PARKING MANAGEMENT SYSTEM
A MINI-PROJECT REPORT

Submitted by

SHARVINKUMAR R   **220701507**

JAYASURIYAA K S   **220701332**

In partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

CHENNAI-602105

JUNE-2024

BONAFIDE CERTIFICATE

Certified that this project **"PARKING MANAGEMENT SYSTEM"** is the bonafide work of **"SHARVINKUMAR R(220701507), JAYASURIYAA K S(220701332)"** who carried out the project work under my supervision.

Submitted for the Practical examination to be Held on _____

SIGNATURE

Mrs. K. Mahesmeena
Assistant professor
Computer Science and Engineering,
Rajalakshmi Engineering College(Autonomous),
Thandalam, Chennai-602105.

INTERNAL EXAMINER   EXTERNAL EXAMINER

ABSTRACT

The project aims to develop an automatic parking space allocation system using sensor technology. Traditional parking systems often lead to inefficiencies and frustration due to manual allocation and lack of real-time information on available spaces. By implementing sensors to detect vacant parking spots and an automated allocation algorithm, this project seeks to optimize parking space utilization, reduce congestion, and enhance the overall parking experience for users. In addition to addressing immediate concerns such as congestion and user frustration, the proposed system holds the potential to revolutionize urban planning strategies.

The aforementioned project represents a significant stride in advancing the efficiency and competitiveness of parking management systems. By meticulously allocating parking spaces through sensor technology and automated algorithms, the project endeavors to elevate service standards and responsiveness. With a customer-centric approach, the ultimate aim is to bolster the entire urban parking ecosystem, empowering local agencies to surpass larger-scale platforms in delivering superior parking solutions.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1   INTRODUCTION :

This project introduces an automatic parking allocation system using sensors to address urban parking challenges. Traditional methods are inefficient due to manual allocation and lack of real-time data. By integrating sensors and automated algorithms, it aims to optimize space usage and more.

1.2   SCOPE OF THE WORK :

The scope of this project includes developing a sensor-based system for automatic parking space allocation, featuring real-time data collection and automated algorithms. It will also create a user-friendly interface and integrate payment processing. The goal is to improve parking efficiency and user experience in urban areas.

.

1.3   PROBLEM STATEMENT :

Traditional parking systems lead to inefficiencies and frustration due to manual allocation and lack of real-time information. These issues result in wasted time, increased congestion, and a poor user experience. Addressing these challenges requires an automated solution to optimize space utilization and streamline the parking process.

1.4   IMPLEMENTATION :

Our Python-based parking management system leverages MySQL for robust data storage and efficient connectivity. This project employs a web-based interface to streamline the allocation and monitoring of parking spaces. Users can effortlessly check availability, reserve spots, and navigate to their designated spaces through the system.

The Python application establishes a reliable connection to the MySQL database in the background, ensuring secure and effective storage of parking data such as spot availability, user reservations, and payment details. This implementation enables parking facility operators to manage spaces efficiently and lays the foundation for future enhancements, such as advanced analytics, real-time occupancy tracking, and dynamic pricing models.

Using a web-based interface, our Python-driven parking management system, which integrates MySQL, optimizes the parking process. By securely storing data in the MySQL database, the system offers an intuitive solution for users and operators. Its scalable architecture supports future expansions, including real-time analytics, enhanced user notifications, and automated billing systems, in addition to efficient parking management.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1   HARDWARE SPECIFICATIONS :

- An Arduino leonard
- Ultrasonic Sensor(HC-SR04)
- RGB LED
- Breadboard
- Jumper wires

PRE-REQUISITIES:

- Python, Mysql

2.2   SOFTWARE SPECIFICATIONS :

PROGRAMMING LANGUAGES   :   Python

OPERATING SYSTEM   :   MICROSOFT WINDOWS 11

SOFTWARES   :

- Python IDLE
- My Sql

CHAPTER 3

PARKING MANAGEMENT SYSTEM

SOURCE CODE

STEP1:
Create database db;

STEP2:
Use db;

