**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** 2](#_Toc155660084)

[Methodology: 2](#_Toc155660086)

[Implementation: 2](#_Toc155660087)

[Conclusion: 2](#_Toc155660088)

# **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.

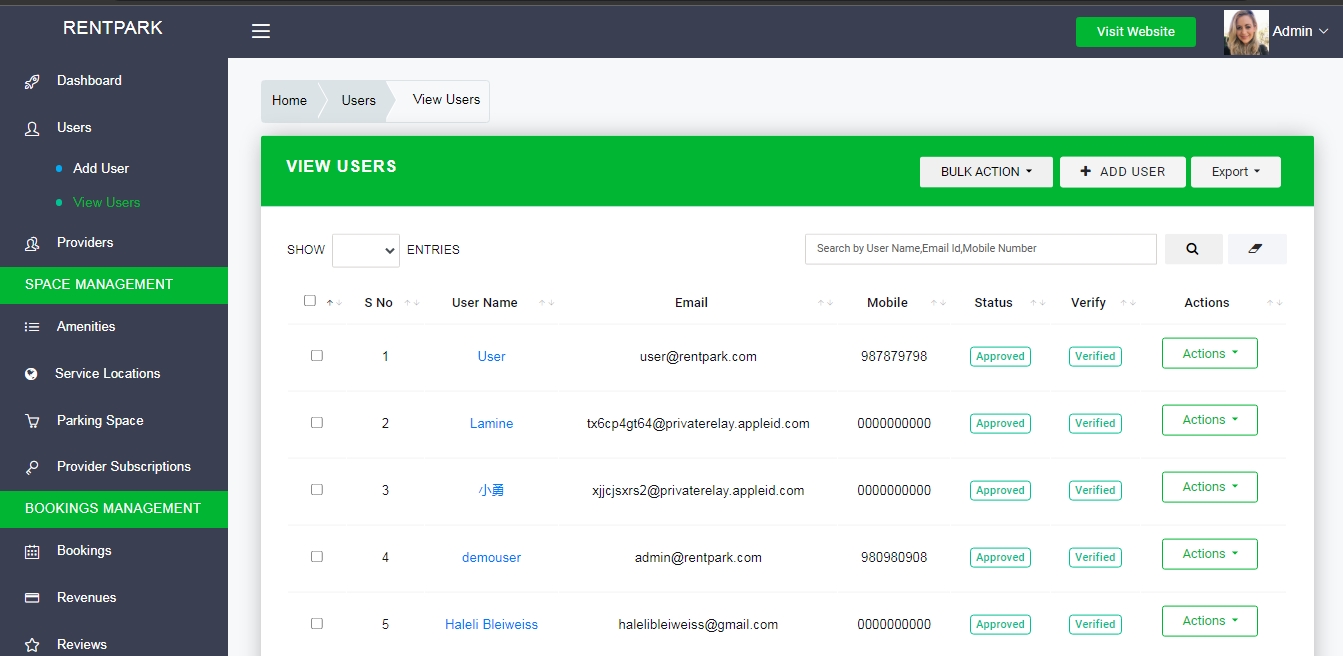
Functionalities:

