

**A Project Report on**

**“Smart Parking”**

**Submitted by**

**19203A0003 - Kaushik Shigavan**

**19203A0029 - Harshad Raurale**

**19203A0040 - Shubham Modi**

**19203A0052- Priyanshu Worlikar**

**Under the guidance of**

**Er. Gauri Bobade**

**Head of the Department**

**Er. Vijay Patil**

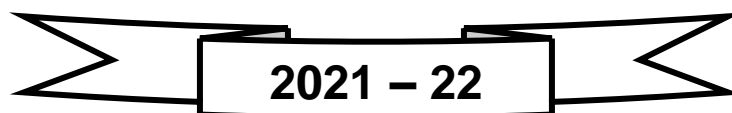
**Department of Computer Engineering**

**(NBA Accredited)**

**Vidyalankar Polytechnic**

**Wadala (E), Mumbai – 400 037**

**Maharashtra State Board of Technical Education, Mumbai**



## Institute Vision

To achieve excellence in imparting **Technical Education** so as to meet the Professional and **Societal needs**.

## Institute Mission

- Developing **technical skills** by imparting knowledge and providing hands on experience.
- Creating an environment that nurtures **ethics, leadership and team building**.
- Providing **industrial exposure** for minimizing the gap between academics & industry.

## Program Vision

To empower students with domain knowledge of Computer Engineering and interpersonal skills to cater to the industrial and societal needs.

## Program Mission

M1: Encouraging academic excellence and a passion for learning through the use of learner-oriented teaching methodologies.

M2: Providing an environment that inculcates ethics and effective soft-skills and focuses on the development of learners.

M3: Establishing and reinforcing a symbiotic institute-industry interface so that learners can gain exposure to real-life applications of Information Technology.

## Program Education Objectives

**PEO1:** To empower the students to apply their knowledge of various domains and cultivate a habit of life-long learning in them.

**PEO2:** To develop the students' overall personality and thereby encourage just social behaviour in the student.

## Program Outcomes

**Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

**Problem analysis:** Identify and analyse well-defined engineering problems using codified standard methods.

**Design/ development of solutions:** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

**Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.

**Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.

**Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

**Life-long learning:** Ability to analyse individual needs and engage in updating in the context of technological changes.

## Program Specific Outcomes

**PSO1:** Computer Software and Hardware Usage: Use state-of-the-art technologies for operation and application of computer software and hardware.

**PSO2:** Computer Engineering Maintenance: Maintain computer engineering related software and hardware systems.



# *Certificate*

This is to certify that the following students

19203A0003 - Kaushik Shigavan

19203A0029 - Harshad Raurale

19203A0040 - Shubham Modi

19203A0052- Priyanshu Worlikar

Have completed the Project on  
**Smart Parking System**

A partial fulfilment of the requirement of Third Year Diploma in  
Computer Engineering affiliated to Maharashtra State Board of  
Technical Education, Mumbai for the Academic Year 2021-22.

Internal Examiner

External Examiner

Project Guide

Head of Department

Principal

# **Acknowledgement**

**This project is a huge team effort. My team and I extend our deepest gratitude and thanks to the following people who have helped us to achieve our work.**

**Special thanks to our project guide Er. Gauri Bobade for guiding us and helping in time when we needed most. We got to learn many things from him and it was our pleasure to work with him. My team and I extend thanks to our faculties of our college whom we have approached for academic help with regards to our project. We also would like to thank our HOD Er. Vijay Patil and Principal Prof. Ashish Ukidve for their support and guidance.**

**(Kaushik Shigavan)**

**(Harshad Raurale)**

**(Shubham Modi)**

**(Priyanshu Worlikar)**

# **Abstract**

- Parking is a problem in many cities. Usually, it is not possible to build new parking lots due to insufficient available areas.
- We will present a miniature model of an automated car parking system that can regulate and manage the number of cars.
- Automated parking is a method of parking and exiting cars using sensing devices.
- The entering to or leaving from the parking lot is commanded by an Android based application.

## Table of Contents

<b>Chapter No.</b>	<b>Title</b>	<b>Pg. No</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2.</b>	<b>REVIEW OF LITERATURE</b>	<b>3</b>
<b>3.</b>	<b>PLAN OF WORK</b>	<b>10</b>
<b>4.</b>	<b>SYSTEM REQUIREMENTS</b>	<b>16</b>
<b>5.</b>	<b>DESIGNING</b>	<b>18</b>
<b>6.</b>	<b>MODULE DESCRIPTION</b>	<b>22</b>
<b>7.</b>	<b>TESTING</b>	<b>24</b>
<b>8.</b>	<b>ADVANTAGES &amp; DRAWBACKS</b>	<b>27</b>
<b>9.</b>	<b>APPLICATIONS</b>	<b>31</b>
<b>10.</b>	<b>SCREENSHOTS</b>	<b>33</b>
<b>11.</b>	<b>CONCLUSION</b>	<b>41</b>
<b>12.</b>	<b>BIBLIOGRAPHY</b>	<b>43</b>

## TABLE OF FIGURES

<b>SR.NO</b>	<b>Figures List</b>	<b>Page No.</b>
<b>1.</b>	<b>Fig 5.1 Use Case Diagram of the System</b>	<b>19</b>
<b>2.</b>	<b>Fig 5.2 Flow Chart of the System</b>	<b>20</b>
<b>3.</b>	<b>Fig 5.3 Entity Relationship Diagram of the System</b>	<b>21</b>



# **CHAPTER 1**

## **INTRODUCTION**

## INTRODUCTION:

- In the project Car Parking System, we have shown the concept of an automatic car parking system.
- As in the modern world everything is going automatic, we have built a system which will display the number of cars in the parking lot and park the car in a vertical manner, so that we can park more cars in less space.
- We will develop a system which is used to allocate parking space for cars and depending upon whether there is a capacity of cars to enter, it either Free space is there or not.
- It is also possible to free space when any car enters in the parking lot or when a car exits from the parking lot.
- If the availability of Parking space is confirmed, the user commands the parking system to park the car in the designated slot.

