Database Management Systems LAB

(3+1 Credit Hours)

CSL-220

--Parking Management System--
Project Report

Couse Instructor: **Mr. Malik M. Ali**
Lab Instructor: **Ms. Hafsa Munawar**
Semester: **Fall23**
Class: BSCS **4B**

Group Members:

Enrollment	Name
02-134221-068	Emad Tariq
02-134221-090	Sofia Haider
02-134221-82	Ahsan Naeem

DEPARTMENT OF COMPUTER SCIENCE

BAHRIA UNIVERSITY, KARACHI, PAKISTAN

Table of Content:

Analysis 3
Methodology: 5
Implementation: 7
Conclusion: 32

Analysis

Objective:

The objective of a Parking Management System in a database is to efficiently manage parking spaces, facilitate user registration and vehicle information storage, enable parking reservations, track entry and exit of vehicles, implement billing and payment systems, provide real-time space availability information, ensure security and access control, integrate with parking equipment, offer reporting and analytics tools, comply with regulations, and create a user-friendly interface. The system aims to optimize parking facility usage, enhance user experience, and provide administrators with tools for efficient management and analysis of parking-related data.

Summary:

The Parking Management System is a comprehensive and user-friendly solution designed to streamline the process of parking space reservation and management through an online platform. This system replaces conventional manual methods with an automated approach, significantly improving the efficiency and precision of parking operations. For users, the system offers the ability to check current parking space availability, reserve a parking spot, and make secure payments. On the administrative side, parking facility staff can effortlessly update the parking space database, incorporating new additions, modifications, and removals. They can also manage parking schedules, reservation, available users. The system ensures the accuracy of parking data and provides real-time updates to both users and administrators, contributing to an enhanced parking experience and more effective operational control.

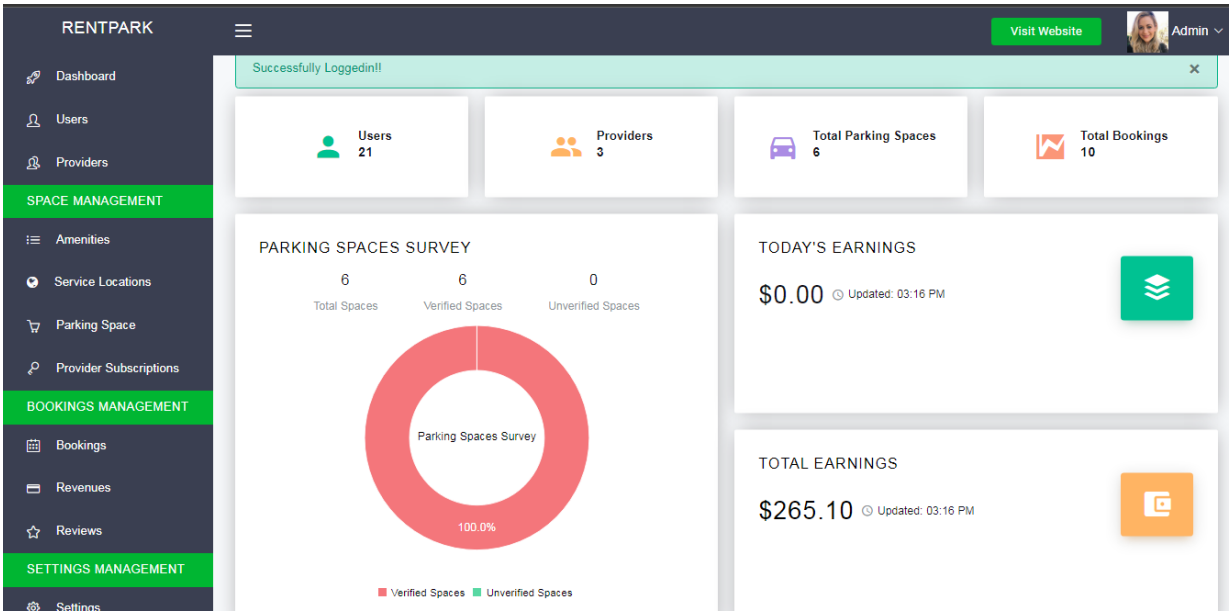
Functionalities:

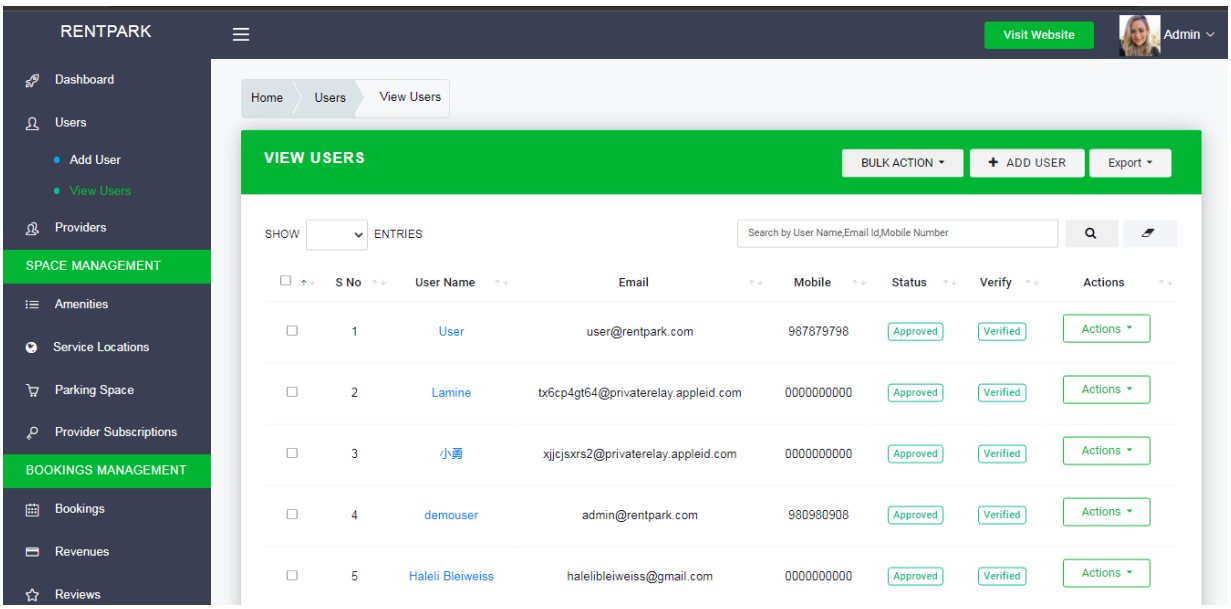
1. **Availability Tracking:**
    - Allows users to check the availability of parking spaces in different lots.
2. **Parking History:**
    - Provides a history of entry and exit transactions for a specific vehicle.
3. **Revenue Generation:**
    - Calculates and reports the total revenue generated by the parking system over a specified time period.

