Why Develop a Database ??

Problems Without Using a Database

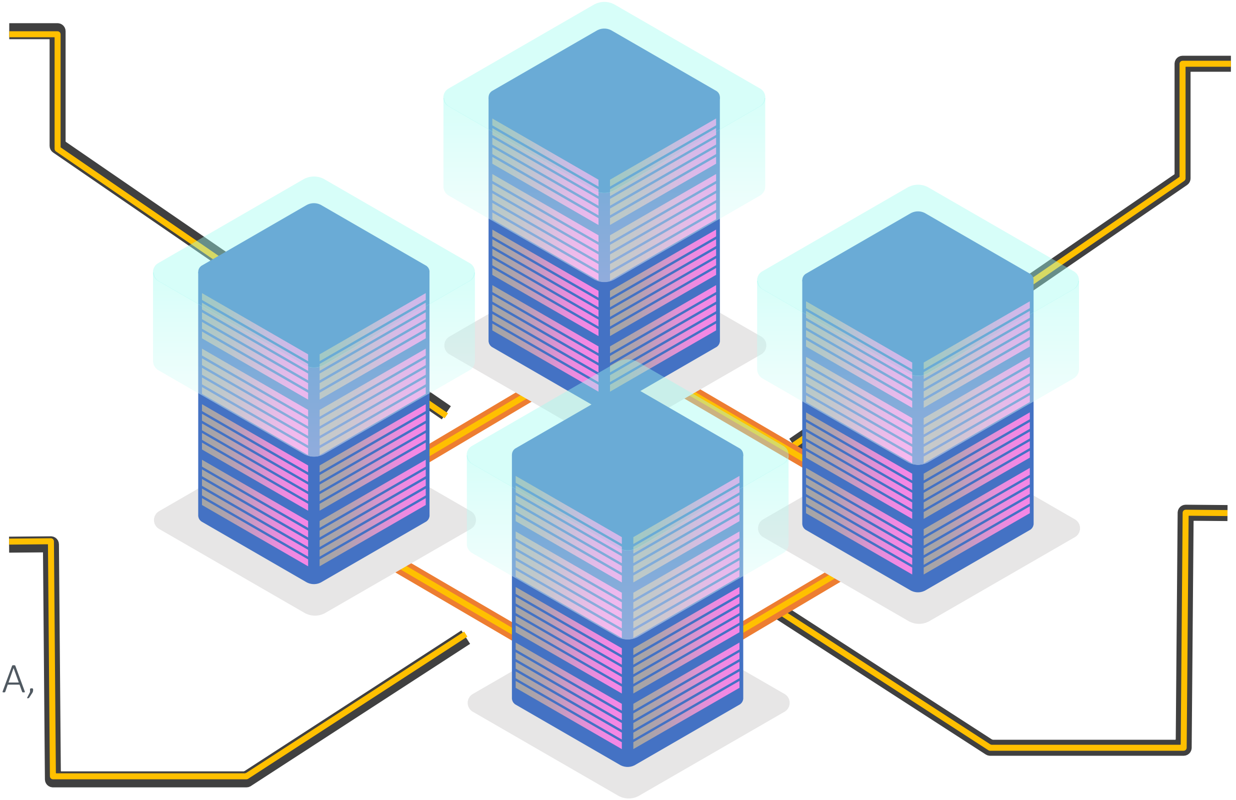
- Data Volatility
  - Data stored in memory or files is easily lost during shutdowns or crashes.
  - No reliable history or tracking of parking activity.
- Inconsistency and Redundancy
  - Manually managed or scattered data can lead to
    - Duplicate entries
    - Conflicting status of the same parking space.
- Limited Scalability
  - Hard to handle more users or spaces without a structured system.
  - As the project grows, file-based or manual storage becomes slow and error-prone.

The Role of the Database in the System

Centralized Data Management

- All user info, space statuses, and session records are stored in one system.
- Easy to access and manage via SQL queries.

Can connect with other apps and systems (Python, SCADA, mobile apps, etc.)