## PROBLEM STATEMENT:

### 1. Why is there a need of the selected topic?

In the last one decade, there is a rapid increase in the number of Vehicles. Approximately there is at least one Car in every household. As a result of this there is more cars and less parking space.

### 2. How is the topic better from existing product / technology?

It is less time consuming as the work is done through mobile application. In most of the vertical parking system it is not easy to access cars that are parked at the top. In our Parking System we will overcome this problem.

### 3. Who will benefit from the project topic selected?

It will be beneficial for Malls, Society and Residential area.

# **CHAPTER 2**

## **REVIEW OF LITERATURE**

## REVIEW:

1. An automated Car parking system is a mechanical system designed to minimize the area and/or volume required for parking cars.
2. Like a multi-story parking garage, an APS provides parking for cars on multiple levels stacked vertically to maximize the number of parking spaces while minimizing land usage.
3. The APS, however, utilizes a mechanical system to transport cars to and from parking spaces in order to eliminate much of the space wasted in a multi-story parking garage.
4. While a multi-story parking garage is similar to multiple parking lots stacked vertically, an APS is more similar to an automated storage and retrieval system for cars.

## DEVELOPING TOOLS:

- Apache Server
- My-SQL Database
- PHP
- Xampp
- phpMyAdmin

### 2.1. Apache Server:

The Apache HTTP Server, colloquially called Apache, is a free and open-source cross platform web server software, released under the terms of Apache Licence 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from authentication schemes to supporting server-side programming languages such as Perl, Python, Tel and PHP. Popular authentication modules include `mod_access`, `mod_auth`, `mod_digest` and `mod_auth_digest`, the successor to `mod_digest`. A sample of other features include Secure Sockets Layer and Transport Layer Security support (`mod_ssl`), a proxy module (`mod_proxy`), a URL rewriting module (`mod_rewrite`), custom log files (`mod_log_config`), and filtering support (`mod_include` and `mod_ext_filter`).



## 2.2 My SQL Database

**MySQL** is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.



For proprietary use, several paid editions are available, and offer additional functionality. MySQL is written in C and C++. Its SQL parser is written in yucca, but it uses a home-brewed lexical analyser. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, eComStation, i5/OS, IRI X, Linux, macOS, Microsoft Windows, Net BSD, Novell NetWare, OpenBSD, Open Solaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO Open Server, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists

### Why We Used MY SQL: -

- Cross-platform support.
- Updatable views.
- A set of SQL Mode options to control runtime behaviour, including strict

mode to better adhere to SQL standards.

- Embedded database library.
- Unicode support.
- Partitioned tables with pruning of partitions in optimizer.
- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.
- Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

## 2.3 PHP:

Hypertext Pre-processor (or simply PHP) is a server-side scripting language designed for web development. It was originally created by Rasmus Lerdorf in 1994 the PHP reference implementation is now produced by The PHP Group.<sup>[5]</sup> PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Pre-processor.

PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more. There are three main areas where PHP scripts are used. Server-side scripting.



Why to use PHP:

One of the important advantages of PHP is that it is Open Source. Therefore, PHP is readily available and is entirely free. In contrast to other scripting languages used for web development which requires the user to pay for the support files, PHP is open to everyone, anytime and anywhere.

Best Language for Server-Side Development. Python and PHP are two of the most popular high-level programming languages. PHP is traditionally used as a server-side scripting language, while Python is valued for its dynamics, availability, and simplicity.

## 2.4. Xampp:

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.[3][4] Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such as WordPress and Joomla! Can also be installed with similar ease using Bitnami.



# XAMPP

XAMPP is regularly updated to the latest releases of Apache, MariaDB, PHP and Perl. It also includes OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress and more. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a full and a standard version (Smaller version).

## 2.5 phpMyAdmin:

phpMyAdmin is a free and open-source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services.

### Features of phpMyAdmin:

- Web interface
- MySQL and MariaDB database management
- Import data from CSV and SQL
- Export data to various formats: CSV, SQL, XML, PDF (via the TCPDF library), ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel, LaTeX and others

- Administering multiple servers



## 2.6 Android Studio:

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps.



# Android Studio



## 2.7 Arduino UNO:

The Arduino UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.



## 2.8 NodeMCU:

NodeMCU is an open-source platform based on ESP8266 which can connect objects and let data transfer using the Wi-Fi protocol. In addition, by providing some of the most important features of microcontrollers such as GPIO, PWM, ADC, and etc, it can solve many of the project's needs alone.



# **CHAPTER 3**

## **PLAN OF WORK**

### **3.1 APPLICATION/SOFTWARE DEVELOPMENT LIFE CYCLE:**

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

SDLC usually begins with determining customer business needs, which is followed by implementation and testing. The cycle ends when all requirements have been fulfilled.

The System Development Life Cycle (SDLC) has five phases: The Planning phase, the Analysis phase, the Design phase, the Implementation phase and the Maintenance Planning of the SDLC.

#### **1. Requirement Analysis:**

Software Development Life Cycle begins with Requirement Analysis phase, where the stakeholders discuss the requirements of the software that needs to be developed to achieve a goal. The aim of the requirement analysis phase is to capture the detail of each requirement and to make sure everyone understands the scope of the work and how each requirement is going to be fulfilled.

#### **2. Design:**

The next stage of Software Development Life Cycle is the Design phase. During the design phase, developers and technical architects start the high-level design of the software and system to be able to deliver each requirement.

#### **3. Implementation:**

After the requirements and design activity is completed, the next phase of the Software Development Life Cycle is the implementation or development of the software. In this phase, developers start coding according to the requirements and the design discussed in previous phases.

#### 4. Testing:

Testing is the last phase of the Software Development Life Cycle before the software is delivered to customers. During testing, experienced testers start to test the system against the requirements. The testers aim to find defects within the system as well as verifying whether the application behaves as expected and according to what was documented in the requirements analysis phase.

#### 5. Deployment and Maintenance:

Once the software has been fully tested and no high priority issues remain in the software, it is time to deploy to production where customers can use the system.

Once a version of the software is released to production, there is usually a maintenance team that look after any post-production issues.

