

ЗМІСТ

1	ВСТУП.....	2
2	ОПИС ТЕХНОЛОГІЙ	3
3	СХЕМА БАЗИ ДАНИХ	7
3.1.	Загальний опис бази даних.....	7
3.2.	Опис таблиць	7
3.3.	Можливості бази даних	9
4	ЗАПИТИ ДО БАЗИ ДАНИХ	10
5	ІНСТРУКЦІЯ КОРИСТУВАЧА.....	20
	ВИСНОВКИ.....	23
	ПЕРЕЛІК ПОСИЛАНЬ	24

1 ВСТУП

Керування альпіністською базою — це складний процес, який потребує багатьох різноматних бухгалтерських звітів по прибуткам компанії, потребує детального спостереження за реальним станом високогірних маршрутів, і звісно бажано збирати певну статистику по популярності послуг.

А тепер уявіть що у вас є декілька альпіністських баз в різних країнах і всіма цими базами ви хочете управляти дистанційно.

Ви звісно можете скористатись електроною поштою, на яку вам присилатимуть різноманітні звіти, які вам потім доведеться довго і нудно перечитувати, щоб знайти якусь корисну інформацію.

Але є інший вибір: можна використати зручний веб-додаток, який сам збиратиме всю цю інформацію і буде вам демонструвати в зручному і доступному вигляді.

Тим паче додатком зможете користуватись не лише ви, а й ваші менеджери, яких цікавить інформація про популярність послуг та високогірних маршрутів, ваші бухгалтери, які тепер зможуть рахувати ваші прибутки в кілька кліків, і тепер вашому начальнику рятувальної служби буде легко слідкувати за станом маршрутів і вчасно їх ремонтувати.

Дана курсова робота значно полегшить процес бухгалтерського обліку, адміністрування та керування альпіністською базою.

Для вирішення даної задачі необхідно розробити зручний для користувача інтерфейс, механізми взаємодії з користувачем та систему статистики.

Для реалізації цього веб-проекту будуть використовуватись ASP.NET MVC для реалізації взаємодії між користувачем і базою даних, СУБД Microsoft SQL Server для зберігання різноманітних даних стосовно альпіністської бази. Razor/HTML/CSS/JS — для створення інтерфейсу користувача, бібліотека Bootstrap для створення гарного та акуратного дизайну та бібліотека Chart.js для створення різноманітних діаграм, гістограм та графіків.

2 ОПИС ТЕХНОЛОГІЙ

Microsoft SQL Server

Microsoft SQL Server — комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. 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.

Завдяки інтеграції Microsoft SQL Server з середовищем розробки Microsoft Visual Studio дозволяє швидко розробляти додатки з використанням бази даних з широкими можливостями, які забезпечують поліпшену безпеку сховищ і швидке розгортання.

Microsoft SQL Server містить інтегроване середовище для управління SQL інфраструктурою SQL Server Management Studio (SSMS).

SQL Server Management Studio у своєму складі містить засоби для створення баз даних, роботи з інформацією баз даних, перенесення даних з інших систем і в інші системи, резервного копіювання та відновлення даних, розвинену систему транзакцій, систему реплікації даних, реляційну підсистему для аналізу, оптимізації та виконання запитів клієнтів, систему безпеки для управління правами доступу до об'єктів бази даних та ін.

SQL Server будується за структурою таблиці на основі рядків, а також з'єднує пов'язані елементи даних у різних таблицях, уникаючи необхідності резервного зберігання даних у кількох місцях в межах бази даних. Реляційна модель дотримує принцип ACID (атомарність, узгодженість, ізолюваність, довговічність), тобто гарантує надійну роботу транзакцій бази даних.

Основним компонентом Microsoft SQL Server є SQL Server Database Engine, який керує збереженням, обробкою даних та безпекою даних. Цей компонент

включає в себе реляційний двигун, який обробляє команди та запити, а також двигун зберігання, який керує файлами, таблицями, сторінками, індексами, буферами даних та транзакціями. Збережені процедури, активатори, перегляди та інші об'єкти бази даних також створюються та виконуються за допомогою двигуна бази даних.

ASP.NET MVC

Платформа ASP.NET MVC представляє собою фреймворк для створення сайтів і веб-додатків за допомогою реалізації паттерна MVC.

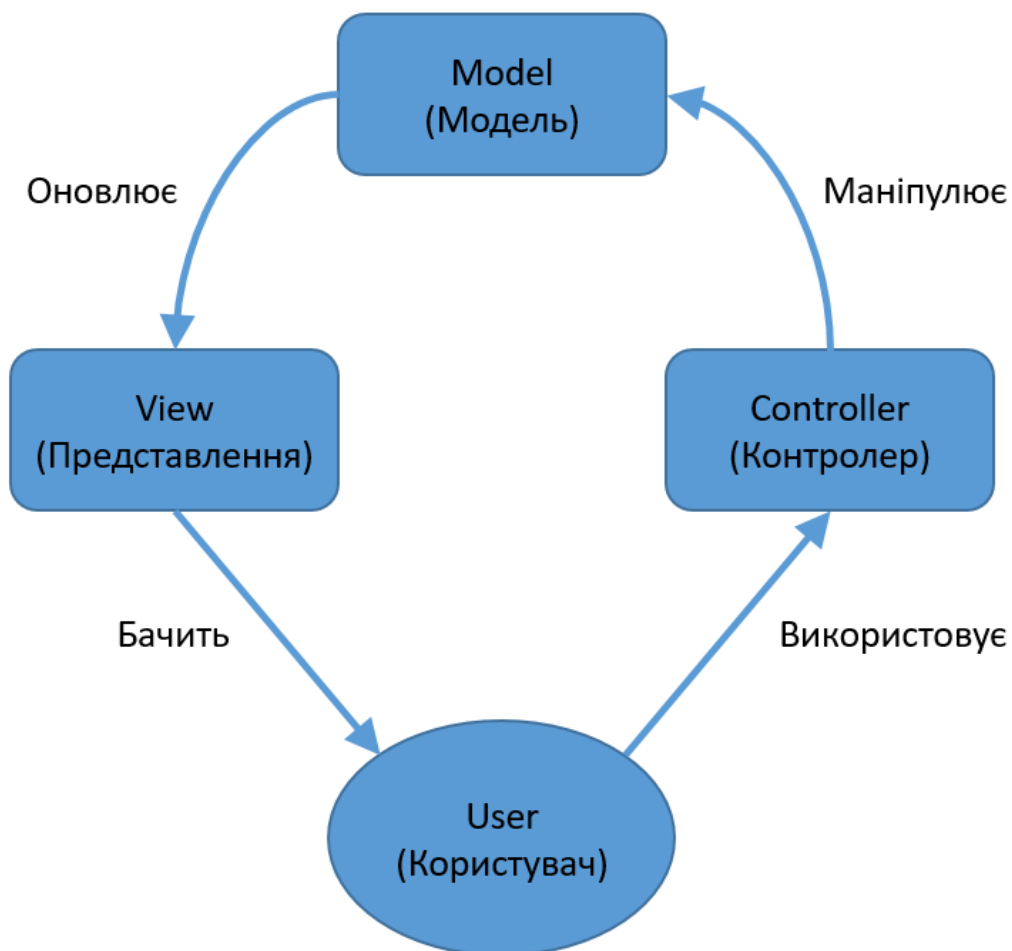


Рисунок 2.1 – Стандартна модель MVC

Концепція паттерну (шаблону) MVC (model-view-controller) передбачає розділення додатку на три компоненти. Завдяки цьому реалізується концепція

розподілу відповідальності, у зв'язку з чим легше побудувати роботу над окремими компонентами.