STEP3:
CREATE TABLE vehicle VALUES(vehicle NO int not null PRIMARY KEY ,
Entrytiming time NOT NULL,Exittingtime time NOT NULL ,Cost int NOT
NULL,parkingslotNO int NOT NULL);

STEP4:
Insert into vehicle value (11,'03:55:00','08:40:07',500,121,11);

STEP5:
SELECT*FROM vehicle;

STEP6:
Insert into vehicle value (12,'05:50:00','10:50:07',500,122,12);

STEP7:
SELECT*FROM vehicle;

STEP8:
Show the database

STEP9:
Use db;

STEP10:
Show the table

STEP11:
SELECT*FROM vehicle;

STEP12:
Alter table vehicle add totaltiming int not null;

STEP13:
SELECT*FROM vehicle;

2.USING PYTHON :

```
signup_window = tk.Tk()
signup_window.title("Signup Form")
signup_window.geometry('1000x1000')
signup_window.config(bg="black")


# background_image = PhotoImage(file="pngimg.com - deadpool_PNG38.jpeg")
# Change "your_image.png" to the path of your image

bg_image = Image.open("rec.jpg")
bg_photo = ImageTk.PhotoImage(bg_image)
lable=tk.Label(signup_window,image=bg_photo)
lable.place(relx=1,rely=1)

canvas = tk.Canvas(signup_window, width=1000, height=1000)
canvas.pack(fill="both", expand=True)
canvas.create_image(0, 0, image=bg_photo)
```

```
form_frame = tk.Frame(canvas)
form_frame.place(relx=0.5, rely=0.5, anchor="center")

firstname_label = tk.Label(form_frame, text="First Name:")
firstname_label.grid(row=1, column=0, padx=10, pady=5, sticky="e")
firstname_entry = tk.Entry(form_frame)
firstname_entry.grid(row=1, column=1, padx=10, pady=5)

lastname_label = tk.Label(form_frame, text="Last Name:")
lastname_label.grid(row=2, column=0, padx=10, pady=5, sticky="e")
lastname_entry = tk.Entry(form_frame)
lastname_entry.grid(row=2, column=1, padx=10, pady=5)

email_label = tk.Label(form_frame, text="Email:")
email_label.grid(row=3, column=0, padx=10, pady=5, sticky="e")
email_entry = tk.Entry(form_frame)
email_entry.grid(row=3, column=1, padx=10, pady=5)

password_label = tk.Label(form_frame, text="Password:")
password_label.grid(row=4, column=0, padx=10, pady=5, sticky="e")
password_entry = tk.Entry(form_frame, show="*")
password_entry.grid(row=4, column=1, padx=10, pady=5)

signup_button = tk.Button(form_frame, text="Signup", command=register)
signup_button.grid(row=5, column=0, columnspan=2, pady=10)
signin_button = tk.Button(form_frame, text="Signin",
command=open_login_page)
signin_button.grid(row=6, column=0, columnspan=2, pady=10)
signup_window.mainloop()
```