#### 6. Process Model

There are different types of models used by a software team to do their work systematic that is step by step. The original process models have certainly given a guidance or roadmap for the whole software development process or software engineering.

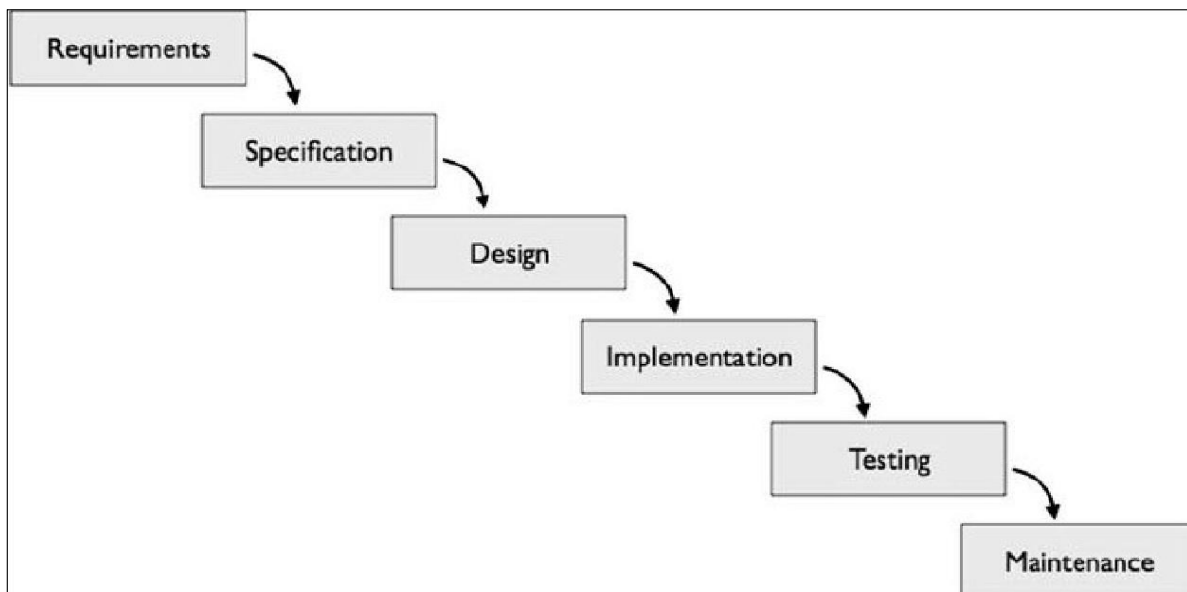
The process models are properly structured so that at least structure of the process can be understood.

- The Different process models are:
  - **Waterfall Model**
  - Incremental Model
  - RAD Model
  - Prototype Model
  - Spiral Model

We have chosen the **Waterfall Model** for development of our project.

#### **WATERFALL MODEL**

The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. The approach is typical for certain areas of engineering design. In software development, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, deployment and maintenance.



**Fig 3.2 WATERFALL MODEL**

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

### 3.2 HARDWARE DEVELOPMENT:

Sr. No.	Details of Activity	Planned Start Date	Planned Finish Date	Name of Responsible Team Members
1	Selecting Component's	28-02-2022	05-03-2022	Kaushik Shigavan Harshad Raurale Shubham Modi Priyanshu Worlikar
2	Assembly Arduino uno, IR sensor and NodeMCU	07-03-2022	10-03-2022	Kaushik Shigavan Harshad Raurale Priyanshu Worlikar
3	Coding Arduino and NodeMCU	12-03-2022	20-03-2022	Priyanshu Worlikar
4	Assembly Stepper Motor with its Driver and Adapter	25-03-2022	27-03-2022	Shubham Modi Priyanshu Worlikar
5	Embedding Stepper Motor in Arduino and NodeMCU	01-04-2022	03-04-2022	Harshad Raurale Priyanshu Worlikar

# **CHAPTER 4**

## **SYSTEM REQUIREMENTS**



#### 4.1 HARDWARE REQUIREMENTS:

Name of the component	Specification
Processor	Intel i5(3.6GHz)
RAM	8GB
Arduino uno	A microcontroller board based on the <b>ATmega328P</b> . It has 14 digital input/output pins
ESP8266	The ESP8266 is a low-cost Wi-Fi microchip, with built-in <b>TCP/IP</b> networking software, and microcontroller capability
Infrared sensor (IR sensor)	A radiation-sensitive optoelectronic component with a spectral sensitivity in the Infrared Wavelength <b>780 nm</b>
Stepper Motor	Brushless DC electric motor that divides a full rotation into a number of equal steps, support on 12 voltage power supply.

#### 4.2 SOFTWARE REQUIREMENTS:

Name of the component	Specification
Operating System	Windows 10
Language	PHP
Database	MYSQL, APACHE
Webserver	XAMPP
Browser	Google Chrome
Android Studio	Bumble Bee Version
Arduino IDE	Arduino Code Editor

# **CHAPTER 5**

# **DESIGNING**

## 5.1 USE CASE DIAGRAM:

A use-case diagram is usually referred to as behaviour diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

- Elements of Use case Diagram:

- Actor: It is an external entity that interacts with the system and it is denoted by a stick man
- Use case: It specifies a distinct function of a system and it is denoted by an oval
- System Boundary Box: It is used to collect all the use cases of a system and it is denoted by a rectangle.
- Association: It is used to denote the functionality performed by an actor and it is denoted by directed line.