- **Модель.** Об'єкти моделей є частинами програми, що реалізують логіку для домену даних додатків. Об'єкти моделі часто отримують і зберігають стан моделі в базі даних. У невеликих додатках ця модель передбачає концептуальне, а не фізичне розділення. Наприклад, якщо додаток тільки зчитує набір даних і відправляє його в представлення, то фізичного шару моделі та пов'язаних класів немає. У цьому випадку набір даних приймає роль об'єкта моделі
- **Представлення.** Представлення служать для відображення користувацького інтерфейсу програми. Користувацький інтерфейс зазвичай створюється на основі даних моделі.
- **Контролер.** Контролери здійснюють взаємодію з користувачем, роботу з моделлю, а також вибір представлення, що відображає користувацький інтерфейс. В додатку MVC представлення тільки відображають дані, а контролер обробляє введені дані і відповідає на дії користувача. Наприклад, контролер може обробляти рядкові значення запиту та передавати їх в модель, яка може використовувати ці значення для відправки запиту в базу даних.

Платформа ASP.NET MVC дозволяє використовувати Entity Framework — ORM(object-relational mapper), який дозволяє розробникам .NET працювати з базою даних, що використовує об'єкти .NET.

Entity Framework підтримує підхід "Code first", який передбачає збереження або вилучення інформації з БД на SQL Server без створення схем бази даних або використання дизайнера в Visual Studio. Навпаки, ми створюємо звичайні класи, а Entity Framework сама визначає, як і де зберігати об'єкти цих класів. Але звичайний підхід підключення готової бази даних також підтримується

Razor

Razor — це синтаксис програмування ASP.NET, який використовується для створення динамічних веб-сторінок з мовами програмування C# або Visual Basic .NET. Razor — це простий синтаксичний механізм перегляду, який був випущений як частина MVC 3 і набору інструментів WebMatrix.

Ідея Razor полягає в тому, щоб забезпечити оптимізований синтаксис для генерації HTML з використанням підходу, орієнтованого на кодову схему, з мінімальним переходом між HTML і кодом. Конструкція зменшує кількість символів і натискань клавіш і дозволяє використовувати більш гнучкий робочий процес кодування, не вимагаючи явно позначених блоків сервера в HTML-кодi.

Для простішого створення використовується бібліотека CSS стилів Bootstrap.

3 СХЕМА БАЗИ ДАНИХ

3.1. Загальний опис бази даних

Для створення бази даних програмного забезпечення була обрана СУБД MS SQL Server, оскільки для додатку ASP.NET MVC в цій СУБД реалізовано багато можливостей.

База даних була створена з метою зберігання інформації про альпіністську базу, альпіністів, замовлень харчування, замовлень проживання, маршрутів та їх станів.

Схема бази даних наведена на рисунку 3.1.

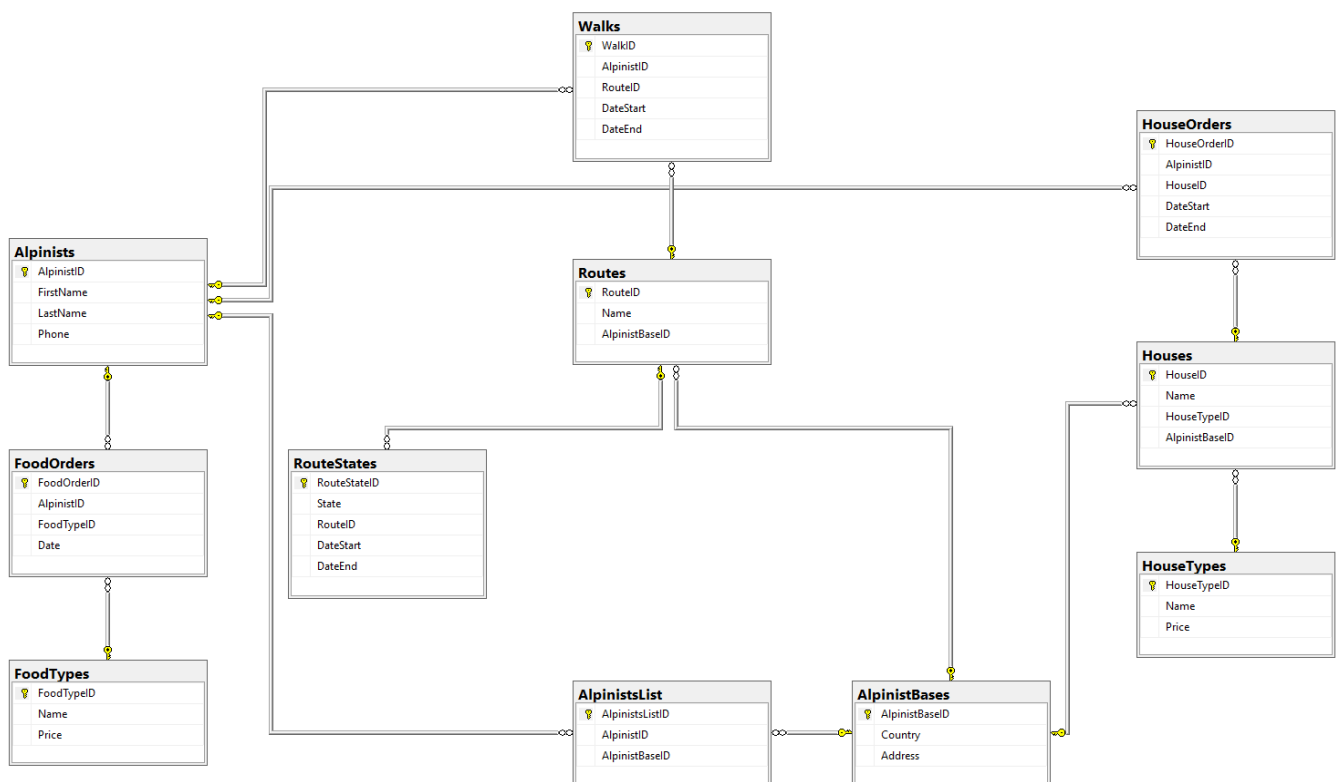


Рисунок 3.1 – Схема бази даних

3.2. Опис таблиць

- **AlpinistBases** – містить дані про альпіністські бази. Поля таблиці: унікальний ідентифікатор альпіністської бази, країна, адреса альпіністської бази.
- **Alpinists** – містить дані про альпіністів. Поля таблиці: унікальний ідентифікатор альпініста, ім'я та прізвище альпініста, телефон.

- **AlpinistsList** – містить данні про належність альпініста до альпіністської бази. Поля таблиці: унікальний ідентифікатор, унікальний ідентифікатор альпініста, унікальний ідентифікатор альпіністської бази.
- **FoodTypes** – містить данні про тип харчування. Поля таблиці: унікальний ідентифікатор типу харчування, ім'я типу, ціна.
- **FoodOrders** – містить данні про замовлення харчування. Поля таблиці: унікальний ідентифікатор замовлення харчування, унікальний ідентифікатор альпініста, унікальний ідентифікатор типу харчування, дата замовлення.
- **Routes** – містить данні про маршрути. Поля таблиці: унікальний ідентифікатор маршруту, ім'я маршруту, унікальний ідентифікатор альпіністської бази.
- **RouteStates** – містить данні про стани маршрутів. Поля таблиці: унікальний ідентифікатор стану маршруту, стан маршруту, унікальний ідентифікатор маршруту, дата початку стану, дата кінця стану.
- **Walks** – містить данні про прогулянки. Поля таблиці: унікальний ідентифікатор прогулянки, унікальний ідентифікатор альпініста, унікальний ідентифікатор маршруту, дата початку прогулянки, дата кінця прогулянки.
- **HouseTypes** – містить данні про типи будинків. Поля таблиці: унікальний ідентифікатор типу будинку, ім'я типу будинку, ціна проживання.
- **Houses** – містить данні про будинки. Поля таблиці: унікальний ідентифікатор будинку, ім'я будинку, унікальний ідентифікатор типу будинку, унікальний ідентифікатор альпіністської бази.
- **HouseOrders** – містить данні про замовлення проживання. Поля таблиці: унікальний ідентифікатор замовлення проживання, унікальний ідентифікатор альпініста, унікальний ідентифікатор будинку, дата початку проживання, дата кінця проживання.

