FROM Calendar C
                        LEFT JOIN DayTypes D ON D.Id = C.DayTypeId";

        using var connection = new SqlConnection(connectionString);
        return connection.Query<CalendarModel, DayTypeModel, CalendarModel>(query,
            (calendar, dayType) =>
            {
                calendar.DayType = dayType;
                return calendar;
            }, splitOn: "Id").ToList();
    }

    public List<CalendarModel> GetByRange(DateTime startDate, DateTime endDate)
    {
        string query = @"
                        SELECT
                        C.Id,
                        C.Date,
                        C.HoursToWork,
                        C.DayTypeId,
                        D.Id,
                        D.Name
                        FROM Calendar C
                        LEFT JOIN DayTypes D ON D.Id = C.DayTypeId
                        WHERE C.Date between @StartDate AND @EndDate";

        using var connection = new SqlConnection(connectionString);
        return connection.Query<CalendarModel, DayTypeModel, CalendarModel>(query,
            (calendar, dayType) =>
            {
                calendar.DayType = dayType;
                return calendar;
            }, splitOn: "Id", param: new {StartDate = startDate, EndDate =
endDate}).ToList();
    }
}

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public CalendarModel? GetById(int id)
{
    string query = @"
        SELECT
            C.Id,
            C.Date,
            C.HoursToWork,
            C.DayTypeId,
            D.Id,
            D.Name
        FROM Calendar C
        LEFT JOIN DayTypes D ON D.Id = C.DayTypeId
        WHERE C.Id = @Id";

    using var connection = new SqlConnection(connectionString);

    return connection.Query<CalendarModel, DayTypeModel, CalendarModel>(query,
(calendar, dayType) =>
    {
        calendar.DayType = dayType;
        return calendar;
    }, splitOn: "Id", param: new {Id = id}).FirstOrDefault();
}

public CalendarModel? GetByDate(DateTime date)
{
    string query = @"
        SELECT
            C.Id,
            C.Date,
            C.HoursToWork,
            C.DayTypeId,
            D.Id,
            D.Name
        FROM Calendar C
        LEFT JOIN DayTypes D ON D.Id = C.DayTypeId
        WHERE C.Date = @Date";

    using var connection = new SqlConnection(connectionString);
    return connection.Query<CalendarModel, DayTypeModel, CalendarModel>(query,
(calendar, dayType) =>
    {
        calendar.DayType = dayType;

        return calendar;
    }, splitOn: "Id", param: new {Date = date}).FirstOrDefault();
}
}

```

Лістинг коду OptionsRepository:

```

using Server.Business.Interfaces;
using Server.Business.Entities;
using Microsoft.Data.SqlClient;
using Dapper;
using Microsoft.Extensions.Configuration;
using Server.MSSQL.Models;

```

```

namespace Server.MSSQL.Repositories
{

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public class OptionRepository : IOptionRepository
{
    private readonly string connectionString;

    public OptionRepository(IConfiguration configuration)
    {
        connectionString = configuration.GetConnectionString("MsSql");
    }

    public OptionModel? GetById(int id)
    {
        string query = @"
            SELECT * FROM Options
            WHERE Id = @Id
        ";

        using var connection = new SqlConnection(connectionString);
        return connection.QuerySingleOrDefault<OptionModel>(query, new { Id = id });
    }

    public OptionModel? GetByCode(string code)
    {
        string query = @"
            SELECT * FROM Options
            WHERE Code = @Code
        ";

        using var connection = new SqlConnection(connectionString);
        return connection.QuerySingleOrDefault<OptionModel>(query, new { Code = code
    });
}

    public IEnumerable<OptionModel> GetAll()
    {
        string query = @"
            SELECT * FROM Options
        ";

        using var connection = new SqlConnection(connectionString);
        return connection.Query<OptionModel>(query);
    }

    public IEnumerable<OptionModel> GetAllByUserId(int userId)
    {
        string query = @"
            SELECT * FROM OptionsUsers
            INNER JOIN Options ON OptionsUsers.OptionId = Options.Id
            WHERE UserId = @UserId
        ";

        using var connection = new SqlConnection(connectionString);
        return connection.Query<OptionsUsersDbModel, OptionModel, OptionModel>(query,
(optionsUsers, option) => option, param: new { UserId = userId});
    }

    public void RemoveAllForUserId(int userId)
    {
        string query = @"
            DELETE FROM OptionsUsers
            WHERE UserId = @UserId
        ";

        using var connection = new SqlConnection(connectionString);
        connection.Execute(query, new { UserId = userId});
    }
}