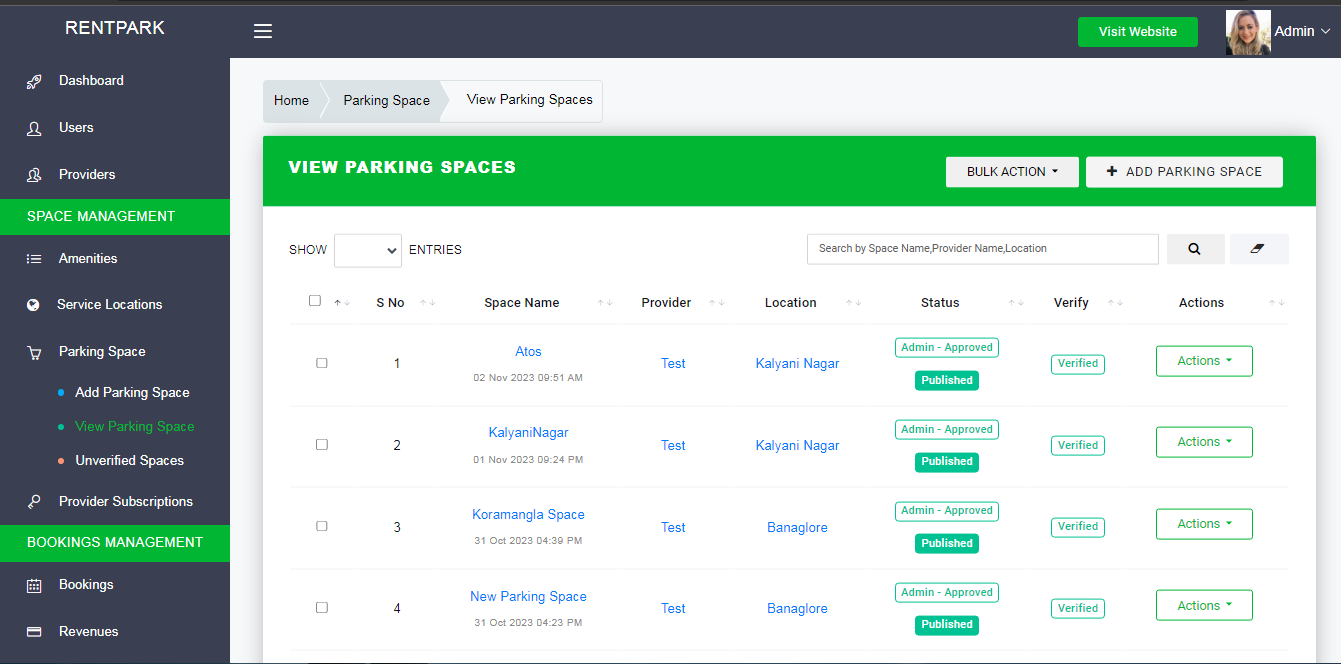
1. **Availability Tracking:**
   * Allows users to check the availability of parking spaces in different lots.
2. **Parking History:**
   * Provides a history of entry and exit transactions for a specific vehicle.
3. **Revenue Generation:**
   * Calculates and reports the total revenue generated by the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

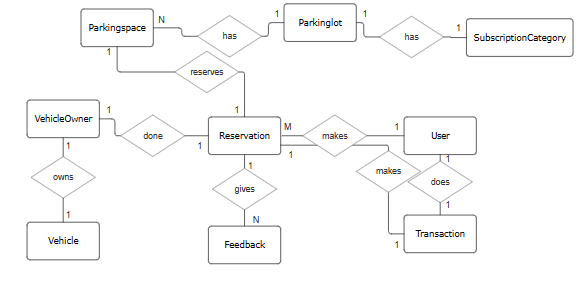
. A user having revervation gives atleast one feedback (one to many relationship)

. 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)

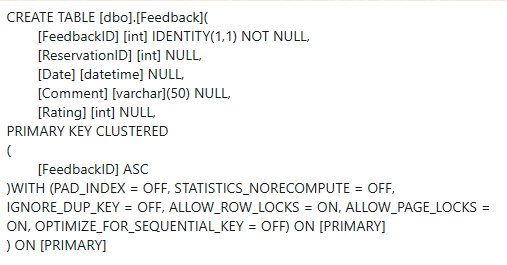
-SubCategory (varchar(25))

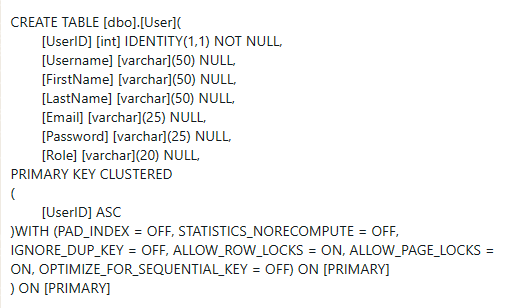
-HourlyRate (decimal(10, 2))