3.3. Можливості бази даних

Створена база даних реалізує наступні можливості для програмного забезпечення:

- Зберігання та отримання повної інформації про альпіністів
- Зберігання та отримання повної інформації про замовлення харчування
- Зберігання та отримання повної інформації про замовлення проживання
- Зберігання та отримання повної інформації про прогулянки
- Зберігання та модифікація інформації про стан маршруту
- Отримання інформації про маршрути, типи будинків, харчування

4 ЗАПИТИ ДО БАЗИ ДАНИХ

4.1. Вибірка загальної інформації про всі замовлення за певний період.

Запит:

```
SELECT
    [FirstName],
    [LastName],
    [DateStart],
    [DateEnd],
    [HT].[Name],
    [Price]*(DateDiff(DD,[DateStart],[DateEnd])+1) AS [Price]
FROM [HouseOrders] AS [HO]
JOIN [Alpinists] AS [A]
    ON [A].[AlpinistID] = [HO].[AlpinistID]
JOIN [Houses] AS [H]
    ON [HO].[HouseID] = [H].[HouseID]
JOIN [HouseTypes] AS [HT]
    ON [H].[HouseTypeID] = [HT].[HouseTypeID]
WHERE ([DateStart] BETWEEN '2017-01-23' AND '2017-09-24') AND ([DateEnd] BETWEEN '2017-01-
23' AND '2017-09-24')
UNION
SELECT
    [FirstName],
    [LastName],
    [Date] AS [DateStart],
    NULL AS [DateEnd],
    [Name],
    [Price]
FROM [FoodOrders] AS [FO]
JOIN [Alpinists] AS [A]
    ON [A].[AlpinistID] = [FO].[AlpinistID]
JOIN [FoodTypes] AS [FT]
    ON [FO].[FoodTypeID] = [FT].[FoodTypeID]
WHERE ([Date] BETWEEN '2017-01-23' AND '2017-09-24')
ORDER BY [DateStart]
```

Результат виконання запиту наведений на рисунку 4.1.1. та 4.1.2.

Results		Messages				
	FirstName	LastName	DateStart	DateEnd	Name	Price
1	Агнія	Умская	2017-09-10	NULL	Вечеря	150
2	Роза	Боярова	2017-09-10	NULL	Сніданок	100
3	Дар'я	Яковець	2017-09-11	NULL	Вечеря	150
4	Роза	Дудник	2017-09-11	NULL	Сніданок	100
5	Ростислав	Флеров	2017-09-11	2017-09-13	Будинок	900
6	Едуард	Цур-Мілен	2017-09-12	NULL	Обід	200
7	Захар	Горбунков	2017-09-12	NULL	Вечеря	150
8	Ізабелла	Машаріна	2017-09-12	NULL	Обід	200
9	Іраїда	Якушкова	2017-09-12	NULL	Вечеря	150
10	Влада	Власова	2017-09-13	NULL	Сніданок	100
11	Злата	Яцкевич	2017-09-13	NULL	Обід	200
12	Соломон	Смирнов	2017-09-13	NULL	Сніданок	100
13	Анастасія	Санькова	2017-09-14	NULL	Сніданок	100
14	Уляна	Дідова	2017-09-14	NULL	Обід	200
15	Володимир	Закруткін	2017-09-15	NULL	Вечеря	150
16	Зінаїда	Бруєвича	2017-09-15	NULL	Сніданок	100
17	Кирило	Лукашенко	2017-09-15	NULL	Вечеря	150
18	Марія	Курзибова	2017-09-15	NULL	Сніданок	100
19	Стела	Ярова	2017-09-15	NULL	Сніданок	100

Рисунок 4.1.1 – Результат виконання запиту про всі замовлення в SQL Server Management Studio (SSMS)

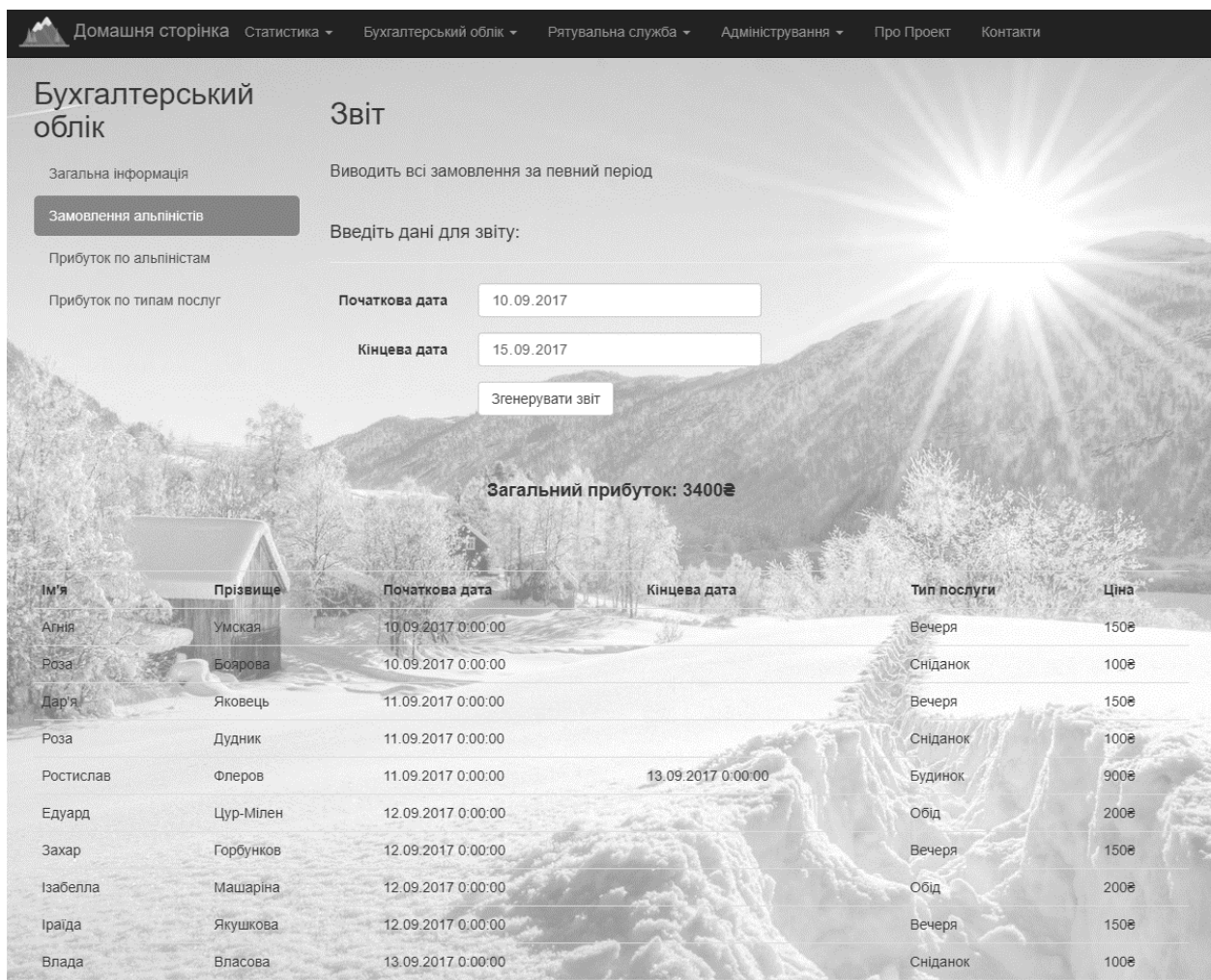


Рисунок 4.1.2 – Результат виконання запиту про всі замовлення в веб-додатку

4.2. Вибірка загального прибутку по типам послуг за певний період.