Scalability

Easily handles more:

1. Users
2. Parking spaces
3. Sessions

Data Integrity and Security

- Constraints (e.g., primary/foreign keys, unique emails) ensure data validity.
- SQL Server supports user authentication and role-based access control.

Database Design - Core Entities

Entities to Manage

1. **Users:** Contains user-related information like login credentials.

   {user_id , user_email, user_password ,otp_code, otp_expiry}

2. **Parking Spaces (Places):** Tracks the availability and status of each parking space.

   {place_id, busy (status of the parking space)}

3. **Parking Sessions:** Keeps track of active parking sessions, including entry/exit times and associated users and spaces.

   {session_id, entrytime, exittime, session_user_id, session_place_id}

Database Tables

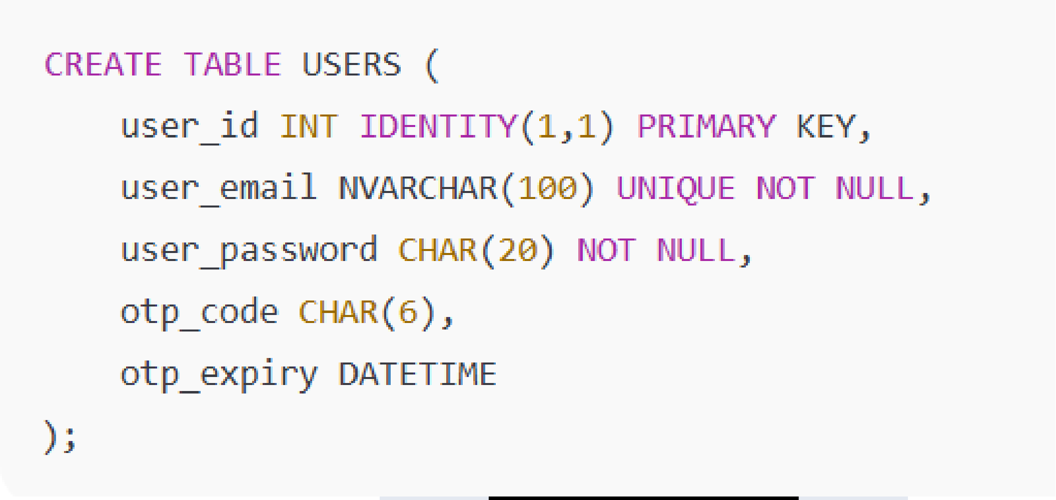
❖ Users Table

purpose

Stores users’ credentials and OTP details

Key

user_id as the primary key (auto-incremented)



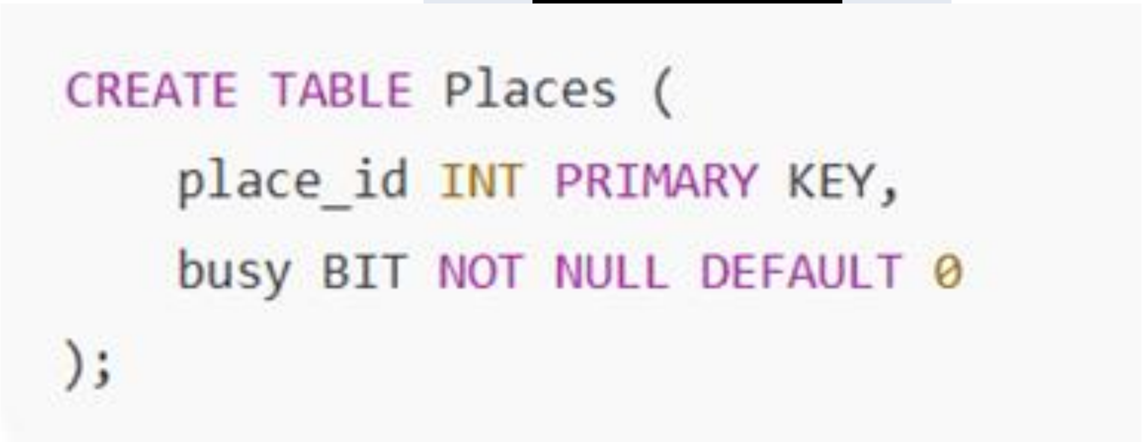
```
CREATE TABLE USERS (
    user_id INT IDENTITY(1,1) PRIMARY KEY,
    user_email NVARCHAR(100) UNIQUE NOT NULL,
    user_password CHAR(20) NOT NULL,
    otp_code CHAR(6),
    otp_expiry DATETIME
);
```