Database-Mysql

```
Your MySQL connection id is 11
Server version: 8.0.37 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database db;
ERROR 1007 (HY000): Can't create database 'db'; database exists
mysql> create database ms;
Query OK, 1 row affected (0.02 sec)

mysql> create database db;
ERROR 1007 (HY000): Can't create database 'db'; database exists
mysql> CREATE TABLE vichle(vichleNO int NULL PRIMARY KEY ,Entertiming varchar(50) NOT NULL,Exittime varchar(50) NOT NULL
,Cost varchar(50) NOT NULL,prakingslotno varchar(50) NOT NULL);
ERROR 1046 (3D000): No database selected
mysql> select*from vichle;
ERROR 1046 (3D000): No database selected
mysql> create database db1;
Query OK, 1 row affected (0.02 sec)

mysql> CREATE TABLE vichle(vichleNO int NULL PRIMARY KEY ,Entertiming varchar(50) NOT NULL,Exittime varchar(50) NOT NULL
,Cost varchar(50) NOT NULL,prakingslotno varchar(50) NOT NULL);
ERROR 1046 (3D000): No database selected
mysql> -|
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> Use db
Database changed
mysql> Show databases;
+--------------------+
| Database           |
+--------------------+
| db                 |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)

mysql> show tables;
+--------------+
| Tables_in_db |
+--------------+
| vehicle      |
+--------------+
1 row in set (0.00 sec)
```

```
5 rows in set (0.01 sec)

mysql> show tables;
+---------------+
| Tables_in_db  |
+---------------+
| vehicle       |
+---------------+
1 row in set (0.00 sec)

mysql> SELECT*FROM vehicle;
+-----------+-------------+----------+------+---------+--------+
| VehcileNO | Entrytiming | Exitting | Cost | parking | Sensor |
+-----------+-------------+----------+------+---------+--------+
|         1 | 09:06:58    | 10:10:00 |  100 |     111 |      1 |
|         2 | 02:10:54    | 06:55:30 |  500 |     112 |      2 |
|         3 | 10:55:00    | 02:25:17 |  530 |     113 |      3 |
|         4 | 11:55:10    | 03:25:16 |  600 |     114 |      4 |
|         5 | 11:50:55    | 05:45:50 |  700 |     115 |      5 |
|         6 | 05:55:59    | 12:00:00 |  700 |     116 |      6 |
|         7 | 02:35:55    | 03:00:55 |  150 |     117 |      7 |
|         8 | 11:23:55    | 05:00:44 |  705 |     118 |      8 |
|         9 | 11:23:55    | 05:00:44 |  705 |     119 |      9 |
|        10 | 12:03:50    | 05:10:40 |  650 |     120 |     10 |
+-----------+-------------+----------+------+---------+--------+
10 rows in set (0.00 sec)
```

```
mysql> insert into vehicle value (11,'03:55:00','08:40:07',500,121,11);
Query OK, 1 row affected (0.01 sec)

mysql> insert into vehicle value (12,'05:50:00','10:50:07',500,122,12);
Query OK, 1 row affected (0.01 sec)

mysql> insert into vehicle value (13,'06:50:00','11:50:07',500,123,13);
Query OK, 1 row affected (0.01 sec)

mysql> SELECCT*FROM vehicle;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'S
ELECCT*FROM vehicle' at line 1
mysql> SELECT*FROM vehicle;
+-----------+-------------+----------+------+---------+--------+
| VehcileNO | Entrytiming | Exitting | Cost | parking | Sensor |
+-----------+-------------+----------+------+---------+--------+
|         1 | 09:06:58    | 10:10:00 |  100 |     111 |      1 |
|         2 | 02:10:54    | 06:55:30 |  500 |     112 |      2 |
|         3 | 10:55:00    | 02:25:17 |  530 |     113 |      3 |
|         4 | 11:55:10    | 03:25:16 |  600 |     114 |      4 |
|         5 | 11:50:55    | 05:45:50 |  700 |     115 |      5 |
|         6 | 05:55:59    | 12:00:00 |  700 |     116 |      6 |
|         7 | 02:35:55    | 03:00:55 |  150 |     117 |      7 |
|         8 | 11:23:55    | 05:00:44 |  705 |     118 |      8 |
|         9 | 11:23:55    | 05:00:44 |  705 |     119 |      9 |
|        10 | 12:03:50    | 05:10:40 |  650 |     120 |     10 |
|        11 | 03:55:00    | 08:40:07 |  500 |     121 |     11 |
|        12 | 05:50:00    | 10:50:07 |  500 |     122 |     12 |
|        13 | 06:50:00    | 11:50:07 |  500 |     123 |     13 |
+-----------+-------------+----------+------+---------+--------+
13 rows in set (0.01 sec)

mysql>
```

```
mysql> SELECCT*FROM vehicle;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'S
ELECCT*FROM vehicle' at line 1
mysql> SELECT*FROM vehicle;
+-----------+------------+----------+------+---------+--------+
| VehcileNO | Entrytiming | Exitting | Cost | parking | Sensor |
+-----------+------------+----------+------+---------+--------+
|         1 | 09:06:58    | 10:10:00 |  100 |     111 |      1 |
|         2 | 02:10:54    | 06:55:30 |  500 |     112 |      2 |
|         3 | 10:55:00    | 02:25:17 |  530 |     113 |      3 |
|         4 | 11:55:10    | 03:25:16 |  600 |     114 |      4 |
|         5 | 11:50:55    | 05:45:50 |  700 |     115 |      5 |
|         6 | 05:55:59    | 12:00:00 |  700 |     116 |      6 |
|         7 | 02:35:55    | 03:00:55 |  150 |     117 |      7 |
|         8 | 11:23:55    | 05:00:44 |  705 |     118 |      8 |
|         9 | 11:23:55    | 05:00:44 |  705 |     119 |      9 |
|        10 | 12:03:50    | 05:10:40 |  650 |     120 |     10 |
|        11 | 03:55:00    | 08:40:07 |  500 |     121 |     11 |
|        12 | 05:50:00    | 10:50:07 |  500 |     122 |     12 |
|        13 | 06:50:00    | 11:50:07 |  500 |     123 |     13 |
+-----------+------------+----------+------+---------+--------+
13 rows in set (0.01 sec)

mysql> Alter table vehicle add totaltiming int not null;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