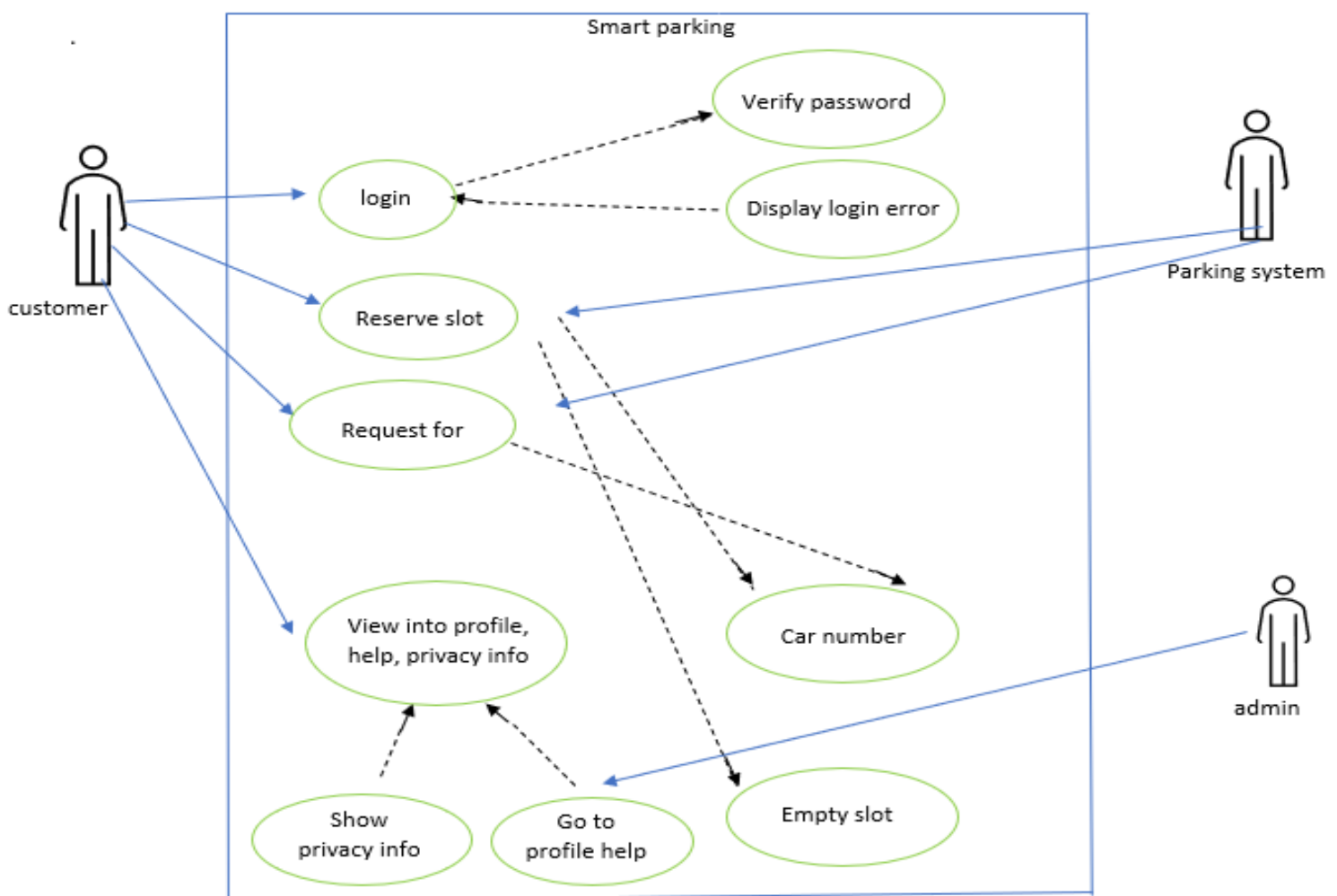


Fig.5.1 Use Case Diagram.

## 5.2. FLOWCHART:

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. A flowchart is described as "cross-functional" when the page is divided into different swim lanes describing the control of different organizational units. A symbol appearing in a particular "lane" is within the control of that organizational unit. This technique allows the author to locate the responsibility for performing an action or making a decision correctly, showing the responsibility of each organizational unit for different parts of a single process. Flowcharts are used in analysing, designing, documenting or managing a process or program in various fields.

The flowchart of our project is to select a category from subject line.

We have constructed our flowcharts in two sections (describing the two features) so that it becomes easy for us to understand the flow of our project.

If the subject line does not contain a recognized word, then the user will select the other category.

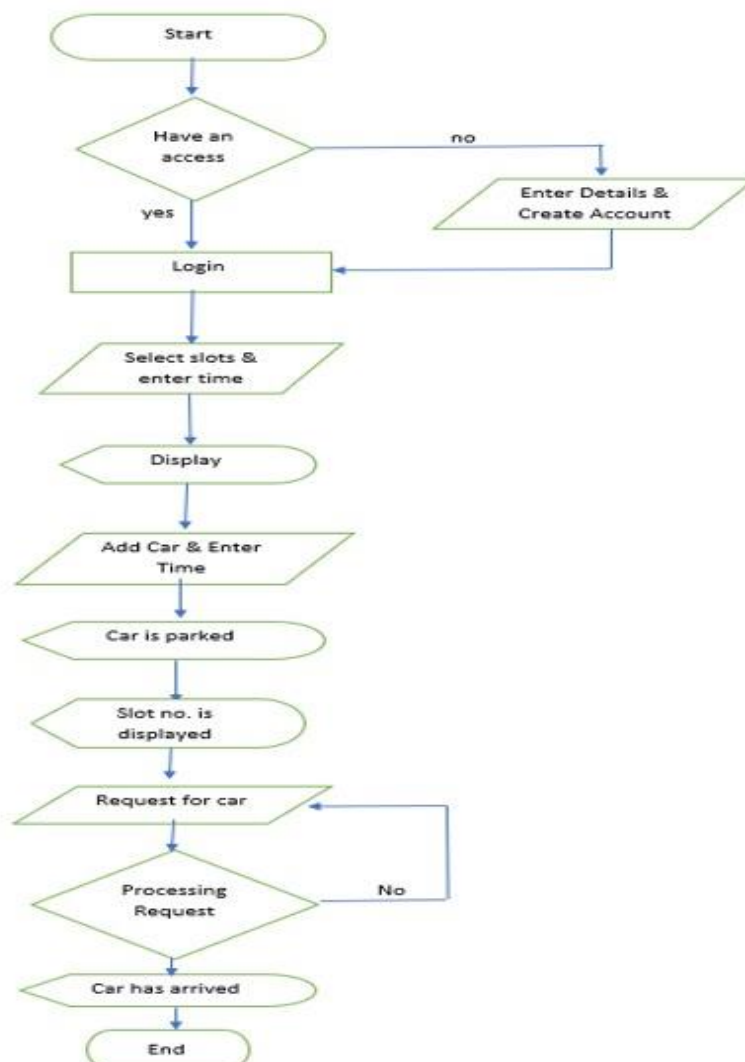


Fig.5.2. Flowchart Diagram.