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public void AddListForUserId(IEnumerable<OptionModel> optionModels, int userId)
{
    foreach (var optionModel in optionModels)
    {
        AddToUser(optionModel.Id, userId);
    }
}

public void AddToUser(int optionId, int userId)
{
    string query = @"
        INSERT INTO OptionsUsers
        (UserId, OptionId)
        VALUES (@UserId, @OptionId)
    ";

    using var connection = new SqlConnection(connectionString);
    connection.Execute(query, new {UserId = userId, OptionId = optionId});
}

public void RevokeFromUser(int optionId, int userId)
{
    string query = @"
        DELETE FROM OptionsUsers
        WHERE UserId = @UserId AND OptionId = @OptionId
    ";

    using var connection = new SqlConnection(connectionString);
    connection.Execute(query, new {UserId = userId, OptionId = optionId});
}
}

```

Лістинг коду SickLeaveRepository:

```

using Server.Business.Interfaces;
using Server.Business.Entities;
using Microsoft.Data.SqlClient;
using Dapper;
using Microsoft.Extensions.Configuration;

namespace Server.MSSQL.Repositories;

public class SickLeaveRepository : ISickLeaveRepository
{
    private readonly string connectionString;

    public SickLeaveRepository(IConfiguration configuration)
    {
        connectionString = configuration.GetConnectionString("MsSql");
    }

    public SickLeaveModel? GetById(int id)
    {
        string query = @"
            SELECT * FROM SickLeaves
            WHERE Id = @Id
        ";

        using var connection = new SqlConnection(connectionString);
        return connection.QueryFirstOrDefault<SickLeaveModel>(query, new {Id = id});
    }
}

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		


```

public SickLeaveModel? GetByUserIdAndDate(int userId, DateTime date)
{
    string query = @"
        SELECT * FROM SickLeaves
        WHERE UserId = @UserId AND @Date BETWEEN StartDate AND EndDate
    ";

    using var connection = new SqlConnection(connectionString);
    return connection.QueryFirstOrDefault<SickLeaveModel>(query, new {UserId =
userId, Date = date});
}

public IEnumerable<SickLeaveModel> GetAllByUserId(int userId)
{
    string query = @"
        SELECT * FROM SickLeaves
        WHERE UserId = @UserId
    ";

    using var connection = new SqlConnection(connectionString);
    return connection.Query<SickLeaveModel>(query, new {UserId = userId});
}

public int Create(SickLeaveModel sickLeaveModel)
{
    string query = @"
        INSERT INTO SickLeaves
        (StartDate, EndDate, UserId)
        OUTPUT INSERTED.Id
        VALUES(@StartDate, @EndDate, @UserId)
    ";

    using var connection = new SqlConnection(connectionString);
    return connection.ExecuteScalar<int>(query, sickLeaveModel);
}

public void Delete(int id)
{
    string query = @"
        DELETE FROM SickLeaves
        WHERE Id = @Id
    ";

    using var connection = new SqlConnection(connectionString);
    connection.Execute(query, new {Id = id});
}

public void Update(SickLeaveModel sickLeaveModel)
{
    string query = @"
        UPDATE SickLeaves
        SET StartDate = @StartDate, EndDate = @EndDate
        WHERE Id = @Id";

    using var connection = new SqlConnection(connectionString);
    connection.Execute(query, sickLeaveModel);
}
}

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг коду SickLeaveRepository:

```
using AutoMapper;
using Dapper;
using Microsoft.Data.SqlClient;
using Microsoft.Extensions.Configuration;
using Server.Business.Entities;
using Server.Business.Interfaces;
using Server.MSSQL.Mapper;
using Server.MSSQL.Models;

namespace Server.MSSQL.Repositories
{
    public class TimeTrackRepository : ITimeTrackRepository
    {
        private readonly string connectionString;
        private readonly IMapper mapper;

        public TimeTrackRepository(IConfiguration configuration)
        {
            connectionString = configuration.GetConnectionString("MsSql");
            mapper = new MapperSetup().Mapper;
        }

        public string CommonTimeTrackQueryString = @"
SELECT
    T.Id,
    T.StartDate,
    T.EndDate,
    T.TotalTime,
    T.UserId,
    T.TimeTrackTypeId,
    T.CreationTypeId,
    TT.Id,
    TT.Name,
    CT.Id,
    CT.Name,
    TU.Id,
    TU.TimeTrackId,
    TU.UpdatorId,
    TU.Date,
    U.*
FROM TimeTracks T
LEFT JOIN TimeTrackTypes TT ON T.TimeTrackTypeId = TT.Id
LEFT JOIN CreationTypes CT ON CT.Id = T.CreationTypeId
LEFT JOIN TimeTracksUpdateHistory TU ON TU.TimeTrackId = T.Id
LEFT JOIN Users U ON TU.UpdatorId = U.Id";

        public int Start(TimeTrackModel timeTrackModel)
        {

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const string query = @"
    INSERT INTO TimeTracks
    (StartDate, EndDate, UserId, TotalTime, TimeTrackTypeId, CreationTypeId)
    OUTPUT Inserted.Id
    VALUES(@StartDate, @EndDate, @UserId, @TotalTime, @TimeTrackTypeId,
@CreationTypeId)
";

var timeTrackDbModel = mapper.Map<TimeTrackDbModel>(timeTrackModel);

using var connection = new SqlConnection(connectionString);
return connection.ExecuteScalar<int>(query, timeTrackDbModel);
}

public void Stop(TimeTrackModel timeTrackModel)
{
    const string query = @"
        UPDATE TimeTracks
        SET
            UserId = @UserId, StartDate = @StartDate,
            EndDate = @EndDate, TotalTime = @TotalTime,
            TimeTrackTypeId = @TimeTrackTypeId,
            CreationTypeId = @CreationTypeId
        WHERE Id = @Id
    ";

    var timeTrackDbModel = mapper.Map<TimeTrackDbModel>(timeTrackModel);

    using var connection = new SqlConnection(connectionString);
    connection.Execute(query, timeTrackDbModel);
}

public TimeTrackModel GetById(int id)
{
    string query = CommonTimeTrackQueryString + " " + "WHERE T.Id = @Id";

    using var connection = new SqlConnection(connectionString);
    return QueryTimeTracks(query, new { Id = id }).FirstOrDefault();
}

public TimeTrackModel GetLastByUserId(int userId)
{
    string query = CommonTimeTrackQueryString + " " + @"WHERE T.UserId = @UserId
        AND T.EndDate IS NULL
        ORDER BY T.StartDate DESC";

    using var connection = new SqlConnection(connectionString);
    return QueryTimeTracks(query, new { UserId = userId }).FirstOrDefault();
}

public List<TimeTrackModel> GetListByUserId(int userId)
{

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        string query = CommonTimeTrackQueryString + " " + @"WHERE T.UserId = @UserId AND
T.EndDate IS NOT NULL

                                ORDER BY T.StartDate DESC";

        using var connection = new SqlConnection(connectionString);
        return QueryTimeTracks(query, new {UserId = userId}).ToList();
    }

    public List<TimeTrackModel> GetAll()
    {
        string query = CommonTimeTrackQueryString + " " + @"WHERE T.EndDate IS NOT NULL
ORDER BY T.StartDate DESC";

        using var connection = new SqlConnection(connectionString);
        return QueryTimeTracks(query, null).ToList();
    }

    public void Update(TimeTrackModel timeTrackModel)
    {
        string query = @"
        UPDATE TimeTracks
        SET
            StartDate = @StartDate,
            EndDate = @EndDate,
            TotalTime = @TotalTime,
            TimeTrackTypeId = @TimeTrackTypeId
        WHERE Id = @Id";

        var timeTrackDbModel = mapper.Map<TimeTrackDbModel>(timeTrackModel);

        using var connection = new SqlConnection(connectionString);
        connection.Execute(query, timeTrackDbModel);
    }

    public List<TimeTrackModel> GetByDateRange(DateTime startDate, DateTime endDate)
    {
        string query = CommonTimeTrackQueryString + " " + @"WHERE T.StartDate BETWEEN
@StartDate AND @EndDate