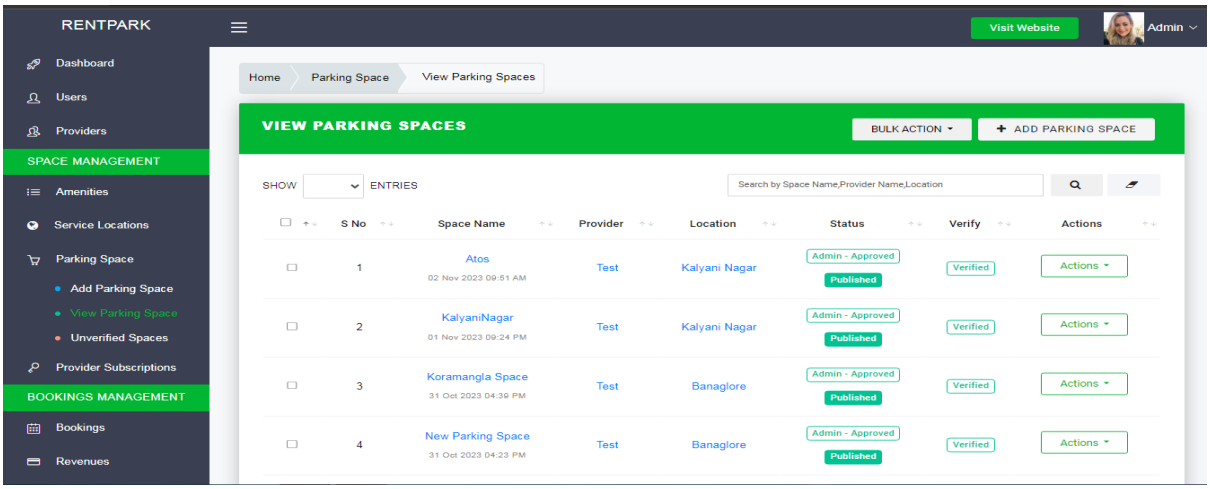
Scope**:**

The Parking Management System aims to efficiently manage parking facilities through a SQL-based relational database. It covers the administration of parking lots, spaces, vehicles, and transactions, offering features such as real-time space availability, accurate transaction recording, and security measures. to revolutionize the conventional approach to parking by offering a seamless and efficient online solution. This comprehensive platform prioritizes user-friendliness, providing an intuitive interface for both customers and administrators. Users can easily navigate through the system to check parking space availability, select preferred slots, and securely complete transactions. On the administrative side, staff members have robust tools to manage the parking space database, allowing for additions, modifications, and removals as needed. Real-time updates ensure that users and administrators stay informed about parking space status and any alterations to the database.

Reference: Rent Park







Methodology:

- **Business Rules**

. VehicleOwner has a vehicle , multiple vehicle can be under owner (one to many relationship)

. One user having reservation can have one parking space in parking lot (one to one relationship)

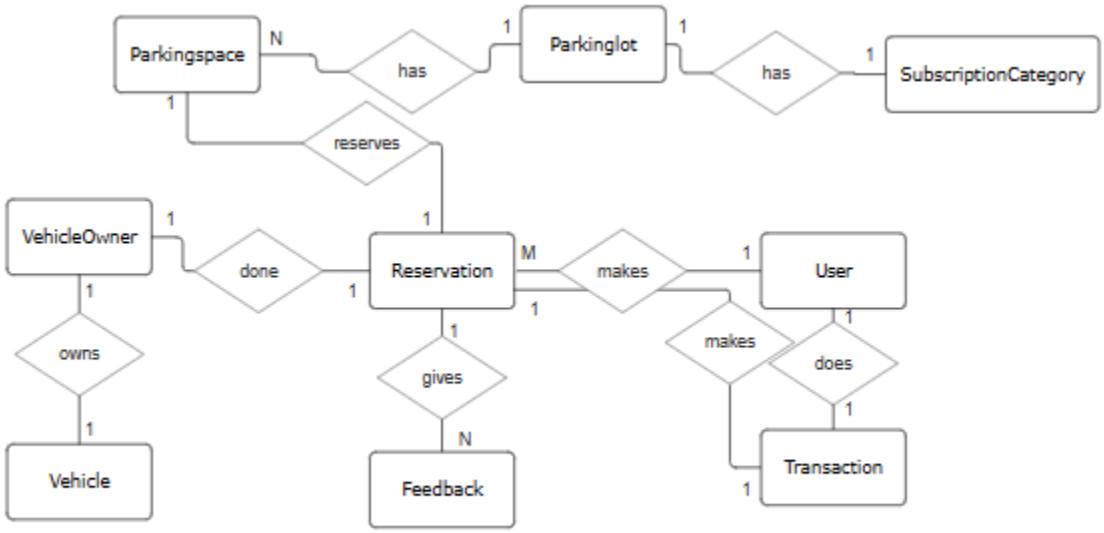
. A user having revervation gives atleast one feedback (one to many relationship)

. One transaction can be made from one user under one reservation (one to one relation)

. A subscriptionCategory has one parking lot and one parkinglot belongs to one category (one to one relationship)

. One parkinglot having multiple parkingspaces (one to many relationship)

Entity Relationship Diagram



Attributes:

Parkingspace -> SpaceID

Parkinglot -> LotID,Capacity,LotName

Subcriptioncategory ->SubID,SubCategory,HourlyRate

VehicleOwner ->UserID

Vehicle ->VehicleID,LiscensePlate

Reservation ->ReservationID,VehiclePlate,ParkDate,ParkTime

User ->UserID,Username,FirstName,LastName,Email,Password,Role

Transaction ->TransactionID,ExitTime,TotalAmount

Implementation:

Conceptual to Logical Mapping

(PARKINGSPACE)

-SpaceID (primarykey, int)

-LotID(foreignkey, int)

(PARKINGLOT)

-SpaceID (foreignkey , int)

-LotID(primarykey, int)

-Capacity

-LocName

(TRANSACTION)

-TransactionID (Primary,int)

-UserID (foreignkey,int)

-ReservationID (foreignkey,int)

-ExitTime (time(7))

-TotalAmount (decimal(10, 2))

(FEEDBACK)

-FeedbackID (primarykey,int)

-ReservationID (foreignkey,int)

-Date (datetime)

-Comment (varchar(50))

-Rating (int)

(USER)

-UserID (primarykey,int)

-Username (varchar(50))

-FirstName (varchar(50))

-LastName (varchar(50))

-Email (varchar(25))

-Password (varchar(25))

-Role (varchar(20))

(RESERVATION)

-ReservationID (primarykey,int)

-UserID (foreignkey,,int)

-VehiclePlate (varchar(25))

-LotID (foreignkey,int)

-SpaceID (foreignkey,int)

-ParkDate (date)

-ParkTime (time(7))

(VEHICLE)

-vehicleID(int)

-LicensePlate(int)

(VEHICLEOWNER)

-VehicleID (Primarykey, int)

-LicensePlate (varchar(20))