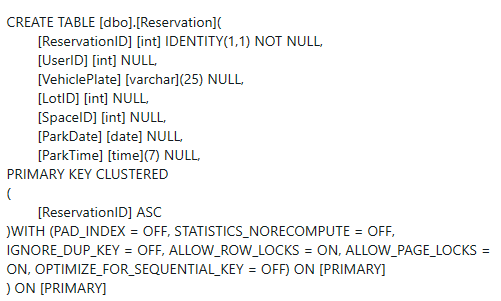
**SQL Server Schema Diagram**

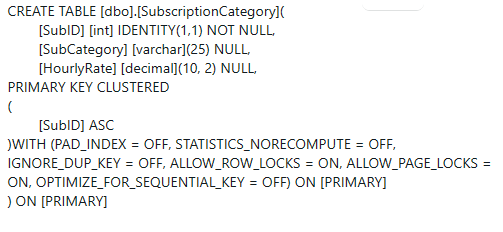


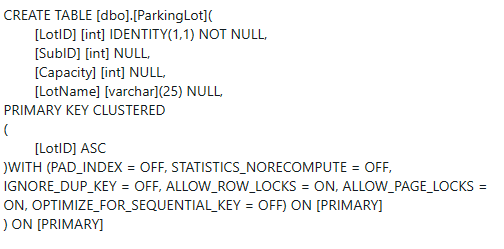
**DDL:**

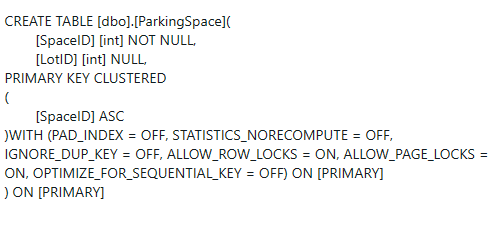


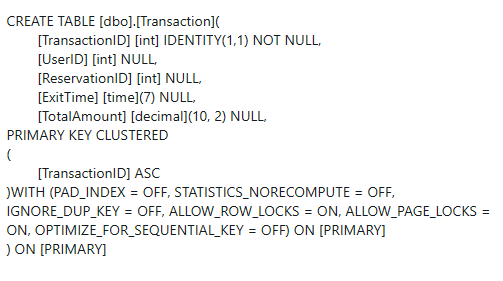


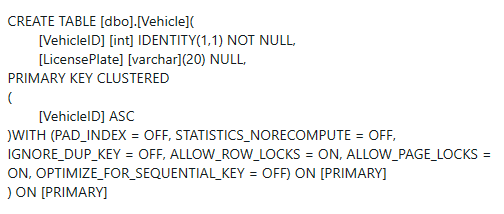


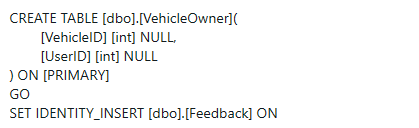






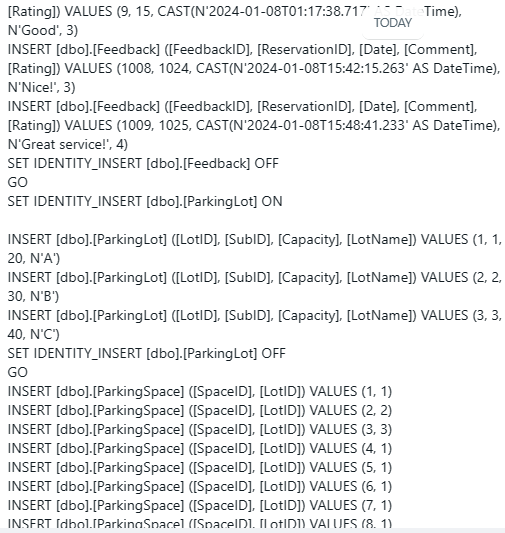






**DML:**







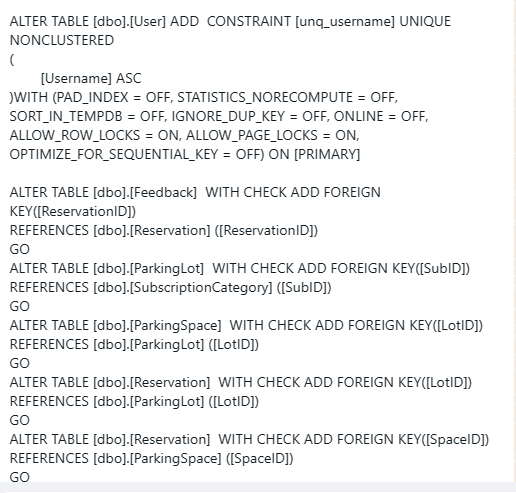


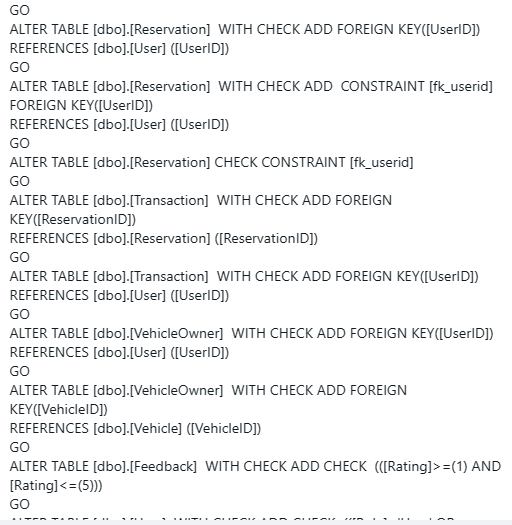