                                ORDER BY T.StartDate DESC";

        var variables = new
        {
            StartDate = startDate,
            EndDate = endDate
        };

        using var connection = new SqlConnection(connectionString);
        return QueryTimeTracks(query, variables).ToList();
    }

    public List<TimeTrackModel> GetByDateRangeAndUserId(int userId, DateTime startDate,
DateTime endDate)

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    string query = CommonTimeTrackQueryString + " " + @"WHERE T.StartDate between
@StartDate and @EndDate
                                AND T.UserId = @UserId AND T.EndDate IS NOT NULL
                                ORDER BY T.StartDate DESC";

    var variables = new
    {
        UserId = userId,
        StartDate = startDate,
        EndDate = endDate
    };

    using var connection = new SqlConnection(connectionString);
    return QueryTimeTracks(query, variables).ToList();
}

public void DeleteById(int id)
{
    const string query = @"
        DELETE FROM TimeTracks
        WHERE Id = @Id
    ";

    using var connection = new SqlConnection(connectionString);
    connection.Execute(query, new { Id = id });
}

public int Create(TimeTrackModel timeTrackModel)
{
    const string query = @"
        INSERT INTO TimeTracks
        (StartDate, EndDate, UserId, TotalTime, TimeTrackTypeId, CreationTypeId)
        OUTPUT Inserted.Id
        VALUES(@StartDate, @EndDate, @UserId, @TotalTime, @TimeTrackTypeId,
@CreationTypeId)
    ";

    var timeTrackDbModel = mapper.Map<TimeTrackDbModel>(timeTrackModel);

    using var connection = new SqlConnection(connectionString);
    return connection.ExecuteScalar<int>(query, timeTrackDbModel);
}

public void CreateCollection(ICollection<TimeTrackModel> timeTrackModels)
{
    const string query = @"
        INSERT INTO TimeTracks
        (StartDate, EndDate, UserId, TotalTime, TimeTrackTypeId, CreationTypeId)
        VALUES(@StartDate, @EndDate, @UserId, @TotalTime, @TimeTrackTypeId,
@CreationTypeId)
    ";

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

var timeTrackDbModels = mapper.Map<List<TimeTrackDbModel>>(timeTrackModels);

using var connection = new SqlConnection(connectionString);
connection.ExecuteScalar(query, timeTrackDbModels);
}

public TimeTrackModel? CheckForSameTimeTrack(DateTime startDate, DateTime endDate, int
userId)
{
    string query = CommonTimeTrackQueryString + " " + @"WHERE T.UserId = @UserId
        AND T.StartDate = @StartDate
        AND T.EndDate = @EndDate
        ORDER BY T.Id DESC";

    var variables = new
    {
        StartDate = startDate,
        EndDate = endDate,
        UserId = userId
    };

    using var connection = new SqlConnection(connectionString);
    return QueryTimeTracks(query, variables).FirstOrDefault();
}

public List<TimeTrackModel> ValidateTimeTrack(DateTime startDate, DateTime endDate, int
userId)
{
    const string firstClause = @"
        WHERE T.UserId = @UserId
        AND @StartDate BETWEEN T.StartDate AND T.EndDate
        AND @EndDate BETWEEN T.StartDate AND T.EndDate
    ";

    const string secondClause = @"
        WHERE T.UserId = @UserId
        AND T.StartDate BETWEEN @StartDate AND @EndDate
        OR T.EndDate BETWEEN @StartDate AND @EndDate
        AND T.UserId = @UserId
    ";

    const string thirdClause = @"
        WHERE T.UserId = @UserId
        AND @StartDate BETWEEN T.StartDate AND T.EndDate
        OR @EndDate BETWEEN T.StartDate AND T.EndDate
        AND T.UserId = @UserId
    ";

    List<string> clauses = new List<string>() {firstClause, secondClause, thirdClause};
    List<string> queries = new List<string>();

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

foreach (var clause in clauses)
{
    queries.Add(CommonTimeTrackQueryString + " " + clause);
}

var variables = new
{
    StartDate = startDate,
    EndDate = endDate,
    UserId = userId
};