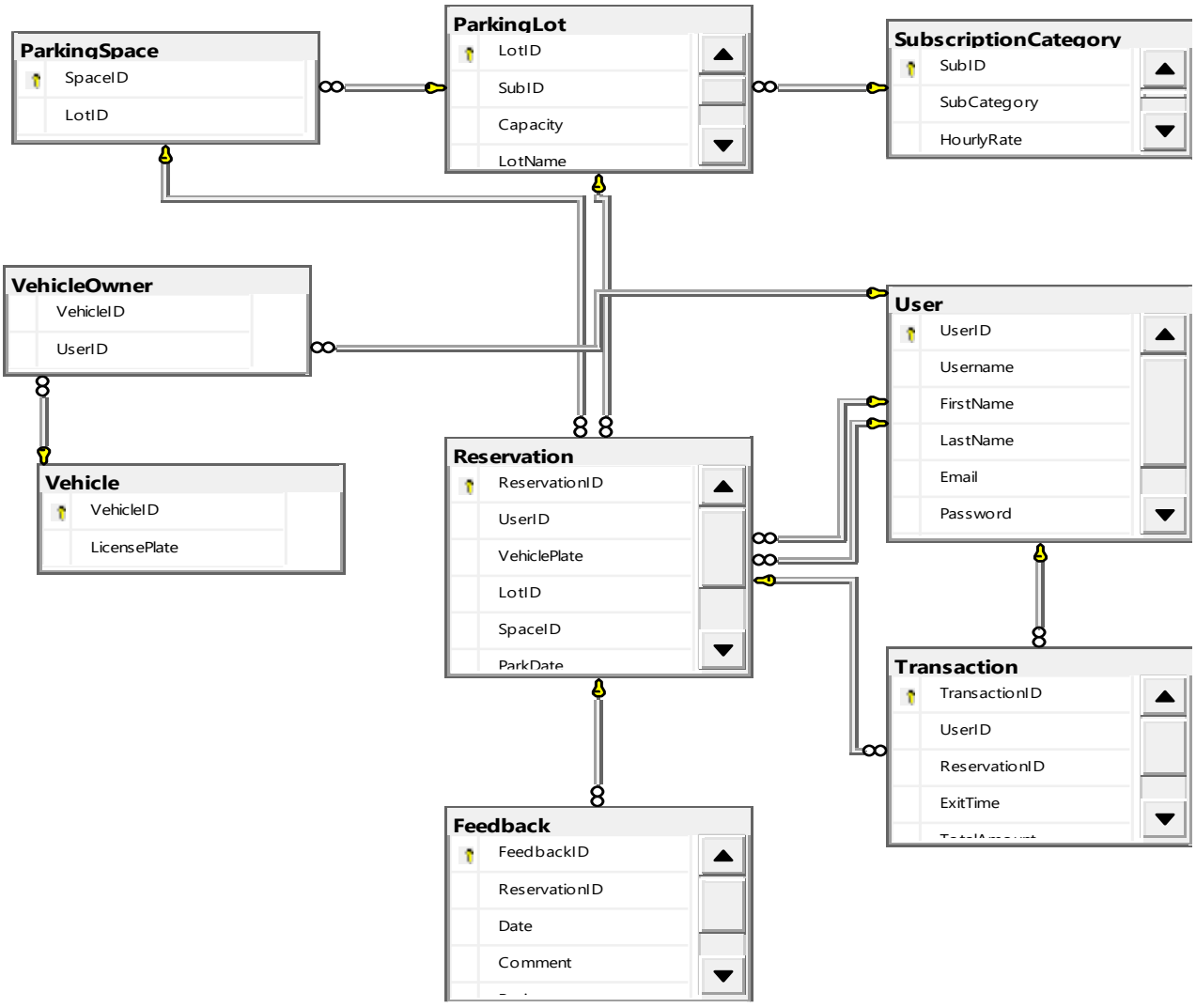
(SUBSCRIPTIONCATEGORY)

-SubID (primarykey,int)

-SubCategory (varchar(25))

-HourlyRate (decimal(10, 2))

SQL Server Schema Diagram



DDL:

```
CREATE TABLE [dbo].[Feedback](
	[FeedbackID] [int] IDENTITY(1,1) NOT NULL,
	[ReservationID] [int] NULL,
	[Date] [datetime] NULL,
	[Comment] [varchar](50) NULL,
	[Rating] [int] NULL,
PRIMARY KEY CLUSTERED
(
	[FeedbackID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[User](
	[UserID] [int] IDENTITY(1,1) NOT NULL,
	[Username] [varchar](50) NULL,
	[FirstName] [varchar](50) NULL,
	[LastName] [varchar](50) NULL,
	[Email] [varchar](25) NULL,
	[Password] [varchar](25) NULL,
	[Role] [varchar](20) NULL,
PRIMARY KEY CLUSTERED
(
	[UserID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[Reservation](
	[ReservationID] [int] IDENTITY(1,1) NOT NULL,
	[UserID] [int] NULL,
	[VehiclePlate] [varchar](25) NULL,
	[LotID] [int] NULL,
	[SpaceID] [int] NULL,
	[ParkDate] [date] NULL,
	[ParkTime] [time](7) NULL,
PRIMARY KEY CLUSTERED
(
	[ReservationID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[SubscriptionCategory](
	[SubID] [int] IDENTITY(1,1) NOT NULL,
	[SubCategory] [varchar](25) NULL,
	[HourlyRate] [decimal](10, 2) NULL,
PRIMARY KEY CLUSTERED
(
	[SubID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[ParkingLot](
	[LotID] [int] IDENTITY(1,1) NOT NULL,
	[SubID] [int] NULL,
	[Capacity] [int] NULL,
	[LotName] [varchar](25) NULL,
PRIMARY KEY CLUSTERED
(
	[LotID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[ParkingSpace](
	[SpaceID] [int] NOT NULL,
	[LotID] [int] NULL,
PRIMARY KEY CLUSTERED
(
	[SpaceID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[Transaction](
	[TransactionID] [int] IDENTITY(1,1) NOT NULL,
	[UserID] [int] NULL,
	[ReservationID] [int] NULL,
	[ExitTime] [time](7) NULL,
	[TotalAmount] [decimal](10, 2) NULL,
PRIMARY KEY CLUSTERED
(
	[TransactionID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[Vehicle](
	[VehicleID] [int] IDENTITY(1,1) NOT NULL,
	[LicensePlate] [varchar](20) NULL,
PRIMARY KEY CLUSTERED
(
	[VehicleID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[VehicleOwner](
    [VehicleID] [int] NULL,
    [UserID] [int] NULL
) ON [PRIMARY]
GO
SET IDENTITY_INSERT [dbo].[Feedback] ON
```