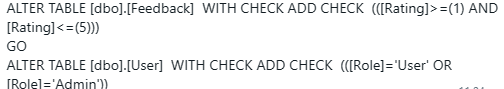




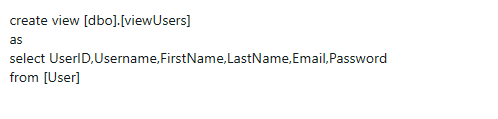


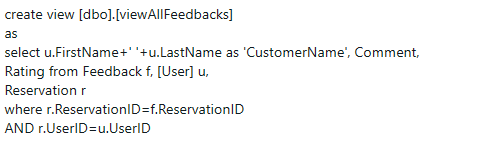


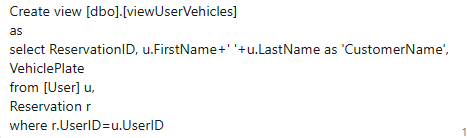


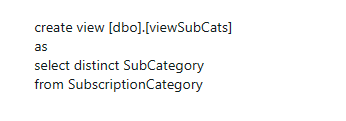


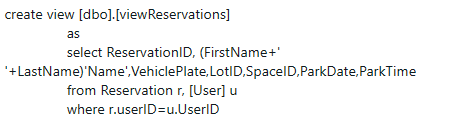
**VIEWS:**

****

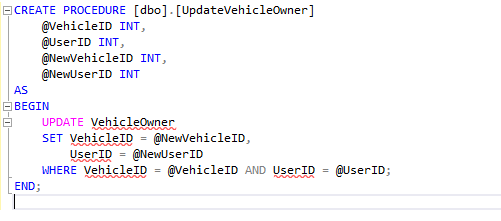
****

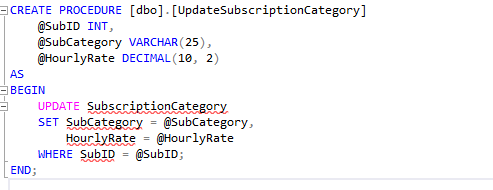
****

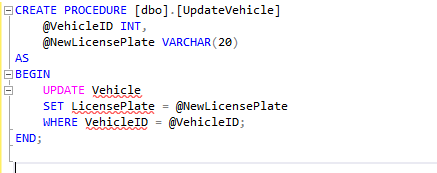
****

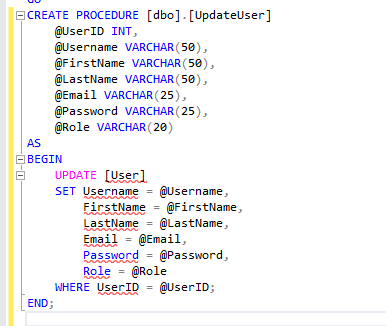
****

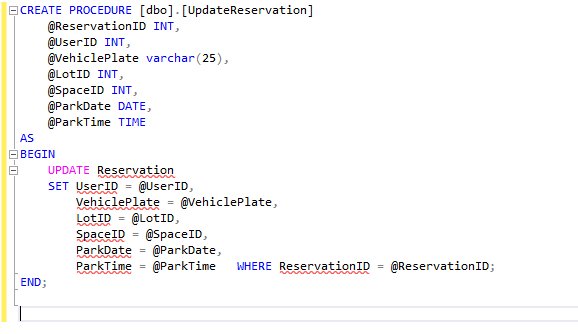