Запит:

```
SELECT
    [Name],
    SUM([Price]) AS [Profit]
FROM [Alpinists] AS [A]
JOIN
    (SELECT
        [A].[AlpinistID],
        [DateStart],
        [DateEnd],
        [HT].[Name],
        [Price] * (DateDiff(DD, [DateStart], [DateEnd]) + 1) AS [Price]
    FROM [HouseOrders] AS [HO]
    JOIN [Alpinists] AS [A]
        ON [A].[AlpinistID] = [HO].[AlpinistID]
    JOIN [Houses] AS [H]
        ON [HO].[HouseID] = [H].[HouseID]
    JOIN [HouseTypes] AS [HT]
        ON [H].[HouseTypeID] = [HT].[HouseTypeID])
```

```

WHERE ([DateStart] BETWEEN '2017-11-01' AND '2017-11-30') AND ([DateEnd] BETWEEN '2017-11-
01' AND '2017-11-30')
UNION
SELECT
    [A].[AlpinistID],
    [Date] AS [DateStart],
    NULL AS [DateEnd],
    [Name],
    [Price]
FROM [FoodOrders] AS [FO]
JOIN [Alpinists] AS [A]
    ON [A].[AlpinistID] = [FO].[AlpinistID]
JOIN [FoodTypes] AS [FT]
    ON [FO].[FoodTypeID] = [FT].[FoodTypeID]
WHERE ([Date] BETWEEN '2017-11-01' AND '2017-11-30')
) AS [P]
ON [A].[AlpinistID] = [P].[AlpinistID]
GROUP BY [Name]
ORDER BY [Profit] DESC

```

Результат виконання запиту наведений на рисунку 4.2.1. та 4.2.2.

The screenshot shows the SQL Server Management Studio interface. The top pane displays a SQL query. The bottom pane shows the results of the query in a table format.

	Name	Profit
1	Будинок	11700
2	Намет	5600
3	Обід	5200
4	Вечеря	5100
5	Сніданок	4200

Рисунок 4.2.1 – Результат виконання запиту в SQL Server Management Studio

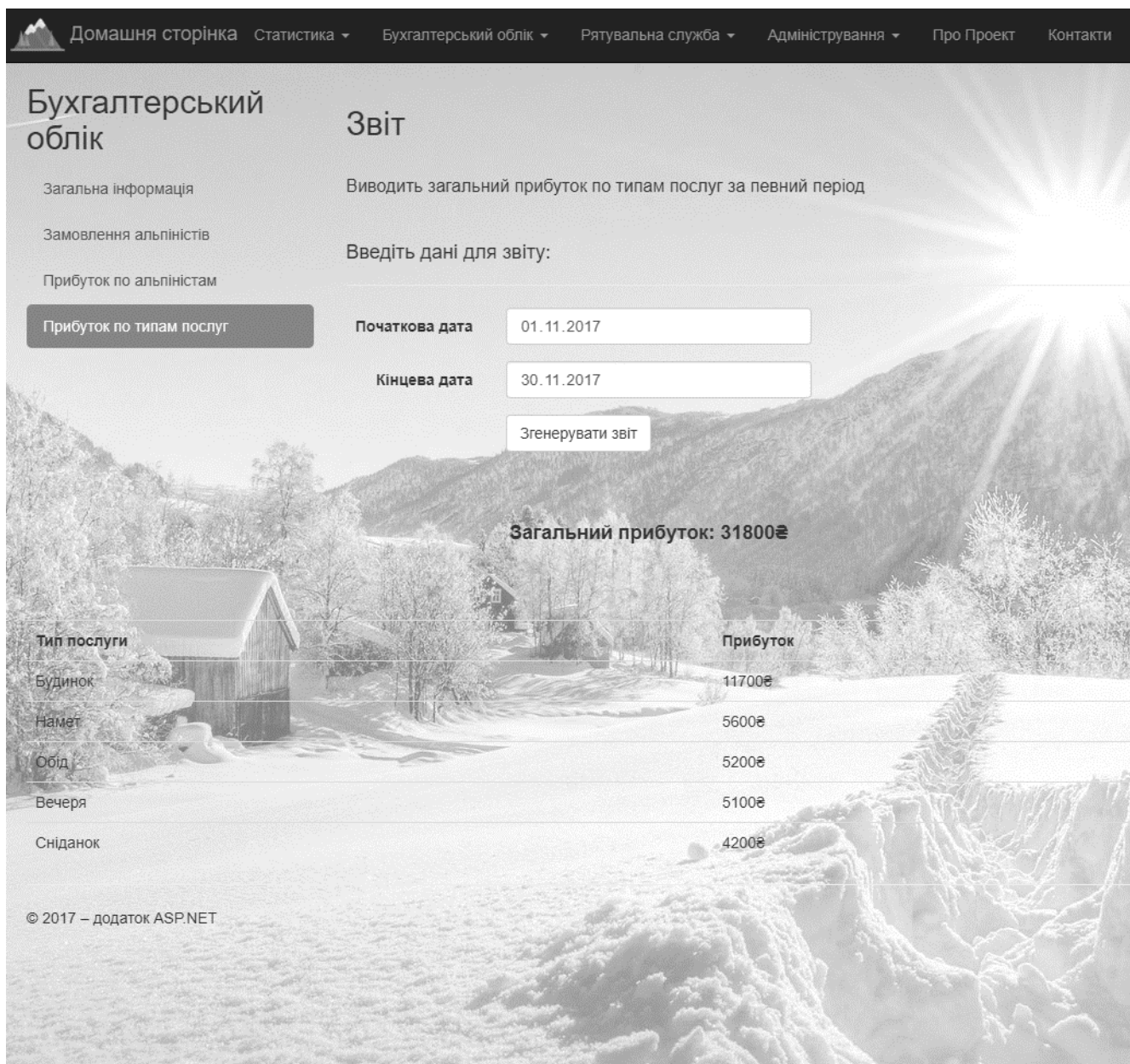


Рисунок 4.2.2 – Результат виконання запиту в веб-додатку

4.3. Вибірка списку маршрутів з їх поточним станом.

Запит:

```
SELECT
    [Name],
    [Country],
    [State]
FROM [Routes] AS [R]
JOIN [RouteStates] AS [RS]
    ON [R].[RouteID] = [RS].[RouteID]
JOIN [AlpinistBases] AS [AB]
    ON [R].[AlpinistBaseID] = [AB].[AlpinistBaseID]
WHERE ([DateEnd] IS NULL)
```

Результат виконання запиту наведений на рисунку 4.3.1. та 4.3.2.