DML:

```
SET IDENTITY_INSERT [dbo].[Feedback] ON

INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment], [Rating]) VALUES (1, 1, CAST(N'2024-01-10T09:30:00.000' AS DateTime), N'Good service!', 4)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment], [Rating]) VALUES (2, 2, CAST(N'2024-01-11T11:45:00.000' AS DateTime), N'Satisfactory.', 3)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment], [Rating]) VALUES (3, 3, CAST(N'2024-01-12T13:15:00.000' AS DateTime), N'Excellent experience!', 5)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment], [Rating]) VALUES (4, 4, CAST(N'2024-01-13T15:30:00.000' AS DateTime), N'Not happy with the service.', 1)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment], [Rating]) VALUES (5, 5, CAST(N'2024-01-14T17:45:00.000' AS DateTime), N'Highly recommended!', 5)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment], [Rating]) VALUES (6, 5, CAST(N'2024-01-05T21:02:01.567' AS DateTime), N'Excelent Work', 5)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment], [Rating]) VALUES (7, 23, CAST(N'2024-01-08T00:11:17.827' AS DateTime), N'Need Improvment', 2)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment], [Rating]) VALUES (8, 1, CAST(N'2024-01-08T01:15:54.223' AS DateTime), N'Good', 4)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment],
```

```
[Rating]) VALUES (9, 15, CAST(N'2024-01-08T01:17:38.717' AS DateTime),
N'Good', 3)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment],
[Rating]) VALUES (1008, 1024, CAST(N'2024-01-08T15:42:15.263' AS DateTime),
N'Nice!', 3)
INSERT [dbo].[Feedback] ([FeedbackID], [ReservationID], [Date], [Comment],
[Rating]) VALUES (1009, 1025, CAST(N'2024-01-08T15:48:41.233' AS DateTime),
N'Great service!', 4)
SET IDENTITY_INSERT [dbo].[Feedback] OFF
GO
SET IDENTITY_INSERT [dbo].[ParkingLot] ON

INSERT [dbo].[ParkingLot] ([LotID], [SubID], [Capacity], [LotName]) VALUES (1, 1,
20, N'A')
INSERT [dbo].[ParkingLot] ([LotID], [SubID], [Capacity], [LotName]) VALUES (2, 2,
30, N'B')
INSERT [dbo].[ParkingLot] ([LotID], [SubID], [Capacity], [LotName]) VALUES (3, 3,
40, N'C')
SET IDENTITY_INSERT [dbo].[ParkingLot] OFF
GO
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (1, 1)
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (2, 2)
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (3, 3)
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (4, 1)
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (5, 1)
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (6, 1)
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (7, 1)
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (8, 1)
```

```
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (8, 1)
INSERT [dbo].[ParkingSpace] ([SpaceID], [LotID]) VALUES (9, 1)
GO
SET IDENTITY_INSERT [dbo].[Reservation] ON

INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (1, 3, N'LKB-999', 1, 1, CAST(N'2024-
01-10' AS Date), CAST(N'09:30:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (2, 4, N'HLB-239', 2, 2, CAST(N'2024-
01-11' AS Date), CAST(N'11:45:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (3, 5, N'JLM-529', 3, 3, CAST(N'2024-
01-12' AS Date), CAST(N'13:15:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (4, 6, N'ABC-897', 3, 3, CAST(N'2024-
01-13' AS Date), CAST(N'15:30:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (5, 7, N'DEI-358', 3, 3, CAST(N'2024-
01-14' AS Date), CAST(N'17:45:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (9, 18, N'XYZ-238', 2, 3,
CAST(N'1900-01-01' AS Date), CAST(N'00:00:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (13, 1, N'ABC-500', 1, 1,
CAST(N'2024-01-07' AS Date), CAST(N'22:00:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (15, 3, N'TMN-123', 3, 4
```

```
CAST(N'2024-01-07' AS Date), CAST(N'22:00:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (15, 3, N'TMN-123', 3, 4,
CAST(N'2024-01-07' AS Date), CAST(N'22:53:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (17, 3, N'BBB-111', 3, 4,
CAST(N'2024-01-07' AS Date), CAST(N'23:44:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (21, 2, N'FFF-500', 2, 3, CAST(N'2024
01-07' AS Date), CAST(N'23:52:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (23, 18, N'STO-621', 1, 7,
CAST(N'2024-01-07' AS Date), CAST(N'23:58:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (24, 3, N'QRP-548', 1, 8,
CAST(N'2024-01-08' AS Date), CAST(N'01:15:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (1024, 19, N'JOB-382', 3, 4,
CAST(N'2024-01-08' AS Date), CAST(N'15:41:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (1025, 21, N'MXY-222', 1, 9,
CAST(N'2024-01-08' AS Date), CAST(N'15:47:00' AS Time))
INSERT [dbo].[Reservation] ([ReservationID], [UserID], [VehiclePlate], [LotID],
[SpaceID], [ParkDate], [ParkTime]) VALUES (1026, 3, N'QER-234', 2, 3,
CAST(N'2024-01-08' AS Date), CAST(N'16:00:00' AS Time))
SET IDENTITY_INSERT [dbo].[Reservation] OFF
GO
SET IDENTITY_INSERT [dbo].[SubscriptionCategory] ON
```

```
INSERT [dbo].[SubscriptionCategory] ([SubID], [SubCategory], [HourlyRate])
VALUES (1, N'Gold', CAST(25.00 AS Decimal(10, 2)))
INSERT [dbo].[SubscriptionCategory] ([SubID], [SubCategory], [HourlyRate])
VALUES (2, N'Silver', CAST(15.00 AS Decimal(10, 2)))
INSERT [dbo].[SubscriptionCategory] ([SubID], [SubCategory], [HourlyRate])
VALUES (3, N'Platinum', CAST(35.00 AS Decimal(10, 2)))
SET IDENTITY_INSERT [dbo].[SubscriptionCategory] OFF
GO
SET IDENTITY_INSERT [dbo].[Transaction] ON

INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (2, 3, 1, CAST(N'11:45:00' AS Time), CAST(12.50 AS
Decimal(10, 2)))
INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (3, 4, 2, CAST(N'13:15:00' AS Time), CAST(18.75 AS
Decimal(10, 2)))
INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (4, 5, 3, CAST(N'15:30:00' AS Time), CAST(15.00 AS