**Stored Procedures:**

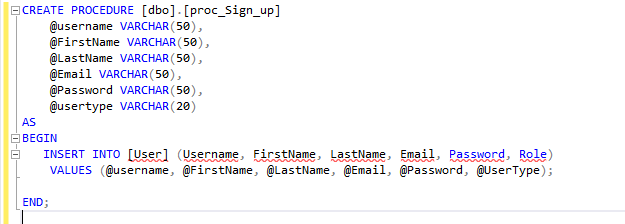


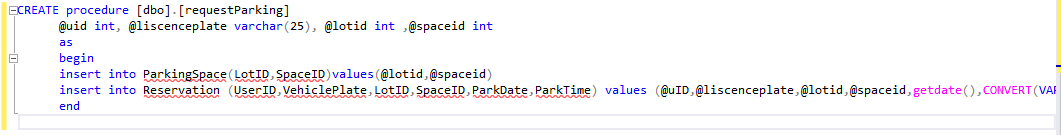


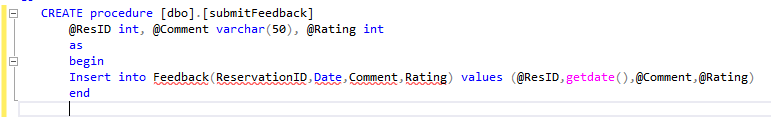


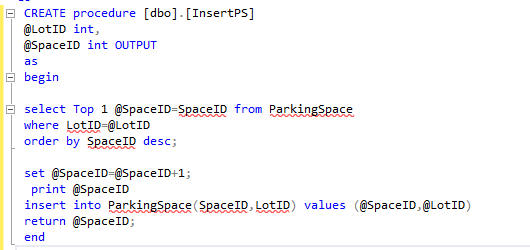


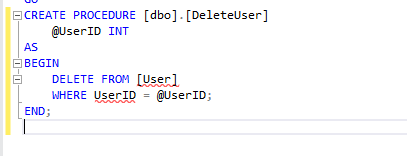


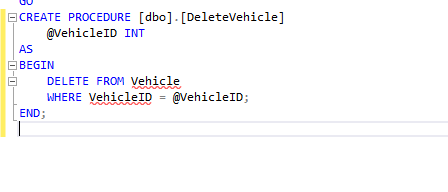


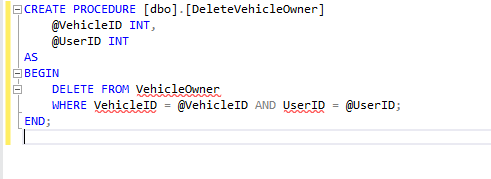


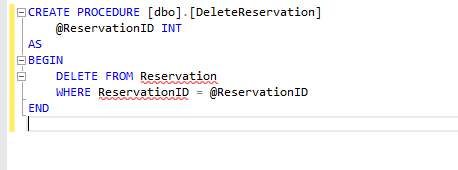


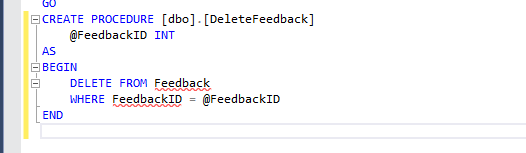


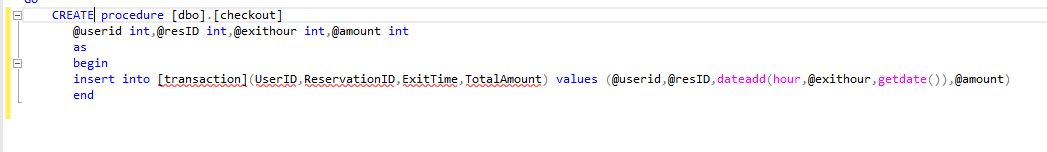




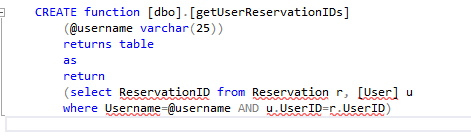