### 5.3. E-R DIAGRAM:

An Entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

- The elements of an ERD are:
  - Entities are the "things" for which we want to store information. An entity is a person, place, thing or event.
  - Attributes are the data we want to collect for an entity.
  - Relationships describe the relations between the entities.
- Steps involved in creating an ERD include:
  - Identifying and defining the entities
  - Determining all interactions between the entities
  - Analysing the nature of interactions/determining the cardinality of the relationships
  - Creating the ERD.



Fig.5.3 E-R Diagram.

# **CHAPTER 6**

## **MODULE DESCRIPTION**

## **6. MODULE DESCRIPTION:**

### **6.1 Registration**

### **6.2 Login**

### **6.3 View Available Slot**

### **6.4 Book Slot**

### **6.5 Check Out Booked Slot**

### **6.6 Logout**

#### **6.1 Registration**

USER will first register him/her self on our application. Registration process involve the details of user i.e., Name, Contact number, Email address, Car number plate and Password. Each and all user's details are saved in database before actually signing in into the software.

#### **6.2 Login**

USER will then Login in our application by using same Email address and Password that he/she used while registration process.

#### **6.3 View Available Slot**

USER will able to view available slot and which are reserved slot.

#### **6.4 Book Slot**

By Checking the availability of slot's USER will able to book any specific slot by our application.

#### **6.5 Check Out Booked Slot**

After the certain time if USER wants to leave, he/she can check out from that particular slot by using application and that slot will become available for next User as soon as car leaves.

#### **6.6 Logout**

USER can Logout from Application after checking availability/booking the slot.

# **CHAPTER 7**

## **TESTING**



## 7. TESTING:

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

### 7.1 TEST CASES

SR.NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
1.	Sign Up	Enter mail id, password	View home page	View home page	Pass
2.	Register	Enter all the required details.	If all the details are filled properly add the data successfully.	User is successfully added.	Pass
4.	Parking	Click on the parking button.	It should show the available slots. If the slot is booked it should show the time it will be free. If the slot is empty, it should allow the user to book the slot.	It shows the available slots and allows the user to book the slot if the slot is empty.	Pass
5.	Slots	Click on the slot button.	If the slot is empty the slot should be booked for the user, else if it is booked or full it should display a message that is booked/full.	It books the slot for the user if it is empty, and if it is full it shows the message displaying it is full.	Pass
6.	Cancel Parking	Click on the button to cancel the booked slot.	It should cancel the slot if the slot is booked.	It cancels the slot that is booked by the user.	Pass
7.	Status	On clicking on the status button.	It should show the status of the slot booked by the user.	It shows the status of the slot booked by the user.	Pass

8.	Park the car.	On clicking park.	If the slot booked by the user is empty and, if the user wants to park his car in the slot, he can click on the park button and the car should get parked.	It parks the user car on the click of the button.	Pass
9.	Get the car.	On clicking get the car button.	If the user has parked his car and he want to retrieve his/her car they can click on the get the car button and system should get the users car from the parking.	On click it gets the users car from the parking.	Pass
10	Logout.	On clicking Logout button.	When the user clicks on the logout button, he should be logged out from the app.	When the user clicks the button, he is logged out from the app.	Pass

# **CHAPTER 8**

## **ADVANTAGES AND DRAWBACKS**

## 8.1 ADVANTAGES:

- There is a greater sense of security due to the fact that patrons do not actually walk to and from their own space.
- It is highly feasible for extremely small sites that are unable to accommodate a conventional ramped parking structure.
- There is high parking efficiency.
- There is no need for driving while looking for an available space.
- Emissions are greatly brought down and reduced.
- The patrons wait for their car in a highly controlled environment.
- There are less chances for vehicle vandalism.
- There is a minimal staff requirement if it is used by known parkers.
- It is possible that the retrieval time is lower than the combined driving/parking/walking time in conventional ramped parking structures.  
There is an easier facade integration since there are no ramping floors or openings in exterior walls.

## 8.2 DISADVANTAGES:

- There is a greater construction cost per space (but this may be offset by the chance for lesser land costs per space and the system manufacturers say that the operating and maintenance cost will be lower as compared to a conventional ramped parking structure).
- Use of redundant systems will result in a greater cost.
- It may be a bit confusing for unfamiliar users.
- It is not recommended for high peak hour volume facilities.
- There may be a fear of breakdown (How do I get my car out?).
- There is an uncertain building department review and approval process.
- It requires a maintenance contract with the supplier.

# **CHAPTER 9**

## **APPLICATIONS**

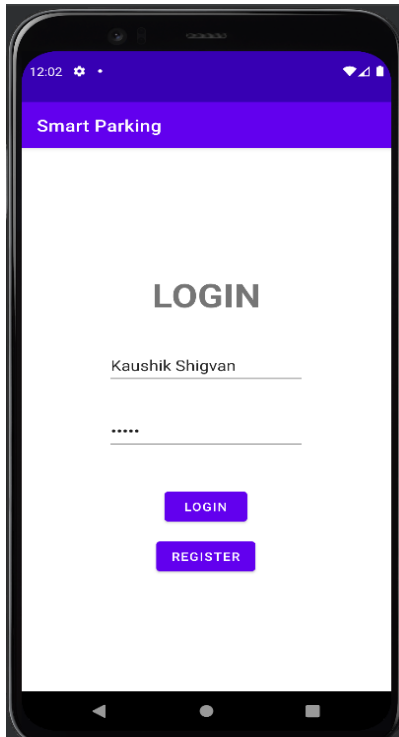
## **9. APPLICATIONS:**

- This Project can be used in Malls, Multiplex.
- It can also be used in Society and Residential areas.
- It can be used in Offices, Colleges and Hospital.
- Help to optimize parking space usage.
- Real time data and application.