Decimal(10, 2)))
INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (5, 6, 4, CAST(N'17:45:00' AS Time), CAST(10.50 AS
Decimal(10, 2)))
INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (6, 7, 5, CAST(N'19:00:00' AS Time), CAST(25.00 AS
Decimal(10, 2)))
INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (7, 18, 23, CAST(N'01:07:42.8966667' AS Time),
```

```
[TotalAmount]) VALUES (7, 18, 23, CAST(N'01:07:42.8966667' AS Time),
CAST(200.00 AS Decimal(10, 2)))
INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (8, 3, 24, CAST(N'05:18:06.7333333' AS Time),
CAST(350.00 AS Decimal(10, 2)))
INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (1008, 19, 1024, CAST(N'18:43:23.4700000' AS Time),
CAST(-175.00 AS Decimal(10, 2)))
INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (1009, 21, 1025, CAST(N'20:49:40.5966667' AS Time),
CAST(-105.00 AS Decimal(10, 2)))
INSERT [dbo].[Transaction] ([TransactionID], [UserID], [ReservationID], [ExitTime],
[TotalAmount]) VALUES (1010, 3, 1026, CAST(N'18:01:07.2433333' AS Time),
CAST(-105.00 AS Decimal(10, 2)))
SET IDENTITY_INSERT [dbo].[Transaction] OFF
GO
SET IDENTITY_INSERT [dbo].[User] ON

INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (1, N'Emad', N'Emad', N'Tariq', N'emad@gmail.com',
N'abc', N'Admin')
INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (2, N'Sofia123', N'Sofia', N'Haider',
N'sofia@gmail.com', N'sofia789', N'Admin')
INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (3, N'ahsan1', N'Ahsan', N'Naeem',
N'Ahsan@gmail.com', N'ahsan789', N'User')
```

```
N'Ahsan@gmail.com', N'ahsan789', N'User')
INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (4, N'SophieM', N'Sophie', N'Miller',
N'sophie.miller@example.com', N'sophie123', N'User')
INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (5, N'RyanK', N'Ryan', N'Khan',
N'ryan.khan@example.com', N'ryan456', N'User')
INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (6, N'EmmaC', N'Emma', N'Clark',
N'emma.clark@example.com', N'emma789', N'User')
INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (7, N'LilyR', N'Lily', N'Roberts',
N'lily.roberts@example.com', N'lily789', N'User')
INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (18, N'lily', N'Emily', N'Alexander',
N'emily@gmail.com', N'emily123', N'User')
INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (19, N'uzi', N'Uzair', N'Faisal', N'uzi@gmail.com',
N'uzi123', N'User')
INSERT [dbo].[User] ([UserID], [Username], [FirstName], [LastName], [Email],
[Password], [Role]) VALUES (21, N'mango', N'Adeel', N'Sharif',
N'mango@gmail.com', N'mango', N'User')
SET IDENTITY_INSERT [dbo].[User] OFF
GO
SET IDENTITY_INSERT [dbo].[Vehicle] ON
```

```
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (1, N'ABC123')
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (2, N'XYZ789')
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (3, N'DEF456')
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (4, N'GHI789')
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (5, N'JKL012')
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (6, N'DEF567')
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (7, N'LMN719')
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (8, N'OKL451')
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (9, N'MNB719')
INSERT [dbo].[Vehicle] ([VehicleID], [LicensePlate]) VALUES (10, N'ORF012')
SET IDENTITY_INSERT [dbo].[Vehicle] OFF
GO
INSERT [dbo].[VehicleOwner] ([VehicleID], [UserID]) VALUES (1, 3)
INSERT [dbo].[VehicleOwner] ([VehicleID], [UserID]) VALUES (2, 4)
INSERT [dbo].[VehicleOwner] ([VehicleID], [UserID]) VALUES (3, 5)
INSERT [dbo].[VehicleOwner] ([VehicleID], [UserID]) VALUES (4, 6)
INSERT [dbo].[VehicleOwner] ([VehicleID], [UserID]) VALUES (5, 7)

ALTER TABLE [dbo].[User] ADD  CONSTRAINT [unq_email] UNIQUE
NONCLUSTERED
(
	[Email] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, ONLINE = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
ALTER TABLE [dbo].[User] ADD  CONSTRAINT [unq_username] UNIQUE
NONCLUSTERED
(
	[Username] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, ONLINE = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

ALTER TABLE [dbo].[Feedback]  WITH CHECK ADD FOREIGN
KEY([ReservationID])
REFERENCES [dbo].[Reservation] ([ReservationID])
GO
ALTER TABLE [dbo].[ParkingLot]  WITH CHECK ADD FOREIGN KEY([SubID])
REFERENCES [dbo].[SubscriptionCategory] ([SubID])
GO
ALTER TABLE [dbo].[ParkingSpace]  WITH CHECK ADD FOREIGN KEY([LotID])
REFERENCES [dbo].[ParkingLot] ([LotID])
GO
ALTER TABLE [dbo].[Reservation]  WITH CHECK ADD FOREIGN KEY([LotID])
REFERENCES [dbo].[ParkingLot] ([LotID])
GO
ALTER TABLE [dbo].[Reservation]  WITH CHECK ADD FOREIGN KEY([SpaceID])
REFERENCES [dbo].[ParkingSpace] ([SpaceID])
GO
```

```
GO
ALTER TABLE [dbo].[Reservation]  WITH CHECK ADD FOREIGN KEY([UserID])
REFERENCES [dbo].[User] ([UserID])
GO
ALTER TABLE [dbo].[Reservation]  WITH CHECK ADD  CONSTRAINT [fk_userid]
FOREIGN KEY([UserID])
REFERENCES [dbo].[User] ([UserID])
GO
ALTER TABLE [dbo].[Reservation] CHECK CONSTRAINT [fk_userid]
GO
ALTER TABLE [dbo].[Transaction]  WITH CHECK ADD FOREIGN
KEY([ReservationID])
REFERENCES [dbo].[Reservation] ([ReservationID])
GO
ALTER TABLE [dbo].[Transaction]  WITH CHECK ADD FOREIGN KEY([UserID])
REFERENCES [dbo].[User] ([UserID])
GO
ALTER TABLE [dbo].[VehicleOwner]  WITH CHECK ADD FOREIGN KEY([UserID])
REFERENCES [dbo].[User] ([UserID])
GO
ALTER TABLE [dbo].[VehicleOwner]  WITH CHECK ADD FOREIGN
KEY([VehicleID])
REFERENCES [dbo].[Vehicle] ([VehicleID])
GO
ALTER TABLE [dbo].[Feedback]  WITH CHECK ADD CHECK  (([Rating]>=(1) AND
[Rating]<=(5)))
GO
```


```
ALTER TABLE [dbo].[Feedback]  WITH CHECK ADD CHECK  (([Rating]>=(1) AND
[Rating]<=(5)))
GO
```


```
ALTER TABLE [dbo].[User]  WITH CHECK ADD CHECK  (([Role]='User' OR
[Role]='Admin'))
```