List<TimeTrackModel> result = new List<TimeTrackModel>();

foreach (var query in queries)
{
    var queryResult = QueryTimeTracks(query, variables).ToList();
    result.AddRange(queryResult);
}

return result;
}

private IEnumerable<TimeTrackModel> QueryTimeTracks(string query, object param)
{
    using var connection = new SqlConnection(connectionString);

    return connection.Query<GetTimeTrackModel, TimeTrackTypeModel, CreationTypeModel,
        TimeTrackUpdateHistoryModel, UserModel, TimeTrackModel>(query,
        (timeTrack, timeTrackType, creationType, timeTrackUpdateHistoryModel, userModel) =>
        {
            if (userModel != null)
            {
                timeTrackUpdateHistoryModel.User = userModel;
            }

            string? totalTime = timeTrack.TotalTime?.ToString();

            return new TimeTrackModel()
            {
                Id = timeTrack.Id,
                UserId = timeTrack.UserId,
                CreationType = creationType,
                StartDate = timeTrack.StartDate,
                EndDate = timeTrack.EndDate,
                TimeTrackType = timeTrackType,
                TimeTrackUpdateHistory = timeTrackUpdateHistoryModel,
                TotalTime = totalTime
            };
        }, splitOn: "Id", param: param);
}
}

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

}

Лістинг коду UserRepository:

```
using System.Runtime.CompilerServices;
using System.Text;
using AutoMapper;
using Server.Business.Interfaces;
using Server.Business.Entities;
using Microsoft.Data.SqlClient;
using Dapper;
using Dapper.Transaction;
using DapperQueryBuilder;
using Microsoft.Extensions.Configuration;
using Server.MSSQL.Mapper;
using Server.MSSQL.Models;
using Server.MSSQL.Utilities;
using static Server.MSSQL.Utilities.FormatUtilities;

namespace Server.MSSQL.Repositories;

public class UserRepository : IUserRepository
{
    private readonly string connectionString;
    private readonly IMapper mapper;

    public UserRepository(IConfiguration configuration)
    {
        connectionString = configuration.GetConnectionString("MsSql");
        mapper = new MapperSetup().Mapper;
    }

    public int Create(UserModel userModel)
    {
        const string insertUserQuery = @"
        INSERT INTO Users
        (
            Email,
            Firstname,
            Lastname,
            Password,
            VacationDays,
            EmploymentDate,
            WorkTypeId
        )
        OUTPUT INSERTED.Id
        VALUES(
            @Email,
            @Firstname,
            @Lastname,
            @Password,
            30,
            @EmploymentDate,
```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        @WorkTypeId
    )
";

const string insertOptionsUsersQuery = @"
    INSERT INTO OptionsUsers
    (UserId, OptionId)
    VALUES (@UserId, @OptionId)
";

const string insertApproveVacationUserQuery = @"
    INSERT INTO ApproveVacationUser
    (UserId, ApproverId)
    VALUES (@UserId, @ApproverId)
";

var userDbModel = mapper.Map<UserDbModel>(userModel);

List<int> approversIds = new List<int>();
if (userModel.VacationApprovers.Count != 0)
{
    approversIds = GetApproversIds(userModel.VacationApprovers); // WARNING: Dont replace
this line under
                                // transaction because of QueryBuilder,
                                // same in Update method
}

using var connection = new SqlConnection(connectionString);
connection.Open();

using var transaction = connection.BeginTransaction();

try
{
    var createdUserId = transaction.ExecuteScalar<int>(insertUserQuery, userDbModel);

    if (userModel.Options != null && userModel.Options.Count != 0)
    {
        var optionsUserDbModels =
            mapper.Map<List<OptionsUsersDbModel>>(
                userModel.Options,
                opt => opt.Items["UserId"] = createdUserId
            );

        transaction.Execute(insertOptionsUsersQuery, optionsUserDbModels);
    }

    if (approversIds.Count != 0)
    {
        var approveVacationUserModels = GetApproveVacationUserModels(approversIds,
createdUserId);
        transaction.Execute(insertApproveVacationUserQuery, approveVacationUserModels);
    }
}