# **CHAPTER 10**

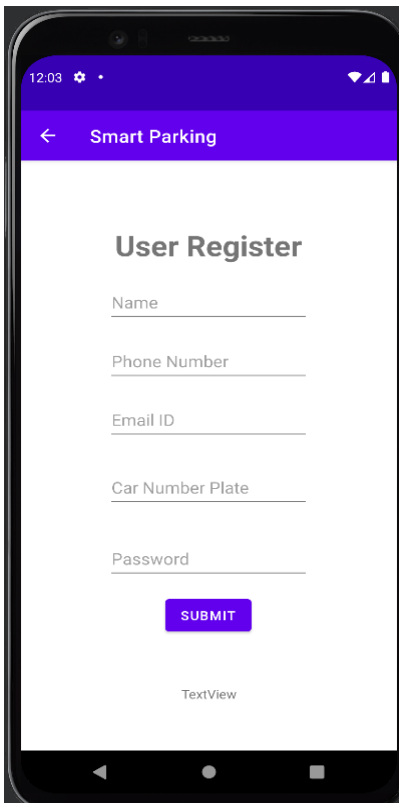
# **SCREENSHOTS**

## 10.1 LOGIN:



The image shows a mobile application interface for the 'Smart Parking' app. At the top, there is a purple header bar with the text 'Smart Parking'. Below the header, the word 'LOGIN' is displayed in a large, bold, black font. Underneath 'LOGIN', there are two input fields: the first one contains the text 'Kaushik Shigvan' and the second one contains six dots, indicating a password field. Below these fields, there are two purple buttons: the top one is labeled 'LOGIN' and the bottom one is labeled 'REGISTER'. The entire interface is set against a white background.

## 10.2 REGISTER:



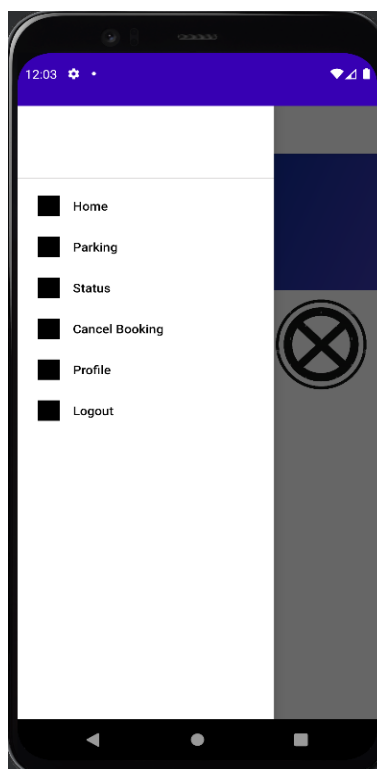
The image shows a mobile application interface for the 'Smart Parking' app, specifically the 'User Register' screen. At the top, there is a purple header bar with a back arrow icon and the text 'Smart Parking'. Below the header, the text 'User Register' is displayed in a bold, black font. Underneath, there are five input fields: 'Name', 'Phone Number', 'Email ID', 'Car Number Plate', and 'Password'. Below these fields, there is a purple button labeled 'SUBMIT'. At the very bottom of the screen, there is a 'TextView' label. The entire interface is set against a white background.

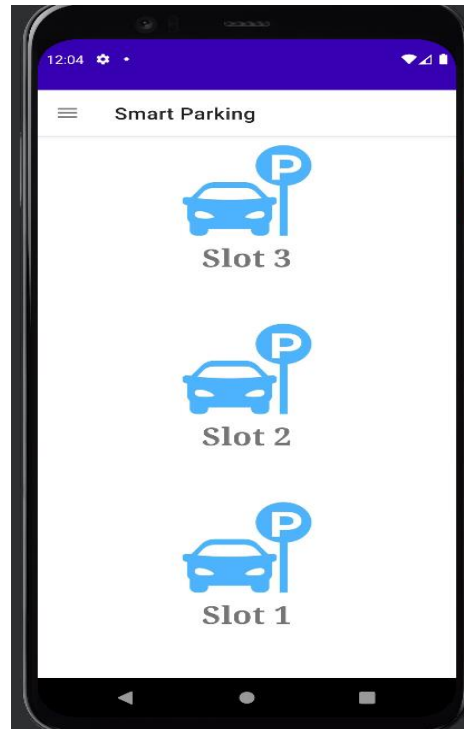
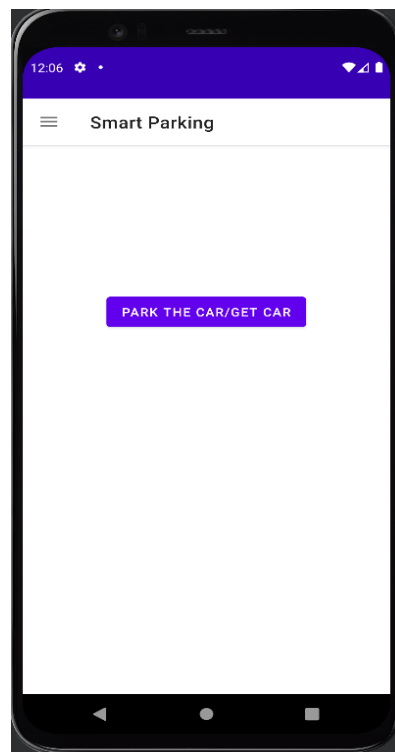