Database Tables

❖ Places Table

purpose

Tracks the status (busy or free) of parking spaces.



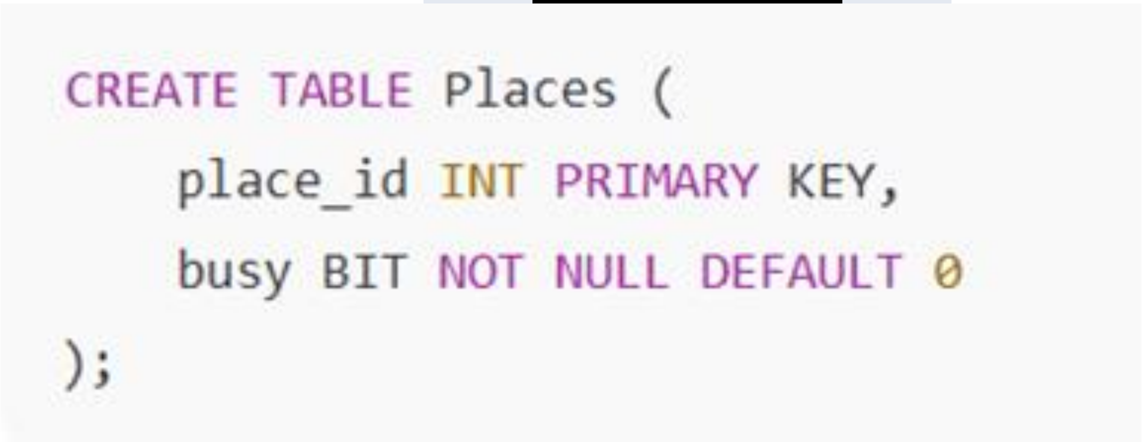
```
CREATE TABLE Places (
    place_id INT PRIMARY KEY,
    busy BIT NOT NULL DEFAULT 0
);
```