VIEWS:

```
create view [dbo].[viewUsers]
as
select UserID,Username,FirstName,LastName,Email,Password
from [User]
```

```
create view [dbo].[viewAllFeedbacks]
as
select u.FirstName+' '+u.LastName as 'CustomerName', Comment,
Rating from Feedback f, [User] u,
Reservation r
where r.ReservationID=f.ReservationID
AND r.UserID=u.UserID
```

```
Create view [dbo].[viewUserVehicles]
as
select ReservationID, u.FirstName+' '+u.LastName as 'CustomerName',
VehiclePlate
from [User] u,
Reservation r
where r.UserID=u.UserID
```

1

```
create view [dbo].[viewSubCats]
as
select distinct SubCategory
from SubscriptionCategory
```

```
create view [dbo].[viewReservations]
        as
        select ReservationID, (FirstName+'
'+LastName)'Name',VehiclePlate,LotID,SpaceID,ParkDate,ParkTime
        from Reservation r, [User] u
        where r.userID=u.UserID
```

Stored Procedures:

```
CREATE PROCEDURE [dbo].[UpdateVehicleOwner]
    @VehicleID INT,
    @UserID INT,
    @NewVehicleID INT,
    @NewUserID INT
AS
BEGIN
    UPDATE VehicleOwner
    SET VehicleID = @NewVehicleID,
        UserID = @NewUserID
    WHERE VehicleID = @VehicleID AND UserID = @UserID;
END;
```

```
CREATE PROCEDURE [dbo].[UpdateSubscriptionCategory]
    @SubID INT,
    @SubCategory VARCHAR(25),
    @HourlyRate DECIMAL(10, 2)
AS
BEGIN
    UPDATE SubscriptionCategory
    SET SubCategory = @SubCategory,
        HourlyRate = @HourlyRate
    WHERE SubID = @SubID;
END;
```

```
CREATE PROCEDURE [dbo].[UpdateVehicle]
    @VehicleID INT,
    @NewLicensePlate VARCHAR(20)
AS
BEGIN
    UPDATE Vehicle
    SET LicensePlate = @NewLicensePlate
    WHERE VehicleID = @VehicleID;
END;
```

```
GO
CREATE PROCEDURE [dbo].[UpdateUser]
    @UserID INT,
    @Username VARCHAR(50),
    @FirstName VARCHAR(50),
    @LastName VARCHAR(50),
    @Email VARCHAR(25),
    @Password VARCHAR(25),
    @Role VARCHAR(20)
AS
BEGIN
    UPDATE [User]
    SET Username = @Username,
        FirstName = @FirstName,
        LastName = @LastName,
        Email = @Email,
        Password = @Password,
        Role = @Role
    WHERE UserID = @UserID;
END;
```

```
CREATE PROCEDURE [dbo].[UpdateReservation]
    @ReservationID INT,
    @UserID INT,
    @VehiclePlate varchar(25),
    @LotID INT,
    @SpaceID INT,
    @ParkDate DATE,
    @ParkTime TIME
AS
BEGIN
    UPDATE Reservation
    SET UserID = @UserID,
        VehiclePlate = @VehiclePlate,
        LotID = @LotID,
        SpaceID = @SpaceID,
        ParkDate = @ParkDate,
        ParkTime = @ParkTime   WHERE ReservationID = @ReservationID;
END;
```

```
CREATE PROCEDURE [dbo].[proc_Sign_up]
    @username VARCHAR(50),
    @FirstName VARCHAR(50),
    @LastName VARCHAR(50),
    @Email VARCHAR(50),
    @Password VARCHAR(50),
    @usertype VARCHAR(20)
AS
BEGIN
   INSERT INTO [User] (Username, FirstName, LastName, Email, Password, Role)
    VALUES (@username, @FirstName, @LastName, @Email, @Password, @UserType);

END;
```

```
CREATE procedure [dbo].[requestParking]
    @uid int, @liscenceplate varchar(25), @lotid int ,@spaceid int
    as
    begin
    insert into ParkingSpace(LotID,SpaceID)values(@lotid,@spaceid)
    insert into Reservation (UserID,VehiclePlate,LotID,SpaceID,ParkDate,ParkTime) values (@uID,@liscenceplate,@lotid,@spaceid,getdate(),CONVERT(VA
    end
```

```
--
  CREATE procedure [dbo].[submitFeedback]
      @ResID int, @Comment varchar(50), @Rating int
      as
      begin
      Insert into Feedback(ReservationID,Date,Comment,Rating) values (@ResID,getdate(),@Comment,@Rating)
      end
```

```
--
CREATE procedure [dbo].[InsertPS]
@LotID int,
@SpaceID int OUTPUT
as
begin

select Top 1 @SpaceID=SpaceID from ParkingSpace
where LotID=@LotID
order by SpaceID desc;

set @SpaceID=@SpaceID+1;
 print @SpaceID
insert into ParkingSpace(SpaceID,LotID) values (@SpaceID,@LotID)
return @SpaceID;
end
```

```
GO
CREATE PROCEDURE [dbo].[DeleteUser]
    @UserID INT
AS
BEGIN
    DELETE FROM [User]
    WHERE UserID = @UserID;
END;
```

```
GO
CREATE PROCEDURE [dbo].[DeleteVehicle]
    @VehicleID INT
AS
BEGIN
    DELETE FROM Vehicle
    WHERE VehicleID = @VehicleID;
END;
```

```
CREATE PROCEDURE [dbo].[DeleteVehicleOwner]
    @VehicleID INT,
    @UserID INT
AS
BEGIN
    DELETE FROM VehicleOwner
    WHERE VehicleID = @VehicleID AND UserID = @UserID;
END;
```

```
CREATE PROCEDURE [dbo].[DeleteReservation]
    @ReservationID INT
AS
BEGIN
    DELETE FROM Reservation
    WHERE ReservationID = @ReservationID
END
```

```
GO
CREATE PROCEDURE [dbo].[DeleteFeedback]
    @FeedbackID INT
AS
BEGIN
    DELETE FROM Feedback
    WHERE FeedbackID = @FeedbackID
END
```

```
CREATE procedure [dbo].[checkout]
    @userid int,@resID int,@exithour int,@amount int
    as
    begin
    insert into [transaction](UserID,ReservationID,ExitTime,TotalAmount) values (@userid,@resID,dateadd(hour,@exithour,getdate()),@amount)
    end
```

FUNCTIONS:

```
CREATE function [dbo].[getUserReservationIDs]
    (@username varchar(25))
    returns table
    as
    return
    (select ReservationID from Reservation r, [User] u
    where Username=@username AND u.UserID=r.UserID)
```

```
CREATE function [dbo].[getUserFeedback]
(@username varchar(25))
returns table
as
return
(select top 1 ReservationID from Reservation r, [User] u
where Username=@username AND u.UserID=r.UserID
order by ReservationID desc)
```

```
CREATE function [dbo].[getUserDetail]
(@uname varchar(25),@Password varchar(25))
returns table
as
return
(Select * from [User]
where username=@uname AND Password=@Password)
```

```sql
CREATE function [dbo].[getFeedback]()
 returns table
 as
 return (Select (FirstName+' '+LastName)'Name', Comment from Feedback f, [User] u, Reservation r
 where f.ReservationID=r.ReservationID AND u.UserID=r.ReservationID)
```

```sql
CREATE FUNCTION [dbo].[loginUser]
(@uname VARCHAR(25), @Password VARCHAR(25), @Role varchar(25))
RETURNS VARCHAR(10)
AS
BEGIN
    DECLARE @userCount INT, @Result VARCHAR(10);

    SELECT @userCount = COUNT(*)
    FROM [User]
    WHERE Username = @uname AND Password = @Password and Role=@Role;

    -- Return 'Success' if a matching user is found, otherwise 'Fail'
    IF @userCount > 0
        SET @Result = 'Success';
    ELSE
        SET @Result = 'Fail';

    RETURN @Result;
END;
```

```sql
CREATE function [dbo].[getUserReservationID]
   (@username varchar(25))
   returns int
   as
   begin
   declare @ResID int;
   select top 1 @ResID=ReservationID from Reservation r, [User] u
   where Username=@username AND u.UserID=r.UserID
   order by ReservationID desc
   return @ResID
   end
```

```
CREATE function [dbo].[getTotalAmount]
(@exithours int, @resid int )
returns decimal (5,2)
as
begin
declare @charges decimal(5,2),@parktime time;
select @parktime=ParkTime from Reservation where ReservationID=@resID


select @charges=datediff(hour,CONVERT(TIME, dateadd(hour,@exithours,getdate()), 108),@parktime)*s.HourlyRate
from Reservation r, ParkingSpace ps, ParkingLot pl, SubscriptionCategory s
where r.LotID = ps.LotID
AND r.SpaceID=ps.SpaceID
and pl.LotID=ps.LotID
AND pl.SubID=s.SubID
AND r.ReservationID=@resID
return @charges
end
```

```
--
CREATE function [dbo].[getReservedUserID]
                (@username varchar(25))
                returns int
        as
        begin
        declare @resID int;
        select top 1 @resID=ReservationID
        from Reservation r, [User] u
        where Username=@username AND u.UserID=r.UserID
        order by ReservationID desc
        return @resID
        end
```

```
CREATE function [dbo].[getLotBySub]
(@SubCat varchar(25))
returns int
as
begin
declare @LotID int;
select @LotID=LotID from ParkingLot pl,  SubscriptionCategory sc
where sc.SubID=pl.subID
AND sc.SubCategory=@SubCat
return @LotID;
 end
```

```
CREATE function [dbo].[getAvailableSpaceID]
(@LotID int)
returns int
as
begin
declare @SpaceID int
 select Top 1 @SpaceID=SpaceID from ParkingSpace
where LotID=@LotID
order by SpaceID desc;

set @SpaceID=@SpaceID+1;
return @SpaceID
end
```

TRIGGERS:

```
create trigger trig_checkcap
on ParkingSpace
for Insert
as
begin
if exists(Select * from ParkingSpace ps, ParkingLot pl, SubscriptionCategory sc
where ps.LotID=pl.LotID AND sc.SubID=pl.SubID
AND ps.SpaceID>pl.Capacity)
begin
Print('Error! Cannot reserve, capacity exceeded!')
rollback
end
end
```

Conclusion:

- **Evaluation of the project's success in meeting its objectives**

  Our Parking Management system project has proven to be a resounding success, meeting and in many instances, surpassing its initial objectives. Here's how the project proves to be a success:

  - **Ease of Navigation**:

    From the get-go, users have been greeted with a clean and straightforward interface. Clear call to-action buttons and a logical flow from User selection to final payment have ensured that even the least tech-savvy users find the process hassle-free.