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    transaction.Commit();
    return createdUserId;
}
catch (Exception)
{
    transaction.Rollback();
    throw;
}
}

public void SetActivationStatus(int id, bool isActivated)
{
    string query = @"
        UPDATE Users
        SET IsActivated = @IsActivated
        WHERE Id = @Id
    ";

    using var connection = new SqlConnection(connectionString);
    connection.Execute(query, new { Id = id, IsActivated = isActivated });
}

public UserModel? GetByEmail(string email)
{
    const string query = @"
        SELECT
            U.*, W.*, O.*
        FROM Users U
        LEFT JOIN WorkTypes W ON (W.Id = U.WorkTypeId)
        LEFT JOIN OptionsUsers OU ON (OU.UserId = U.Id)
        LEFT JOIN Options O ON (O.Id = OU.OptionId)
        WHERE U.Email = @Email
    ";

    using var connection = new SqlConnection(connectionString);
    return connection.QueryUsers(query, new { Email = email }).SingleOrDefault();
}

public UserModel? GetById(int id)
{
    const string query = @"
        SELECT
            U.*, W.*, O.*
        FROM Users U
        LEFT JOIN WorkTypes W ON (W.Id = U.WorkTypeId)
        LEFT JOIN OptionsUsers OU ON (OU.UserId = U.Id)
        LEFT JOIN Options O ON (O.Id = OU.OptionId)
        WHERE U.Id = @Id
    ";

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using var connection = new SqlConnection(connectionString);
return connection.QueryUsers(query, new { Id = id }).SingleOrDefault();
}

public IEnumerable<UserModel> GetAll(
    out int numRecordsUsers,
    PaginationModel? paginationModel = null,
    List<SortModel>? sortModels = null,
    List<FilterModel>? filterModels = null,
    SearchModel? searchModel = null
)
{
    using var connection = new SqlConnection(connectionString);

    FormattableString paginationSql;
    FormattableString sortSql;

    if (paginationModel != null)
    {
        paginationSql = $"@\"
            ORDER BY Firstname
            OFFSET {paginationModel.Offset} ROWS
            FETCH NEXT {paginationModel.Next} ROWS ONLY
        \";
    }
    else
    {
        paginationSql = $"";
    }

    if (sortModels != null && sortModels.Any())
    {
        sortSql = GetUserFormatSort(sortModels);
    }
    else
    {
        sortSql = $"ORDER BY Firstname ASC";
    }

    var usersQueryBuilder = connection.QueryBuilder($"@\"
        SELECT
            U.*, W.*, O.*
        FROM (
            SELECT * FROM Users
            /**where**/
            {paginationSql}
        ) as U
        LEFT JOIN WorkTypes W ON (W.Id = U.WorkTypeId)
        LEFT JOIN OptionsUsers OU ON (OU.UserId = U.Id)
        LEFT JOIN Options O ON (O.Id = OU.OptionId)
        {sortSql}
    \");

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

var numRecordsQueryBuilder = connection.QueryBuilder($"@ "
    SELECT COUNT(*) as UsersRecords
    FROM Users
    /**where**/
");

if (filterModels != null && filterModels.Any())
{
    usersQueryBuilder.Where(GetFormatFilters(filterModels));
    numRecordsQueryBuilder.Where(GetFormatFilters(filterModels));
}

if (searchModel != null)
{
    usersQueryBuilder.Where(GetUserFormatSearch(searchModel));
    numRecordsQueryBuilder.Where(GetUserFormatSearch(searchModel));
}

numRecordsUsers = numRecordsQueryBuilder.Query<int>().Single();

return connection.QueryUsers(usersQueryBuilder.Sql, usersQueryBuilder.Parameters).ToList();
}

public void Update(UserModel userModel)
{
    const string queryUpdateUser = @"
        UPDATE Users
        SET
            Email = @Email,
            Firstname = @Firstname,
            Lastname = @Lastname,
            Password = @Password,
            VacationDays = @VacationDays,
            EmploymentDate = @EmploymentDate,
            WorkTypeId = @WorkTypeId
        WHERE Id = @Id
    ";

    const string queryUpdateOptionsUsers = @"
        IF NOT EXISTS (SELECT * FROM OptionsUsers WHERE UserId = @UserId AND
OptionId = @OptionId)
        BEGIN
            INSERT INTO OptionsUsers
            (UserId, OptionId)
            VALUES (@UserId, @OptionId)
        END
    ";

    const string queryUpdateApproveVacationUser = @"
        INSERT INTO ApproveVacationUser
        (UserId, ApproverId)
    ";