Database Tables

❖ Parking Session Table

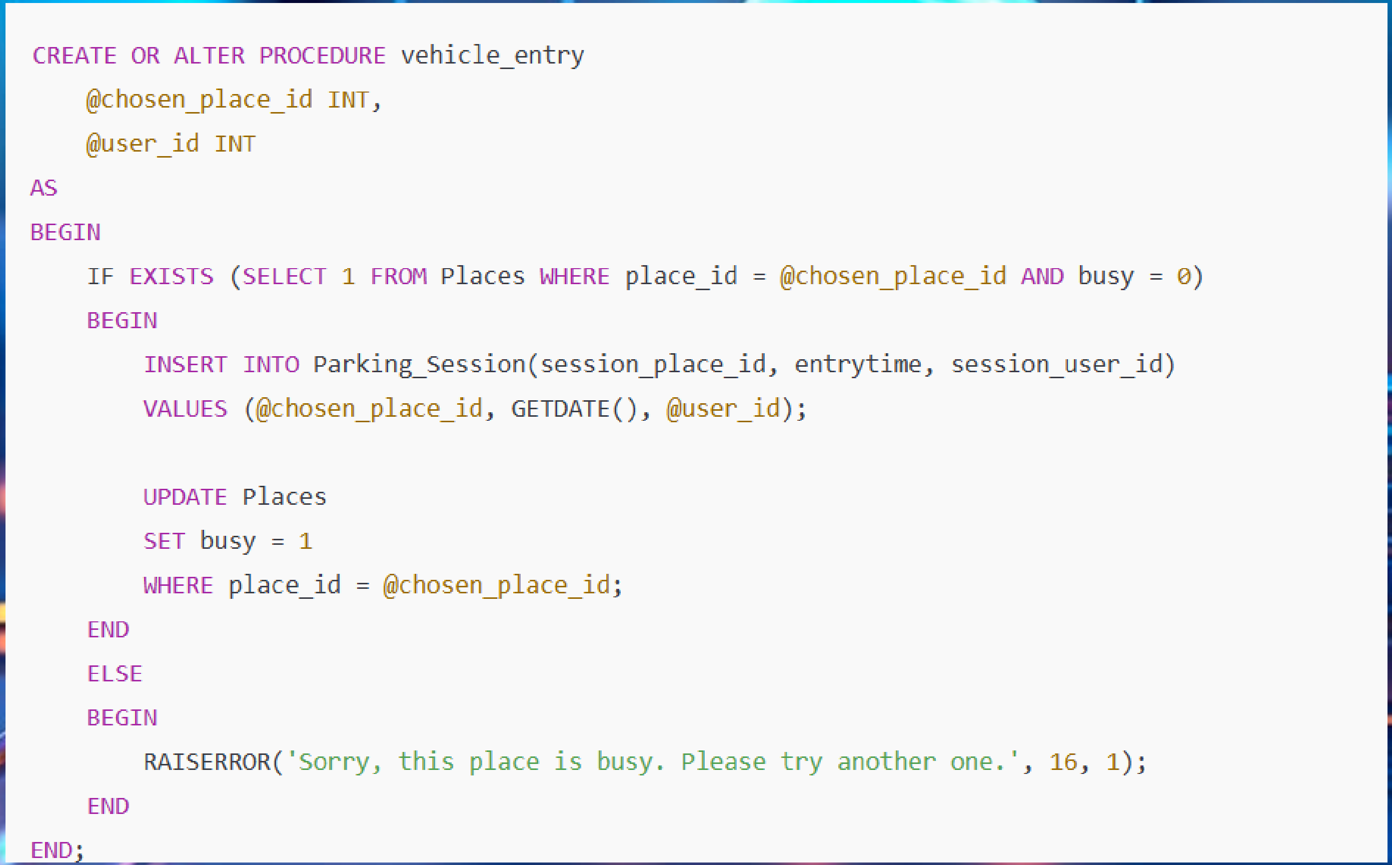
purpose

Records parking sessions for users, linking them to specific parking spots and storing timestamps for entry/exit.

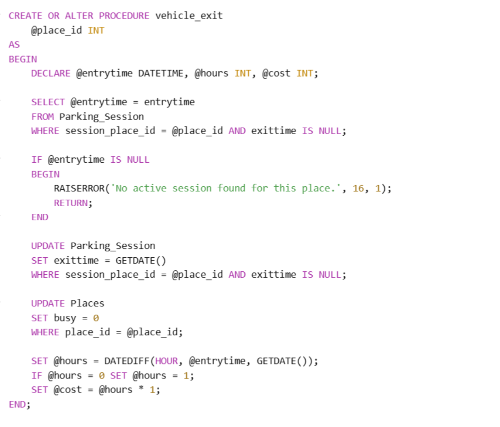


```
CREATE TABLE Places (
    place_id INT PRIMARY KEY,
    busy BIT NOT NULL DEFAULT 0
);
```