Coursework queries...ENTINE\Kappi (54))* X VALENTINE\SQLEXP...rsework - D

```

293
294 SELECT
295     [Name],
296     [Country],
297     [State]
298 FROM [Routes] AS [R]
299 JOIN [RouteStates] AS [RS]
300     ON [R].[RouteID] = [RS].[RouteID]
301 JOIN [AlpinistBases] AS [AB]
302     ON [R].[AlpinistBaseID] = [AB].[AlpinistBaseID]
303 WHERE ([DateEnd] IS NULL)

```

100 %

Results Messages

	Name	Country	State
1	Безенгійська стіна	Грузія	Задовільний
2	Гора Шхара	Грузія	Хороший
3	Гора Ельбрус	Росія	Відмінний
4	Алтайські гори	Росія	Поганий
5	Гори Сибіру і далекого сходу	Росія	Задовільний
6	Гори Тянь-Шань	Киргизстан	Хороший
7	Гори Паміру	Киргизстан	Відмінний
8	Пік Хан-Тенгі	Киргизстан	Поганий
9	Головний Кавказький хребет	Грузія	Very

Рисунок 4.3.1 – Результат виконання запиту в SQL Server Management Studio

Домашня сторінка Статистика Бухгалтерський облік Рятувальна служба Адміністрування Про Проект Контакти

Рятувальна служба

Поточний стан маршрутів:

Маршрут	Країна	Стан маршруту
Алтайські гори	Росія	Поганий
Безенгійська стіна	Грузія	Задовільний
Головний Кавказький хребет	Грузія	Very
Гора Ельбрус	Росія	Відмінний
Гора Шхара	Грузія	Хороший
Гори Паміру	Киргизстан	Відмінний
Гори Сибіру і далекого сходу	Росія	Задовільний
Гори Тянь-Шань	Киргизстан	Хороший
Пік Хан-Тенгі	Киргизстан	Поганий

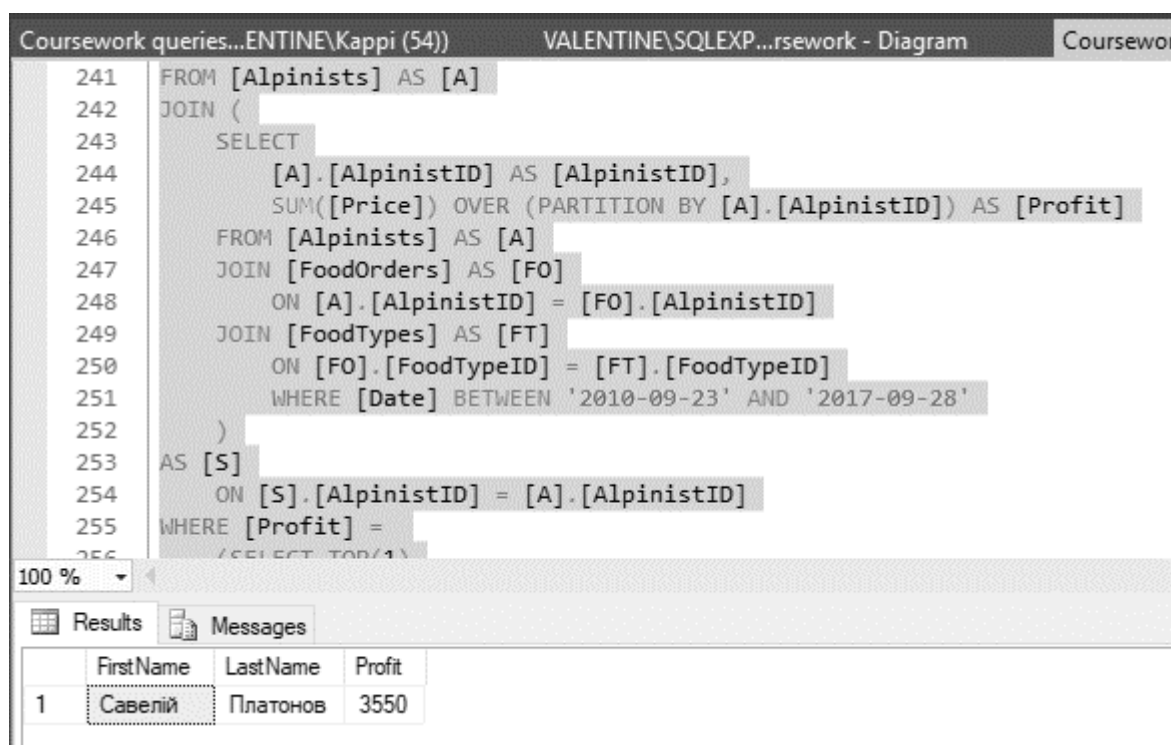
Рисунок 4.3.2 – Результат виконання запиту в веб-додатку

4.4. Вивести, який альпініст витратив більше всіх грошей на харчування з 2010-09-23 по 2017-09-28.

Запит:

```
SELECT DISTINCT
    [FirstName],
    [LastName],
    [Profit]
FROM [Alpinists] AS [A]
JOIN (
    SELECT
        [A].[AlpinistID] AS [AlpinistID],
        SUM([Price]) OVER (PARTITION BY [A].[AlpinistID]) AS [Profit]
    FROM [Alpinists] AS [A]
    JOIN [FoodOrders] AS [FO]
        ON [A].[AlpinistID] = [FO].[AlpinistID]
    JOIN [FoodTypes] AS [FT]
        ON [FO].[FoodTypeID] = [FT].[FoodTypeID]
    WHERE [Date] BETWEEN '2010-09-23' AND '2017-09-28'
)
AS [S]
ON [S].[AlpinistID] = [A].[AlpinistID]
WHERE [Profit] =
    (SELECT TOP(1)
        SUM([Price]) OVER (PARTITION BY [A].[AlpinistID]) AS [Profit]
    FROM [Alpinists] AS [A]
    JOIN [FoodOrders] AS [FO]
        ON [A].[AlpinistID] = [FO].[AlpinistID]
    JOIN [FoodTypes] AS [FT]
        ON [FO].[FoodTypeID] = [FT].[FoodTypeID]
    WHERE [Date] BETWEEN '2010-09-23' AND '2017-09-28'
    ORDER BY [Profit] DESC)
```

Результат виконання запиту наведений на рисунку 4.4.



The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor displays the SQL code from the previous block, with line numbers 241 through 256. The results pane shows a single row of data.