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сугоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

VALUES (@UserId, @ApproverId)
";

const string queryOptionsUsers = @"
SELECT * FROM OptionsUsers
WHERE UserId = @UserId
";

const string queryDeleteOptionsUsers = @"
DELETE FROM OptionsUsers
WHERE UserId = @UserId AND @OptionId = OptionId
";

const string queryDeleteApproveVacationUser = @"
DELETE FROM ApproveVacationUser
WHERE UserId = @UserId AND ApproverId = @ApproverId";

List<ApproveVacationUserModel> approveVacationsUserModels = new
List<ApproveVacationUserModel>();

if (userModel.VacationApprovers.Count != 0)
{
    var approversIds = GetApproversIds(userModel.VacationApprovers);
    approveVacationsUserModels = GetApproveVacationUserModels(approversIds,
userModel.Id);
}

var existsApproveVacationUserForDelete = GetExistingApproveVacationUser(userModel.Id);

var userDbModel = mapper.Map<UserDbModel>(userModel);

using var connection = new SqlConnection(connectionString);
connection.Open();

using var transaction = connection.BeginTransaction();

var existsOptionsUserForDelete =
    transaction.Query<OptionsUsersDbModel>(queryOptionsUsers, new { UserId =
userModel.Id }).ToList();

try
{
    transaction.Execute(queryUpdateUser, userDbModel);

    if (userModel.Options != null && userModel.Options.Count != 0)
    {
        foreach (var option in userModel.Options)
        {
            existsOptionsUserForDelete.RemoveAll(opt => opt.OptionId == option.Id);
        }

        var optionsUserDbModels =

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        mapper.Map<List<OptionsUsersDbModel>>(userModel.Options,
            opt => { opt.Items["UserId"] = userModel.Id; });

        transaction.Execute(queryUpdateOptionsUsers, optionsUserDbModels);
    }

    transaction.Execute(queryDeleteApproveVacationUser,
        existsApproveVacationUserForDelete);

    if (userModel.VacationApprovers.Count != 0)
    {
        transaction.Execute(queryUpdateApproveVacationUser, approveVacationsUserModels);
    }

    transaction.Execute(queryDeleteOptionsUsers, existsOptionsUserForDelete);

    transaction.Commit();
}
catch (Exception)
{
    transaction.Rollback();
    throw;
}
}

private List<ApproveVacationUserModel> GetExistingApproveVacationUser(int userId)
{
    const string query = @"
        SELECT * FROM ApproveVacationUser
        WHERE UserId = @UserId";

    using var connection = new SqlConnection(connectionString);
    return connection.Query<ApproveVacationUserModel>(query, new { UserId = userId }).ToList();
}

public List<int> GetApproversIds(List<string> approversEmails)
{
    using var connection = new SqlConnection(connectionString);

    var query = connection.QueryBuilder($"@"
        SELECT Id FROM Users
        /**where**/"
    );

    var filters = new Filters(Filters.FiltersType.OR);

    foreach (string email in approversEmails)
    {
        var formatString = FormattableStringFactory.Create("Email = {0}", email);
        filters.Add(new Filter(formatString));
    }
}

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        query.Where(filters);

        return connection.Query<int>(query.Sql, query.Parameters).ToList();
    }

    private List<ApproveVacationUserModel> GetApproveVacationUserModels(List<int>
    approversIds, int userId)
    {
        List<ApproveVacationUserModel> approveVacationUserModels = new
        List<ApproveVacationUserModel>();

        foreach (int approverId in approversIds)
        {
            approveVacationUserModels.Add(
                new ApproveVacationUserModel()
                {
                    UserId = userId,
                    ApproverId = approverId
                }
            );
        }

        return approveVacationUserModels;
    }
}

```

		Грибан В.В			ДУ «Житомирська політехніка».23.121.08.000 - ІПЗ	Арк.
		Сухоняк І.І				6
Змн.	Арк.	№ докум.	Підпис	Дата		