Vehicle Entry Procedure

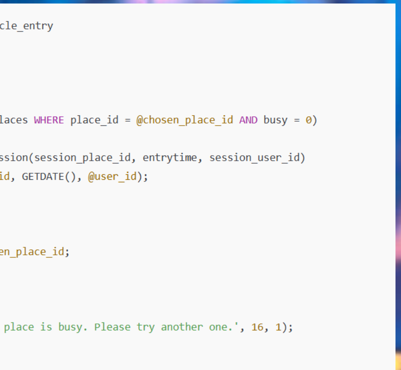


```sql
CREATE OR ALTER PROCEDURE vehicle_entry
    @chosen_place_id INT,
    @user_id INT
AS
BEGIN
    IF EXISTS (SELECT 1 FROM Places WHERE place_id = @chosen_place_id AND busy = 0)
    BEGIN
        INSERT INTO Parking_Session(session_place_id, entrytime, session_user_id)
        VALUES (@chosen_place_id, GETDATE(), @user_id);

        UPDATE Places
        SET busy = 1
        WHERE place_id = @chosen_place_id;
    END
    ELSE
    BEGIN
        RAISERROR('Sorry, this place is busy. Please try another one.', 16, 1);
    END
END;
```

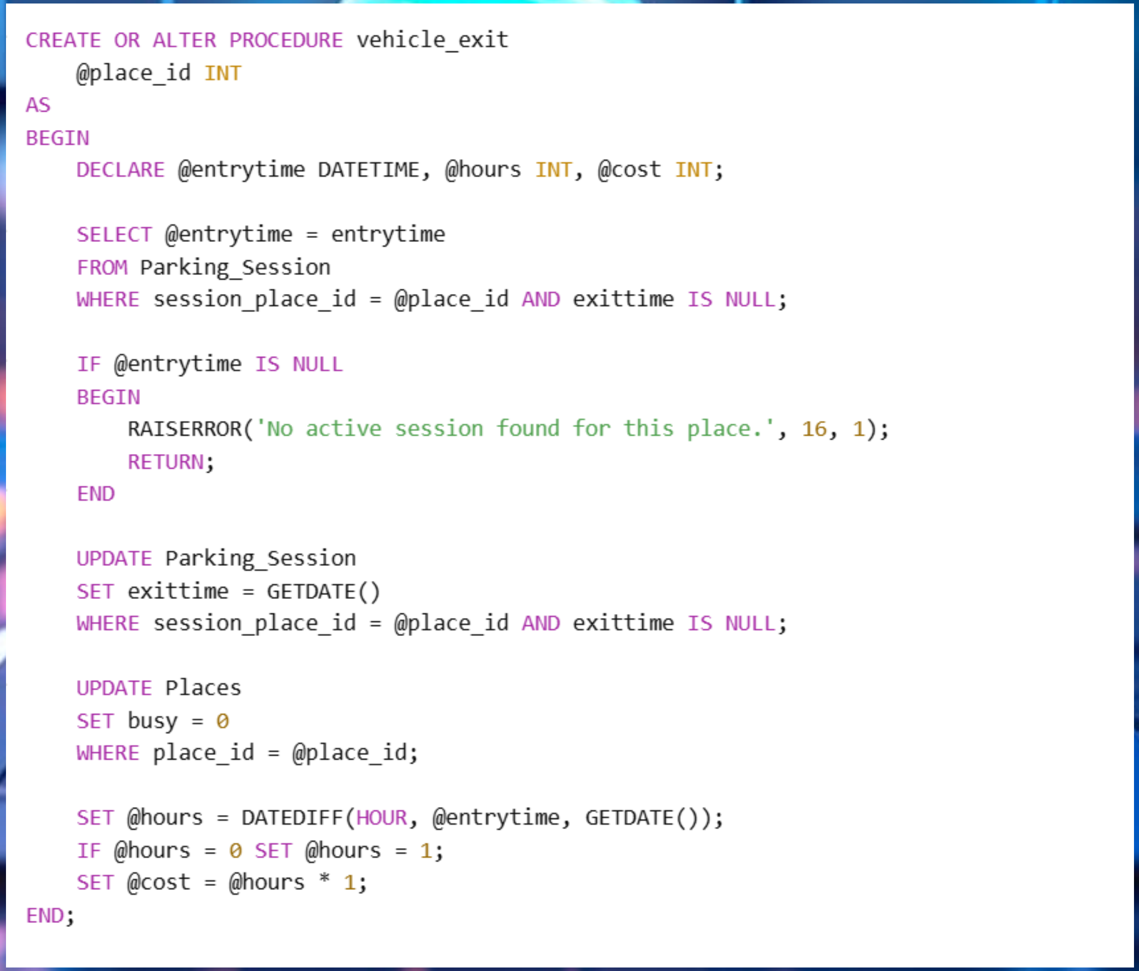


```sql
CREATE OR ALTER PROCEDURE vehicle_exit
    @place_id INT
AS
BEGIN
    DECLARE @entrytime DATETIME, @hours INT, @cost INT;

    SELECT @entrytime = entrytime
    FROM Parking_Session
    WHERE session_place_id = @place_id AND exittime IS NULL;

    IF @entrytime IS NULL
    BEGIN
        RAISERROR('No active session found for this place.', 16, 1);
        RETURN;
    END

    UPDATE Parking_Session
    SET exittime = GETDATE()
    WHERE session_place_id = @place_id AND exittime IS NULL;

    UPDATE Places
    SET busy = 0
    WHERE place_id = @place_id;

    SET @hours = DATEDIFF(HOUR, @entrytime, GETDATE());
    IF @hours = 0 SET @hours = 1;
    SET @cost = @hours * 1;
END;
```