FirstName	LastName	Profit
Савелій	Платонов	3550

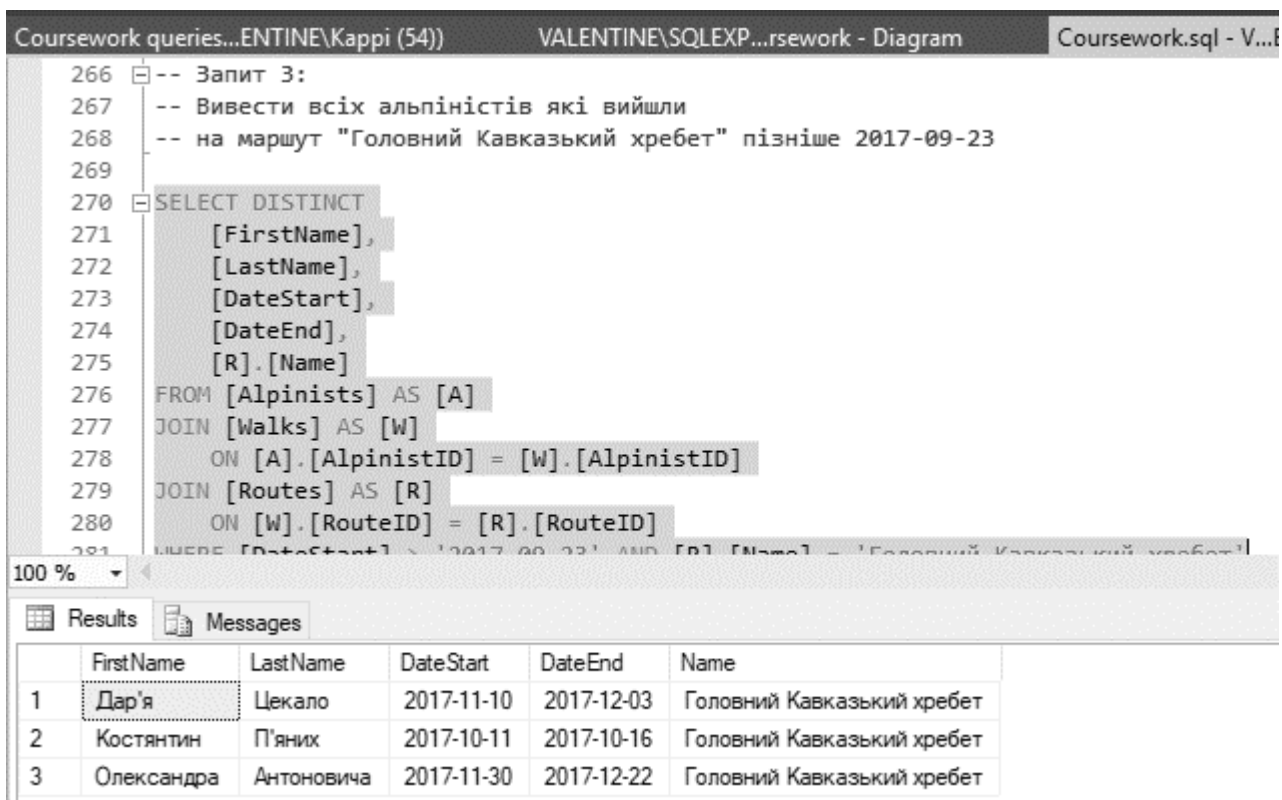
Рисунок 4.4 – Результат виконання запиту

4.5. Вивести всіх альпіністів які вийшли на маршрут "Головний Кавказький хребет" пізніше 2017-09-23.

Запит:

```
SELECT DISTINCT
    [FirstName],
    [LastName],
    [DateStart],
    [DateEnd],
    [R].[Name]
FROM [Alpinists] AS [A]
JOIN [Walks] AS [W]
    ON [A].[AlpinistID] = [W].[AlpinistID]
JOIN [Routes] AS [R]
    ON [W].[RouteID] = [R].[RouteID]
WHERE [DateStart] > '2017-09-23' AND [R].[Name] = 'Головний Кавказький хребет'
```

Результат виконання запиту наведений на рисунку 4.5.



Coursework queries...ENTINE\Kappi (54) VALENTINE\SQLXP...rsework - Diagram Coursework.sql - V...

266 -- Запит 3:
267 -- Вивести всіх альпіністів які вийшли
268 -- на маршрут "Головний Кавказький хребет" пізніше 2017-09-23
269
270 SELECT DISTINCT
271 [FirstName],
272 [LastName],
273 [DateStart],
274 [DateEnd],
275 [R].[Name]
276 FROM [Alpinists] AS [A]
277 JOIN [Walks] AS [W]
278 ON [A].[AlpinistID] = [W].[AlpinistID]
279 JOIN [Routes] AS [R]
280 ON [W].[RouteID] = [R].[RouteID]
281 WHERE [DateStart] > '2017-09-23' AND [R].[Name] = 'Головний Кавказький хребет'

100 %

Results Messages

	FirstName	LastName	DateStart	DateEnd	Name
1	Дар'я	Цекало	2017-11-10	2017-12-03	Головний Кавказький хребет
2	Костянтин	П'яних	2017-10-11	2017-10-16	Головний Кавказький хребет
3	Олександра	Антоновича	2017-11-30	2017-12-22	Головний Кавказький хребет

Рисунок 4.5 – Результат виконання запиту

4.6. Визначити найпопулярніший маршрут(найбільша кількість альпіністів).

Запит:

```
SELECT
    [Name],
```

```

[AlpinistsCount]
FROM [Routes] AS [R]
JOIN (
    SELECT DISTINCT
        [R].[RouteID] AS [RouteID],
        COUNT([A].[AlpinistID]) OVER (PARTITION BY [R].[RouteID]) AS [AlpinistsCount]
    FROM [Alpinists] AS [A]
    JOIN [Walks] AS [W]
        ON [A].[AlpinistID] = [W].[AlpinistID]
    JOIN [Routes] AS [R]
        ON [W].[RouteID] = [R].[RouteID]
    AS [AC]
    ON [AC].[RouteID] = [R].[RouteID]
WHERE [AlpinistsCount] = (
    SELECT TOP(1)
        COUNT([A].[AlpinistID]) OVER (PARTITION BY [R].[RouteID]) AS [AlpinistsCount]
    FROM [Alpinists] AS [A]
    JOIN [Walks] AS [W]
        ON [A].[AlpinistID] = [W].[AlpinistID]
    JOIN [Routes] AS [R]
        ON [W].[RouteID] = [R].[RouteID]
    ORDER BY [AlpinistsCount] DESC)

```

Результат виконання запиту наведений на рисунку 4.6.

The screenshot shows a SQL query window with the following text:

```

281 WHERE [DateStart] > '2017-09-23' AND [R].[Name] = 'Головний Кавказький хребет'
282
283 -- Запит 4:
284 -- Визначити найпопулярніший маршрут(найбільша кількість альпіністів)
285
286 SELECT
287     [Name],
288     [AlpinistsCount]
289 FROM [Routes] AS [R]
290 JOIN (
291     SELECT DISTINCT
292         [R].[RouteID] AS [RouteID],
293         COUNT([A].[AlpinistID]) OVER (PARTITION BY [R].[RouteID]) AS [AlpinistsCount]
294     FROM [Alpinists] AS [A]
295     JOIN [Walks] AS [W]
296         ON [A].[AlpinistID] = [W].[AlpinistID]
297     JOIN [Routes] AS [R]
298         ON [W].[RouteID] = [R].[RouteID]
299     ORDER BY [AlpinistsCount] DESC
300 ) AS [AC]
301 ON [AC].[RouteID] = [R].[RouteID]

```

Below the query, the 'Results' tab is active, showing a table with two columns: 'Name' and 'AlpinistsCount'. The table contains one row with the value 'Гори Паміру' and '235'.

	Name	AlpinistsCount
1	Гори Паміру	235

Рисунок 4.6 – Результат виконання запиту

4.7. Вивести для кожного виду житла кількість альпіністів які там перебувають 2017-09-23.

Запит:

```

SELECT DISTINCT
    [HT].[Name],

```

```

COUNT([A].[AlpinistID]) OVER (PARTITION BY [HT].[HouseTypeID]) AS [AlpinistsCount]
FROM [Alpinists] AS [A]
JOIN [HouseOrders] AS [HO]
    ON [A].[AlpinistID] = [HO].[AlpinistID]
JOIN [Houses] AS [H]
    ON [HO].[HouseID] = [H].[HouseID]
JOIN [HouseTypes] AS [HT]
    ON [H].[HouseTypeID] = [HT].[HouseTypeID]
WHERE '2017-09-23' BETWEEN [DateStart] AND [DateEnd]

```

Результат виконання запиту наведений на рисунку 4.7.