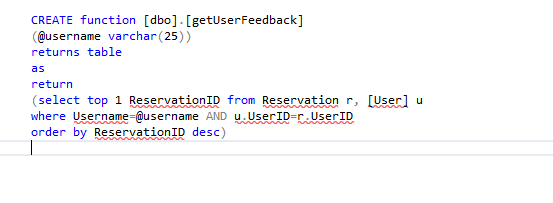


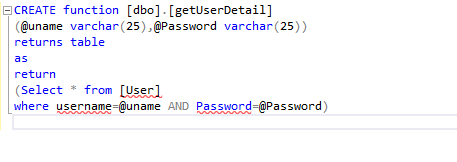


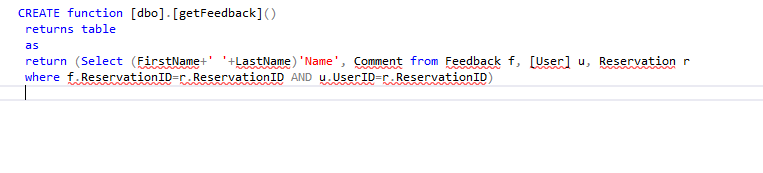


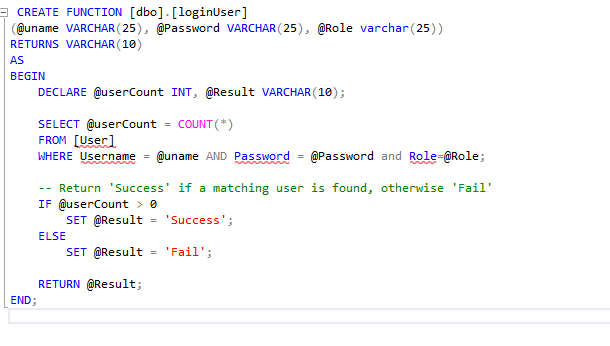
**FUNCTIONS:**

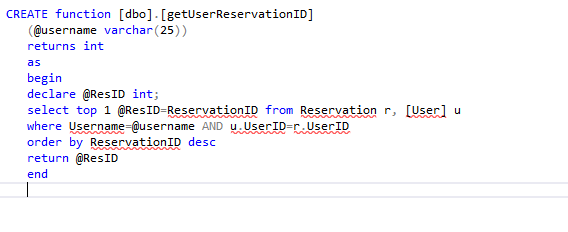


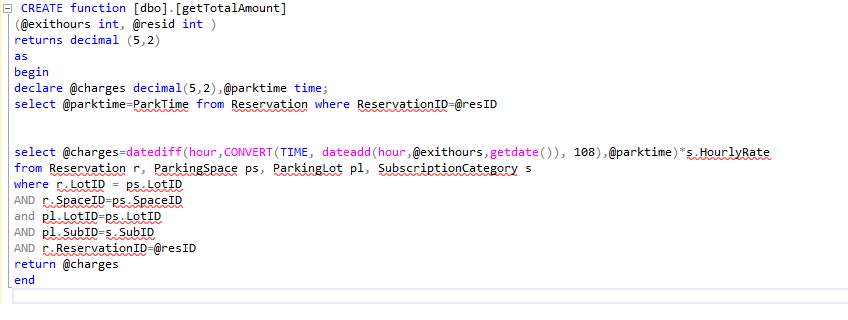


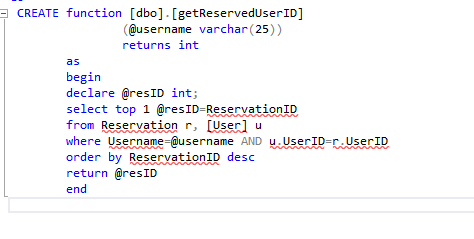


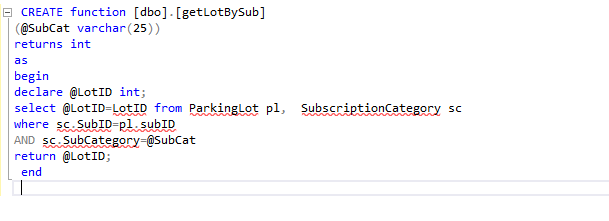


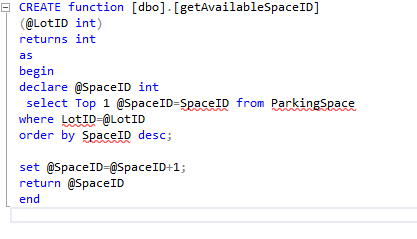




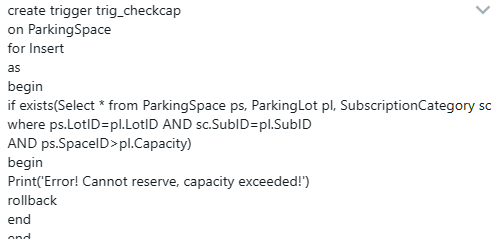








TRIGGERS:



# 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**