Vehicle Exit Procedure



```
cle_entry



laces WHERE place_id = @chosen_place_id AND busy = 0)

ssion(session_place_id, entrytime, session_user_id)
id, GETDATE(), @user_id);



en_place_id;



 place is busy. Please try another one.', 16, 1);
```



```
CREATE OR ALTER PROCEDURE vehicle_exit
    @place_id INT
AS
BEGIN
    DECLARE @entrytime DATETIME, @hours INT, @cost INT;

    SELECT @entrytime = entrytime
    FROM Parking_Session
    WHERE session_place_id = @place_id AND exittime IS NULL;

    IF @entrytime IS NULL
    BEGIN
        RAISERROR('No active session found for this place.', 16, 1);
        RETURN;
    END

    UPDATE Parking_Session
    SET exittime = GETDATE()
    WHERE session_place_id = @place_id AND exittime IS NULL;

    UPDATE Places
    SET busy = 0
    WHERE place_id = @place_id;

    SET @hours = DATEDIFF(HOUR, @entrytime, GETDATE());
    IF @hours = 0 SET @hours = 1;
    SET @cost = @hours * 1;
END;
```

Database Procedures in Action

⇒ Vehicle Entry:

- A user enters the parking slot, and the system checks if the chosen spot is available.
- If yes, a new session is created in Parking_Session, and the Places table is updated to show the spot is occupied.