The screenshot shows a SQL query window with the following text:

```

-- Вивести для кожного виду житла кількість альпіністів які там перебувають 2017-09-23
SELECT DISTINCT
    [HT].[Name],
    COUNT([A].[AlpinistID]) OVER (PARTITION BY [HT].[HouseTypeID]) AS [AlpinistsCount]
FROM [Alpinists] AS [A]
JOIN [HouseOrders] AS [HO]
    ON [A].[AlpinistID] = [HO].[AlpinistID]
JOIN [Houses] AS [H]
    ON [HO].[HouseID] = [H].[HouseID]
JOIN [HouseTypes] AS [HT]
    ON [H].[HouseTypeID] = [HT].[HouseTypeID]
WHERE '2017-09-23' BETWEEN [DateStart] AND [DateEnd]

```

Below the query window, the 'Results' pane displays the following data:

	Name	AlpinistsCount
1	Будинок	2
2	Намет	3

Рисунок 4.7 – Результат виконання запиту

5 ІНСТРУКЦІЯ КОРИСТУВАЧА

1. Завантаження головної сторінки веб-додатку (рисунк 5.1).

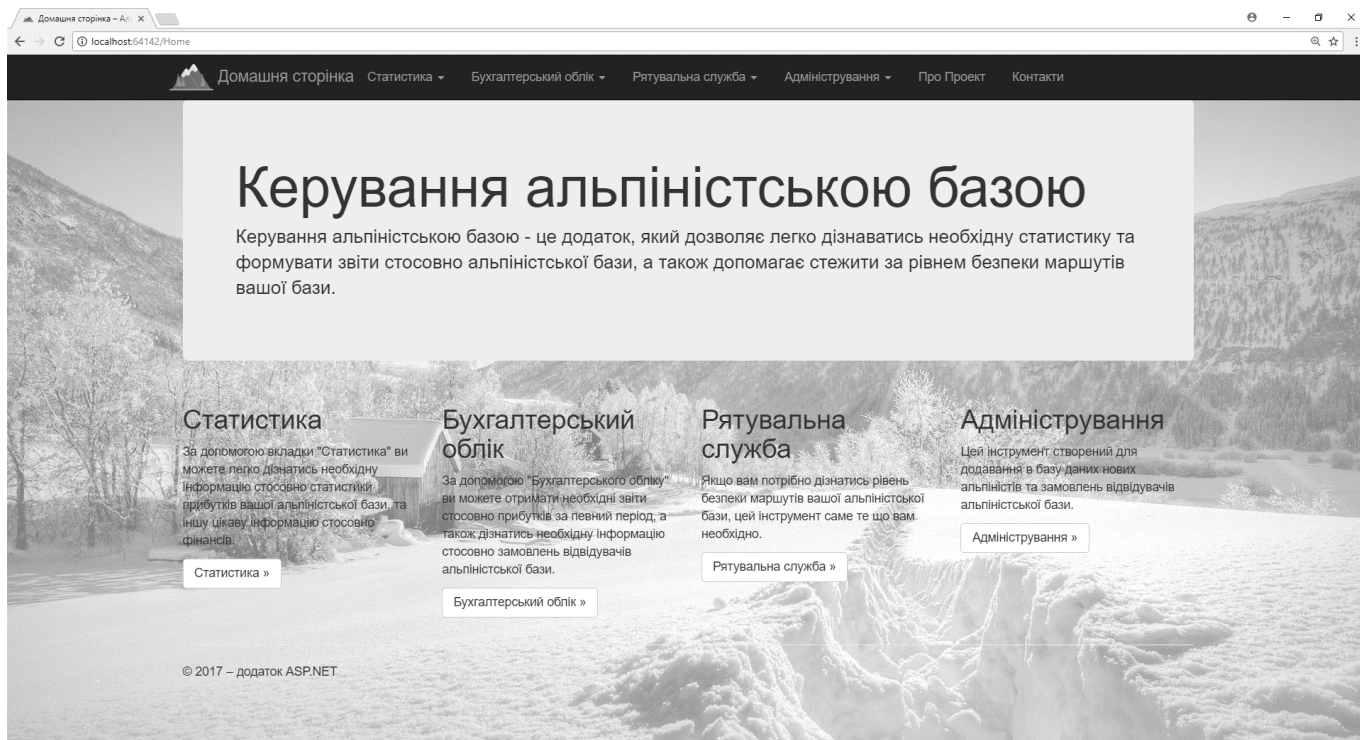


Рисунок 5.1 – Головна сторінка веб-додатку після першого завантаження

2. Перегляд статистики (загальної чи по проживанню чи харчуванню) (рисунк 5.2).

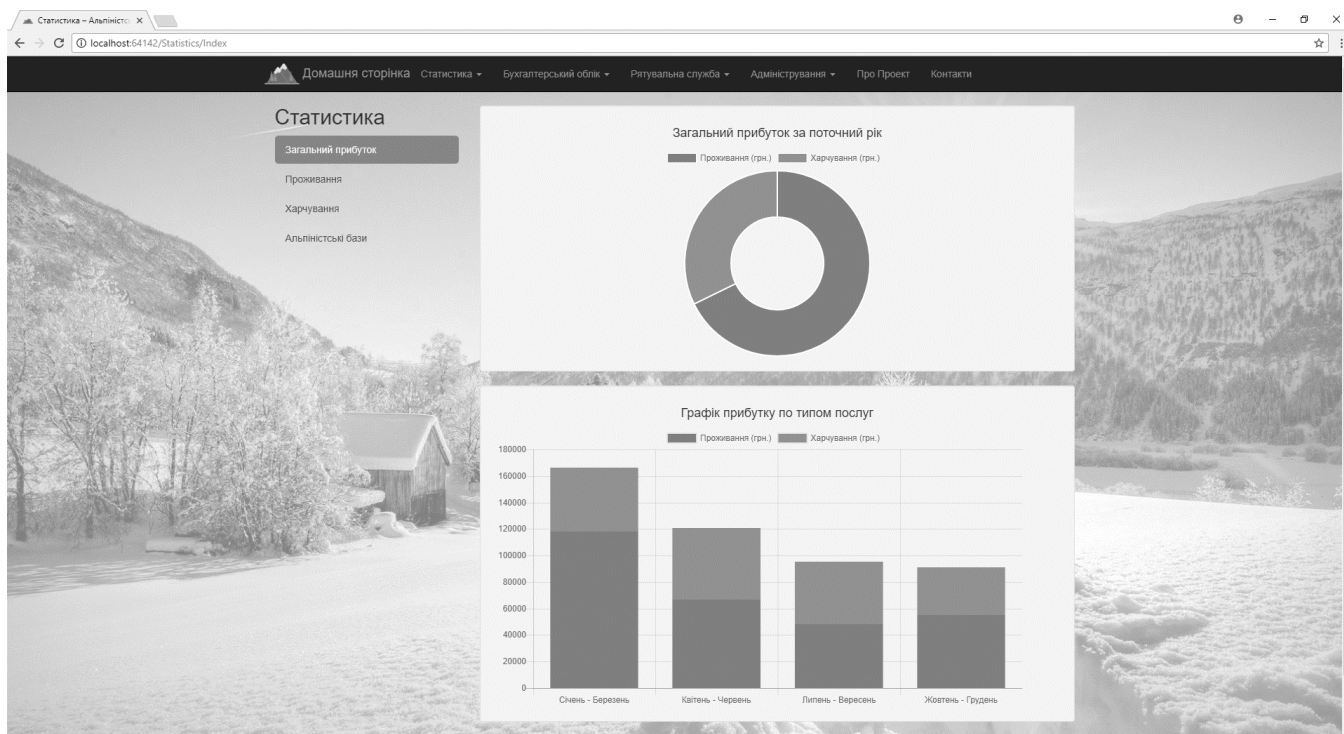


Рисунок 5.2 – Сторінка загальної статистики

3. Створення нового звіту за певний період(Бухгалтерський облік) (рисунк 5.3).