```
mysql> Alter table vehicle add totaltiming int not null;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT*FROM vehicle;
+-----------+------------+----------+------+---------+--------+-------------+
| VehcileNO | Entrytiming | Exitting | Cost | parking | Sensor | totaltiming |
+-----------+------------+----------+------+---------+--------+-------------+
|         1 | 09:06:58    | 10:10:00 |  100 |     111 |      1 |           0 |
|         2 | 02:10:54    | 06:55:30 |  500 |     112 |      2 |           0 |
|         3 | 10:55:00    | 02:25:17 |  530 |     113 |      3 |           0 |
|         4 | 11:55:10    | 03:25:16 |  600 |     114 |      4 |           0 |
|         5 | 11:50:55    | 05:45:50 |  700 |     115 |      5 |           0 |
|         6 | 05:55:59    | 12:00:00 |  700 |     116 |      6 |           0 |
|         7 | 02:35:55    | 03:00:55 |  150 |     117 |      7 |           0 |
|         8 | 11:23:55    | 05:00:44 |  705 |     118 |      8 |           0 |
|         9 | 11:23:55    | 05:00:44 |  705 |     119 |      9 |           0 |
|        10 | 12:03:50    | 05:10:40 |  650 |     120 |     10 |           0 |
|        11 | 03:55:00    | 08:40:07 |  500 |     121 |     11 |           0 |
|        12 | 05:50:00    | 10:50:07 |  500 |     122 |     12 |           0 |
|        13 | 06:50:00    | 11:50:07 |  500 |     123 |     13 |           0 |
+-----------+------------+----------+------+---------+--------+-------------+
13 rows in set (0.00 sec)
```

CHAPTER 4

SNAP SHOTS

1.OUTPUT:IN PYTHON

2.OUTPUT:In Java Swings

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

In the future, our parking management system could incorporate IoT devices and AI for even more precise space allocation and predictive maintenance. The addition of advanced IoT sensors can enable real-time monitoring of parking conditions, allowing for immediate detection of unauthorized parking and improved security. Developing a mobile app would facilitate easier parking reservations, navigation, and real-time notifications for users, enhancing convenience and user engagement.

Integrating machine learning algorithms can further optimize space usage by predicting peak times and adjusting allocation strategies dynamically, improving the system's adaptability to fluctuating demands. Data analytics can offer valuable insights into user behavior, peak usage times, and revenue trends, enabling facility managers to make informed decisions on pricing, promotions, and infrastructure investments. Furthermore, incorporating features like automated payment processing and license plate recognition can streamline entry and exit processes, reducing wait times and enhancing overall efficiency.

The aim of these enhancements is to elevate the overall user experience, significantly boost operational efficiency in urban parking management, and pave the way for a smarter, more sustainable city infrastructure. These advancements will not only improve parking convenience but also contribute to reducing traffic congestion and environmental impact by optimizing resource utilization.

BIBILOGROPHY

This below websities helped us in gaining more knowledge on the subject and in completing the project.

1. https://www.w3schools.com/sql/

2. https://www.abc.net.au/news/2020-03-08/fast-fashionversus-op-shops/12034424

3. https://www.nacro.org.au/fast-facts/