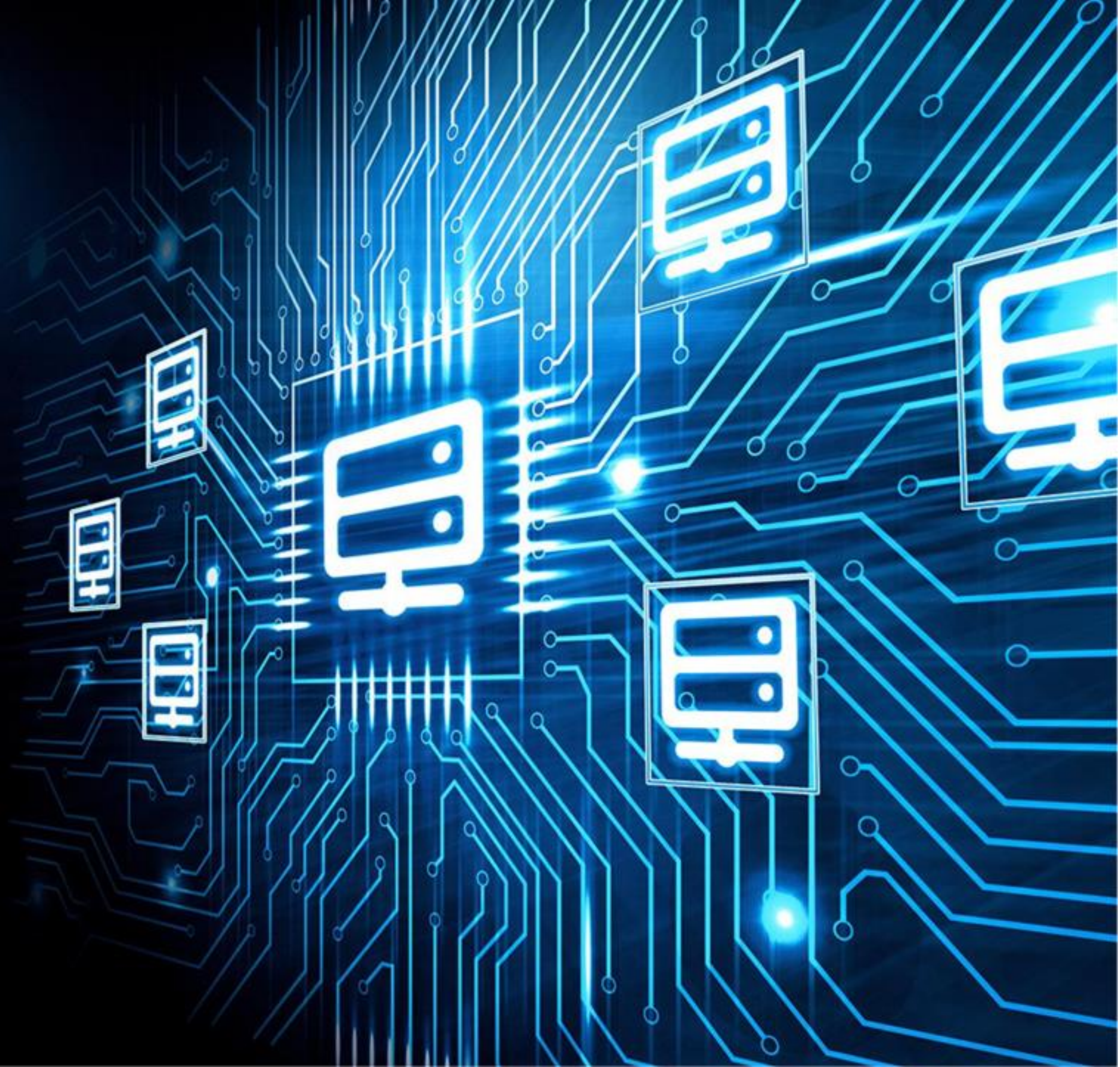
⇒ Vehicle Exit:

- When a vehicle exits, the system finds the corresponding parking session.
- It updates the exittime and marks the space as free in the Places table.
- Cost is calculated based on the time spent in the parking spot.

Future Enhancements and Additions

- Mobile Application Integration
- Automatic Payment System
- Host the database in the cloud (Azure, AWS)
- Use historical parking data to:
  - Predict when specific spots will be free
  - Optimize space utilization during peak hours

Forgetten password ?