Звіт замовлення альпіністів

Домашня сторінка | Статистика | Бухгалтерський облік | Рятувальна служба | Адміністрування | Про Проект | Контакти

Бухгалтерський облік

Загальна інформація

Замовлення альпіністів

Прибуток по альпіністам

Прибуток по типам послуг

Звіт

Виводить всі замовлення за певний період

Введіть дані для звіту:

Початкова дата: 01.12.2017

Кінцева дата: 02.12.2017

Згенерувати звіт

Загальний прибуток: 1050€

Ім'я	Прізвище	Початкова дата	Кінцева дата	Тип послуги	Ціна
Аристарх	Парин	01.12.2017 0:00:00		Сніданок	100€
Артур	Ільїн	01.12.2017 0:00:00		Вечера	150€
Данило	Перевертов	01.12.2017 0:00:00		Вечера	150€
Панкратій	Смишляев	01.12.2017 0:00:00		Сніданок	100€
Фелікс	Часом	01.12.2017 0:00:00		Обід	200€
Агнія	Умская	02.12.2017 0:00:00		Обід	200€
Ксенія	Лісничка	02.12.2017 0:00:00		Вечера	150€

© 2017 – додаток ASP.NET

Рисунок 5.3 – Створення нового звіту з заданим періодом

4. Перегляд та зміна поточного стану маршрутів (рисунк 5.4).

Рятувальна служба

Домашня сторінка | Статистика | Бухгалтерський облік | Рятувальна служба | Адміністрування | Про Проект | Контакти

Рятувальна служба

Поточний стан маршрутів:

Маршрут	Країна	Стан маршруту
Алтайські гори	Росія	Поганий
Безенійська стіна	Грузія	Задовільний
Головний Кавказький хребет	Грузія	Великий
Гора Ельбрус	Росія	Відмінний
Гора Шхара	Грузія	Хороший
Гори Паміру	Киргизстан	Відмінний
Гори Сибіру і далекого сходу	Росія	Задовільний
Гори Тянь-Шань	Киргизстан	Хороший
Пік Хан-Тенгі	Киргизстан	Поганий

Зміна стану маршруту:

Маршрут: Алтайські гори

Стан: Жахливий

Змінити стан маршруту

© 2017 – додаток ASP.NET

Рисунок 5.4 – Перегляд поточного стану маршрутів

5. Реєстрація нового альпініста (опціонально) (рисунк 5.5).

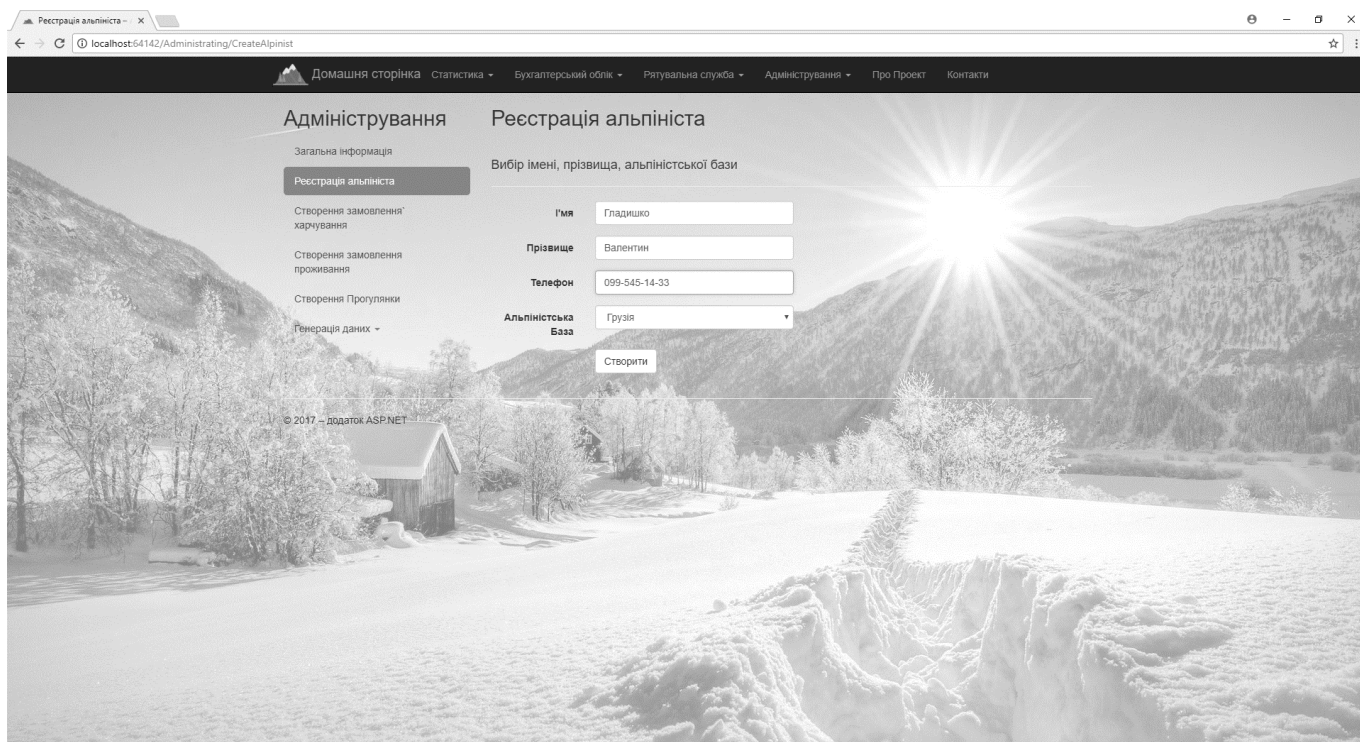


Рисунок 5.5 – Реєстрація нового альпініста

ВИСНОВКИ

В процесі виконання курсової роботи була створена платформа для бухгалтерського обліку, адміністрування та керування альпіністською базою. Зараз ця система складається з веб-сервера та веб-додатку, що розташовані на одній машині, проте сервер може мати будь-яку кількість клієнтів, тому систему в подальшому можна значно розширити.

Для виконання роботи були отримані теоретичні знання та практичні навички з побудови клієнт-серверної архітектури. Отримані навички створення сервера та бази даних на ньому, роботи з базою даних за допомогою різноманітних програмних засобів, прикладного застосування баз даних.

Також були значно поглиблені знання зі створення веб-додатків на платформі ASP.NET MVC, що мають велику кількість функціоналу, а також налагодження взаємодії між веб-додатком та сервером.

Створений веб-додаток, що вирішує проблему з адмініструванням та керуванням альпіністською базою.

ПЕРЕЛІК ПОСИЛАНЬ

1. Простір імен Microsoft.SqlServer.Server .NET Framework [Електронний ресурс] [https://msdn.microsoft.com/ru-ru/library/microsoft.sqlserver.server\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/microsoft.sqlserver.server(v=vs.110).aspx)
2. Простір імен System.Web.Mvc [Електронний ресурс] [https://msdn.microsoft.com/ru-ru/library/system.web.mvc\(v=vs.118\).aspx](https://msdn.microsoft.com/ru-ru/library/system.web.mvc(v=vs.118).aspx)
3. Бібліотека CSS стилів Bootstrap 3 [Електронний ресурс] <https://www.w3schools.com/bootstrap/>
4. Бібліотека для створення діаграм, гістограм та графіків для HTML5 Chart.js [Електронний ресурс] <http://www.chartjs.org/>
5. HTML теги [Електронний ресурс] <http://htmlbook.ru/html>
6. Довідник по Transact-SQL [Електронний ресурс] <https://docs.microsoft.com/ru-ru/sql/t-sql/language-reference>
7. Репозиторій проекту (GitHub) [Електронний ресурс] <https://github.com/ValentineGladysenko/Database-Coursework>