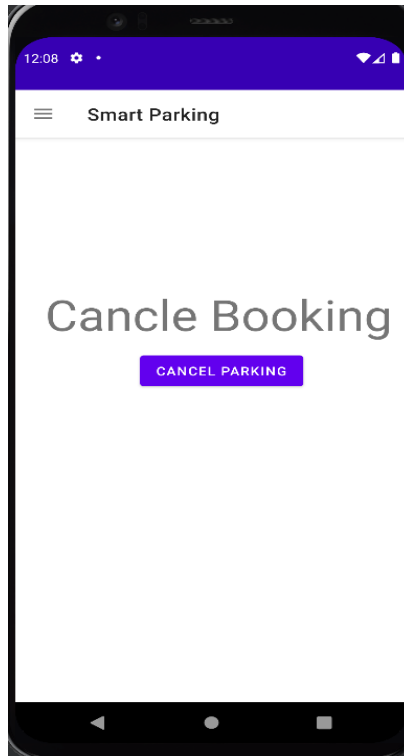
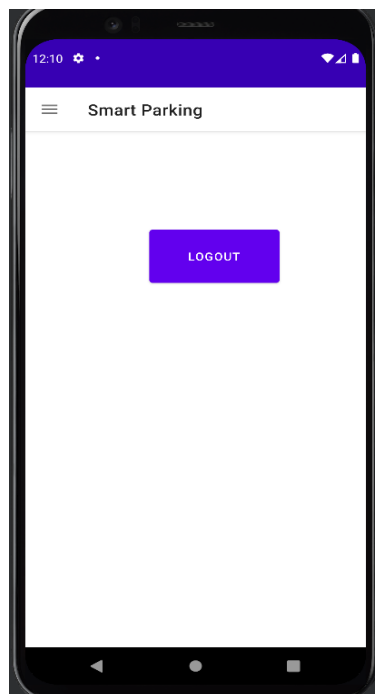
### 10.3 DASHBOARD



### 10.4 NAVIGATION BAR:

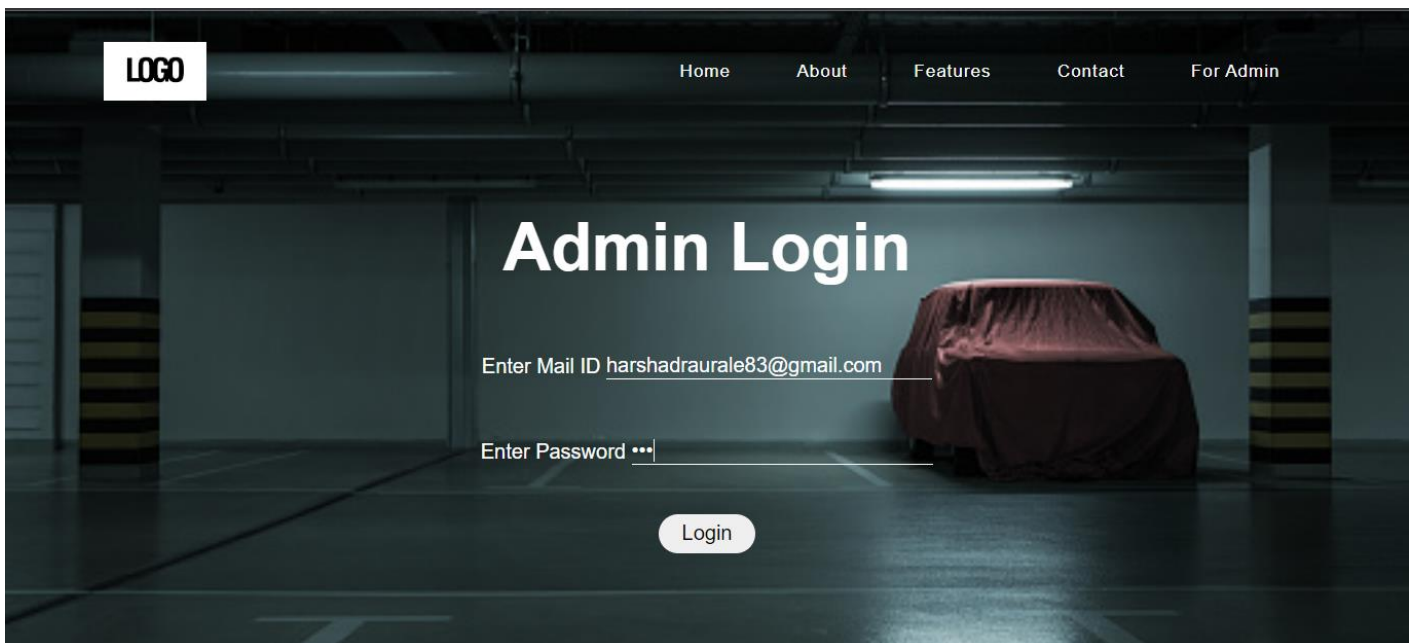
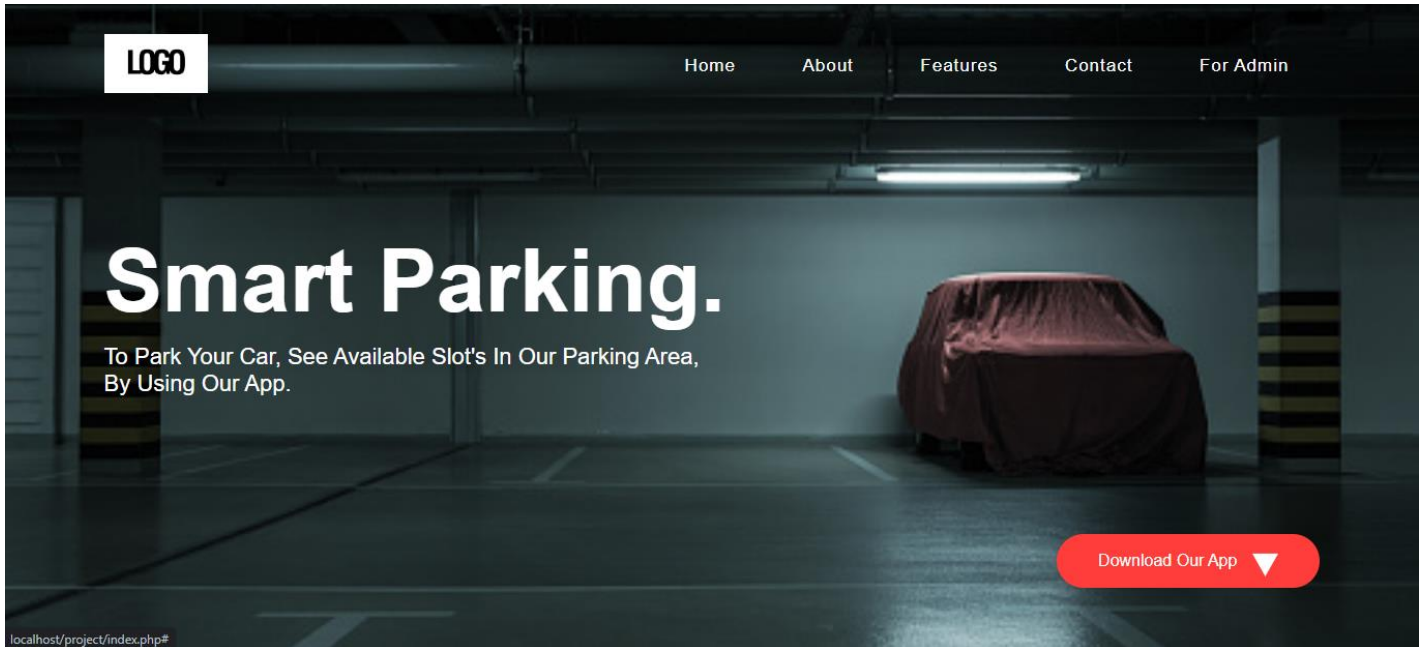