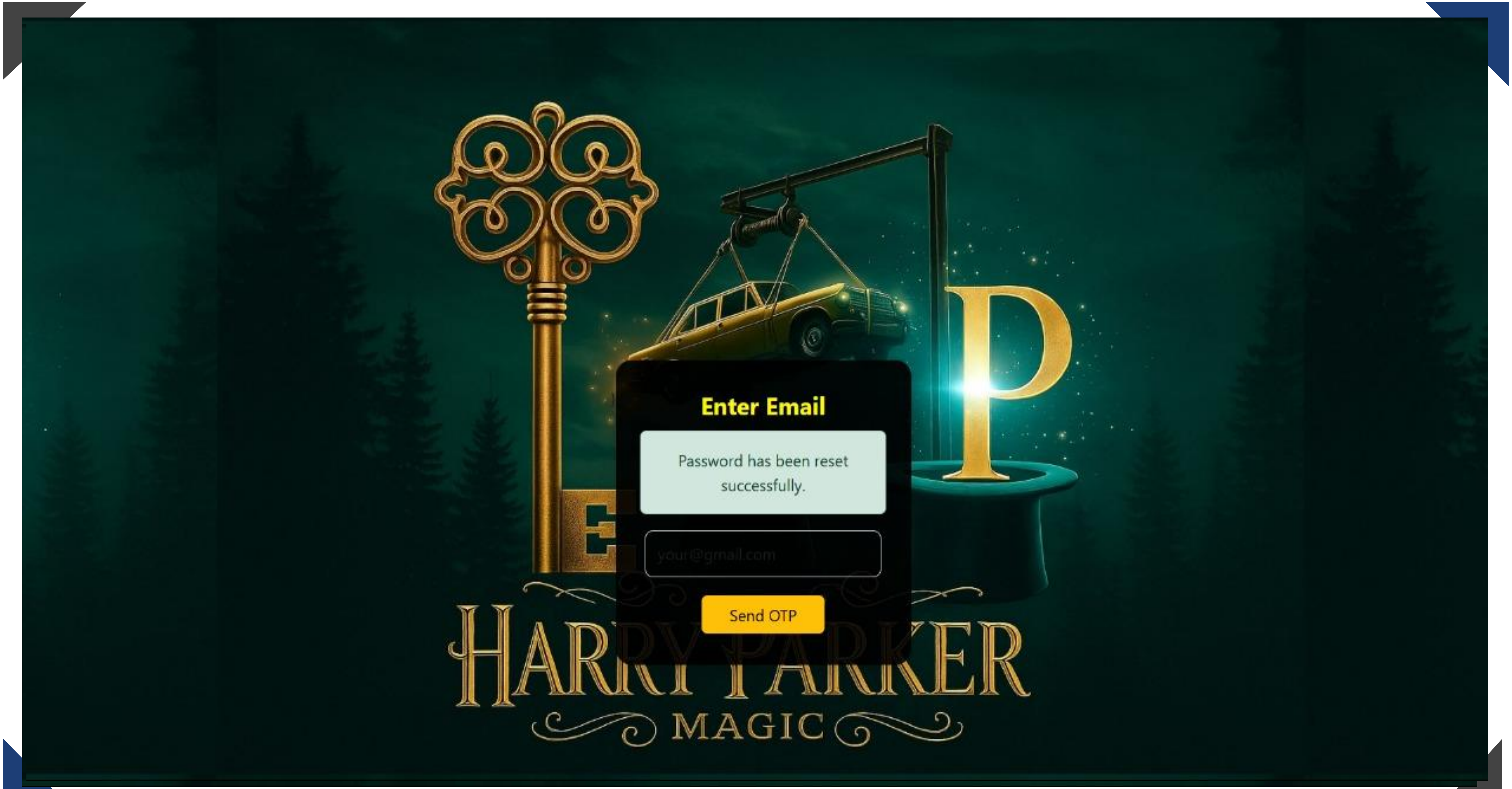
Objective

- Provide users a simple and secure way to reset their forgotten password
- Ensure the system is functional within a local network
- Use OTP (One Time Password) verification via email

How It Works

- User opens the site on 192.168.0.19:5281
- Enters registered email
- System generates a 6-digit OTP
- OTP is sent via email, valid for 3 minutes
- User enters OTP + new password
- Password updated in database

Local Hosting IP:192.168.0.19:5281



Enter Email

Password has been reset successfully.

Security

OTP is auto-expired after 3 minutes

SQL Injection prevention

OTP is stored securely

No password stored in plain text

Future Challenges

Email Deliverability Issues

- Emails may be marked as spam or blocked by providers.
- Free SMTP services may have daily limits.

Solution → Use a trusted mail service (e.g., SendGrid, Mailgun).

Localhost Limitations

- Currently runs only in local network (192.168.0.19).
- Not accessible to users outside the network.

Solution → Deploy online (e.g., on Azure, Vercel, or VPS) for real-world use.

Password Reset Loops → Handling users who request multiple OTPs in a short time.

Solution → Add cooldown timers or CAPTCHA to prevent abuse.