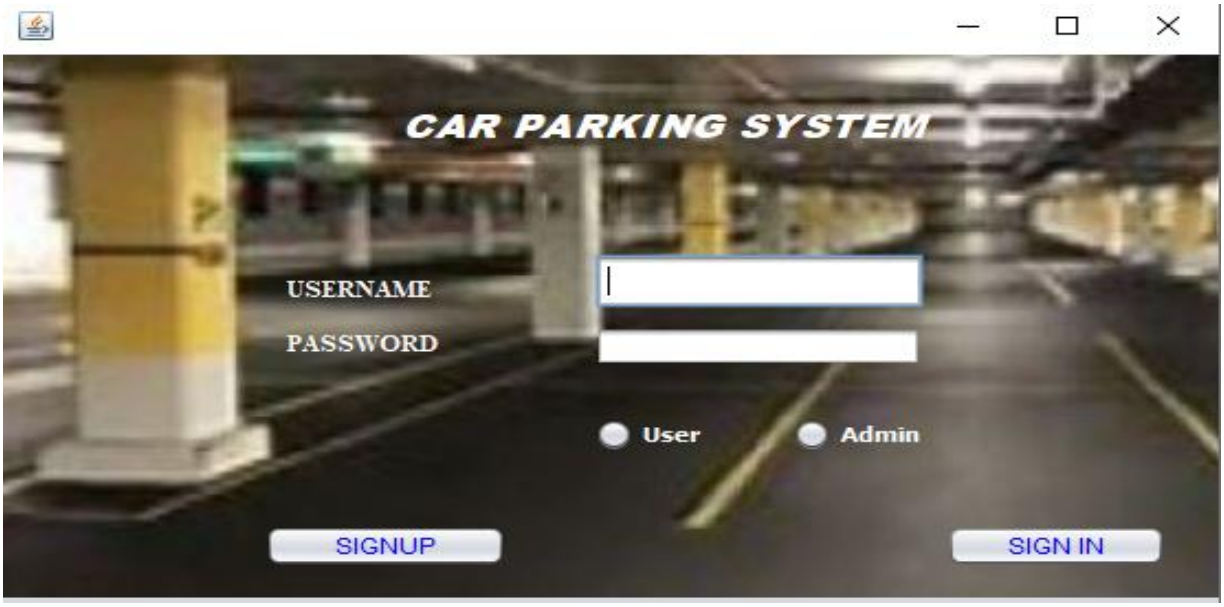
  - **Responsive Design:**

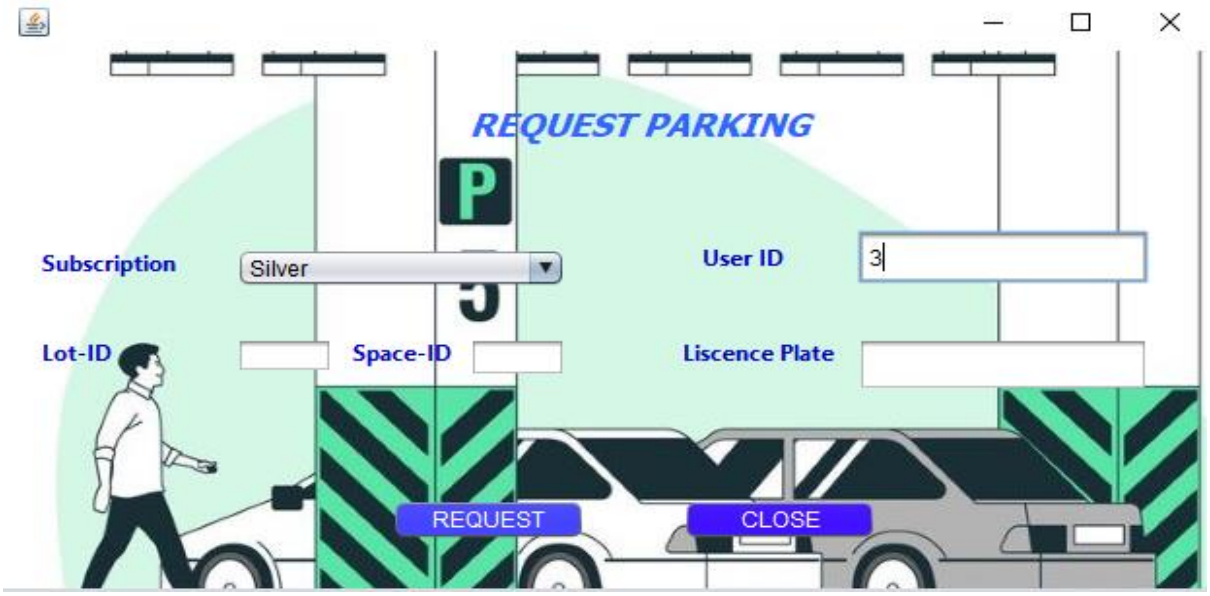
    The user side is fully responsive, ensuring that admins can manage the Parking operations on the go, without losing functionality or experiencing a drop in performance.