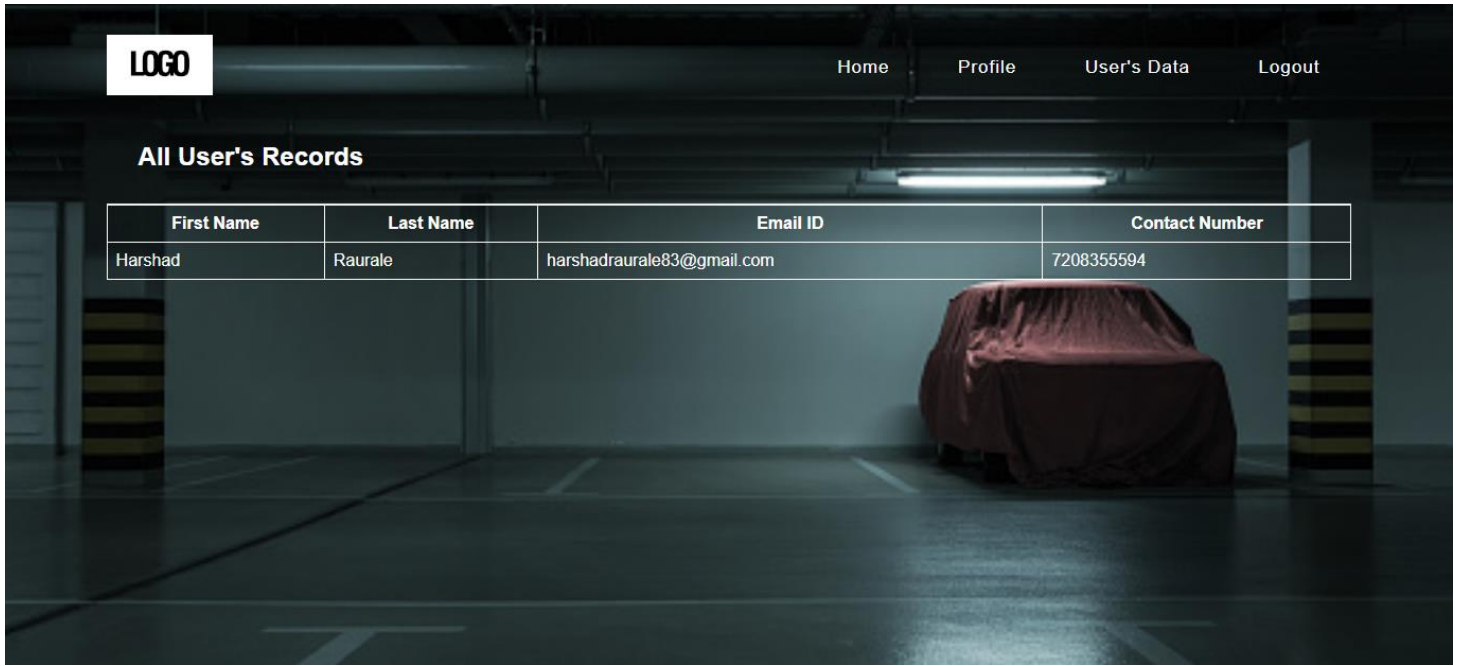
**10.5 SLOT BOOKING:****10.6 PARK OR GET THE CAR FROM THE PARKING:**

**10.7 CANCEL BOOKING:****10.8 LOGOUT:**



## 10.10 ADMIN WEBSITE:



The image shows a web application interface for a Smart Parking System. The background is a dark, dimly lit parking garage with a car covered in a red cloth. The interface has a dark theme. At the top left is a white box with the word "LOGO". To its right are navigation links: "Home", "Profile", "User's Data", and "Logout". Below the navigation bar, the text "All User's Records" is displayed. Underneath this is a table with four columns: "First Name", "Last Name", "Email ID", and "Contact Number". The table contains one row of data for a user named Harshad Raurale.

First Name	Last Name	Email ID	Contact Number
Harshad	Raurale	harshdraurale83@gmail.com	7208355594

# **CHAPTER 11**

# **CONCLUSION**

---

## 11. CONCLUSION:

Analysis of the model has to be done while developing a life size model. The mechanical model has been designed and the software as well as the control circuit has been implemented successfully. It demonstrates the working of the planned automated smart parking system. The main advantages are space optimization, cost effectiveness and security. The system benefits of the smart parking go well beyond avoiding the needless circling of city blocks. It also enables cities to develop fully integrated multimodal intelligent transportation systems that don't rely on cars in the first place.



# **CHAPTER 12**

# **BIBLIOGRAPHY**

---

## 12. BIBLIOGRAPHY:

1. <https://mobidev.biz/blog/iot-based-smart-parking-system>
2. <https://nevonprojects.com/iot-car-parking-system/>
3. Precision - Principles, Practices and Solutions for the Internet of Things Book by Timothy Chou