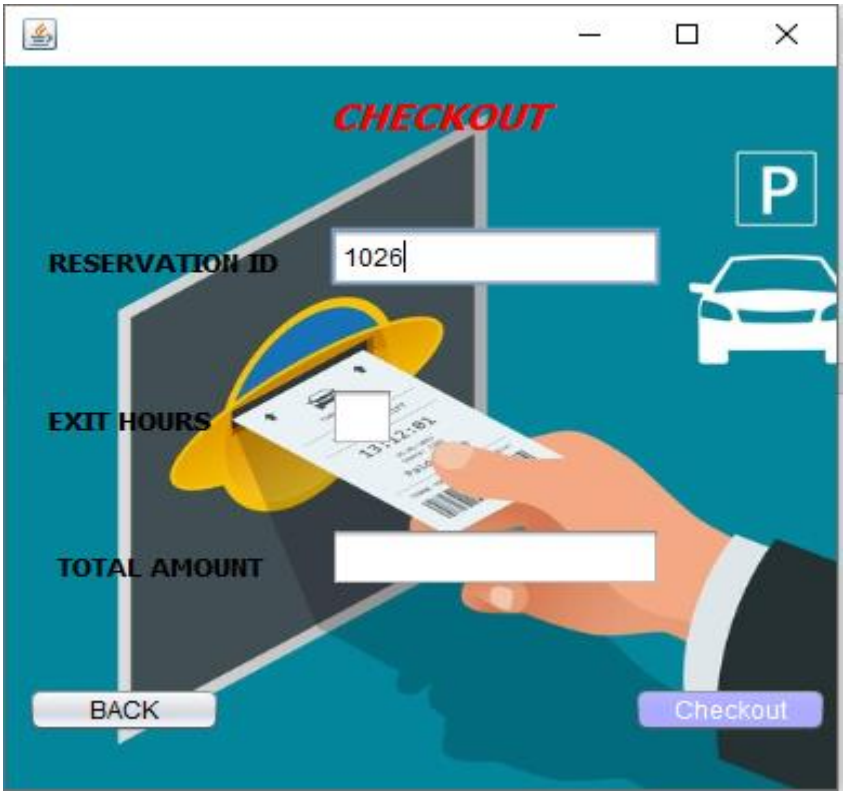
  - **Streamlined Management:**

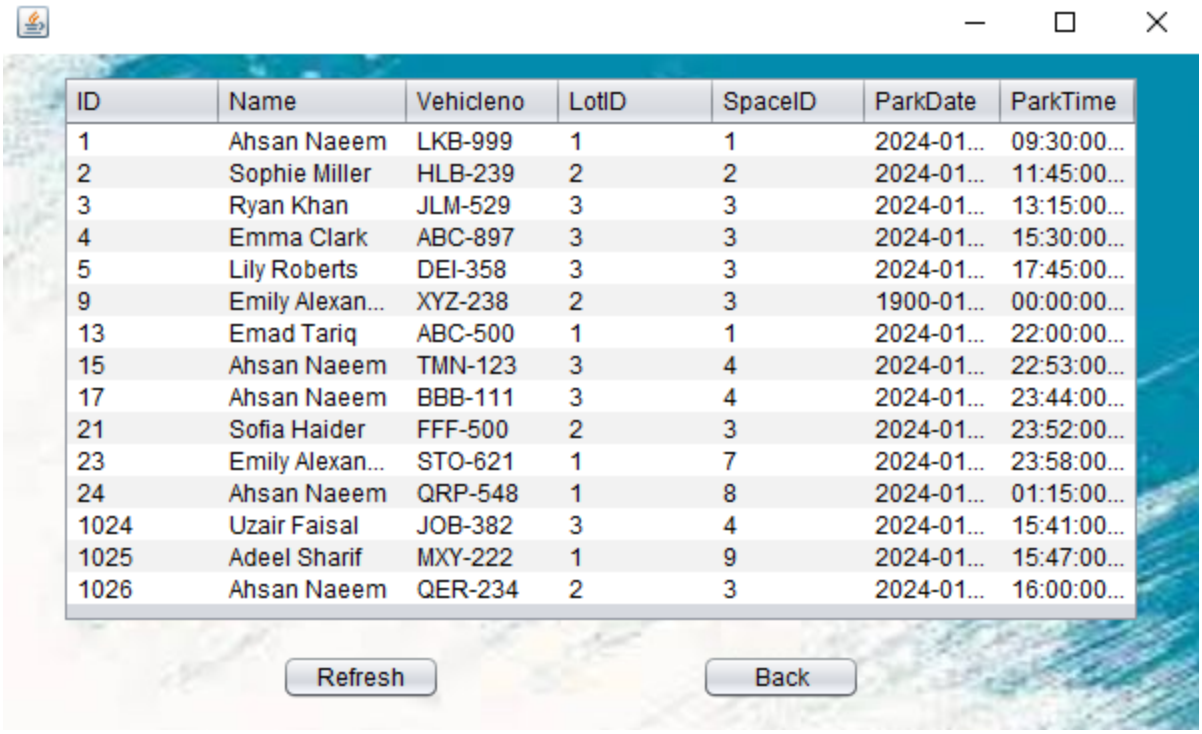
    Admins can add new subscription, Parking Space update details, and remove showings with a few clicks. The interface ensures that changes are reflected in real time, allowing for dynamic management of Parking Spaces

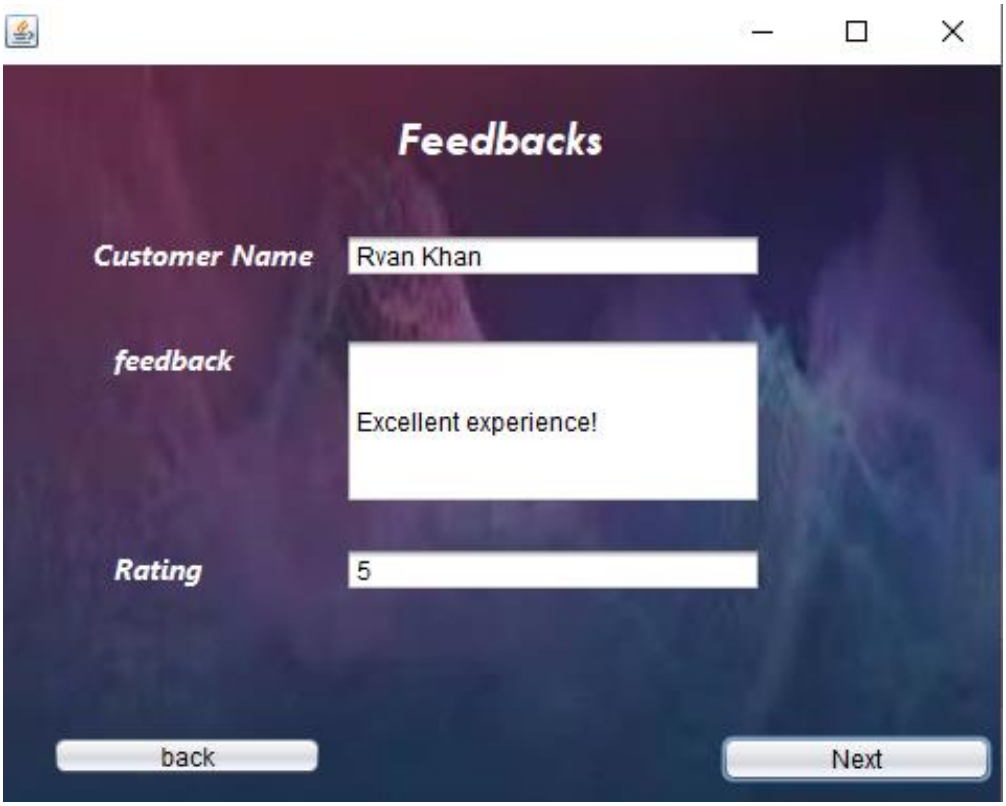
- **Screenshots of major modules' outputs**




REQUEST PARKING
Subscription
Silver
User ID
3
Lot-ID
Space-ID
Liscence Plate
REQUEST
CLOSE


CHECKOUT
RESERVATION ID
1026
EXIT HOURS
TOTAL AMOUNT
BACK
Checkout


ID
Name
Vehicleno
LotID
SpaceID
ParkDate
ParkTime
1 Ahsan Naeem LKB-999 1 1 2024-01... 09:30:00...
2 Sophie Miller HLB-239 2 2 2024-01... 11:45:00...
3 Ryan Khan JLM-529 3 3 2024-01... 13:15:00...
4 Emma Clark ABC-897 3 3 2024-01... 15:30:00...
5 Lily Roberts DEI-358 3 3 2024-01... 17:45:00...
9 Emily Alexan... XYZ-238 2 3 1900-01... 00:00:00...
13 Emad Tariq ABC-500 1 1 2024-01... 22:00:00...
15 Ahsan Naeem TMN-123 3 4 2024-01... 22:53:00...
17 Ahsan Naeem BBB-111 3 4 2024-01... 23:44:00...
21 Sofia Haider FFF-500 2 3 2024-01... 23:52:00...
23 Emily Alexan... STO-621 1 7 2024-01... 23:58:00...
24 Ahsan Naeem QRP-548 1 8 2024-01... 01:15:00...
1024 Uzair Faisal JOB-382 3 4 2024-01... 15:41:00...
1025 Adeel Sharif MXY-222 1 9 2024-01... 15:47:00...
1026 Ahsan Naeem QER-234 2 3 2024-01... 16:00:00...
Refresh
Back


Feedbacks
Customer Name
Rvan Khan
feedback
Excellent experience!
Rating
5
back
Next