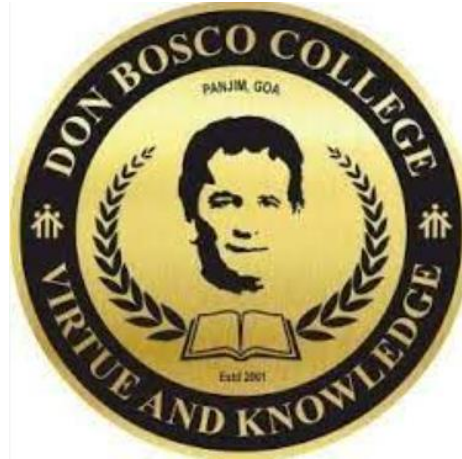


SPARK
A Smart parking web application



A PROJECT REPORT SUBMITTED TO GOA UNIVERSITY IN
PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF BCA.

BY:

| NAME | ROLL NO |
|------------------|---------|
| VISHAL PRASAD | 65 |
| ALDRICH MONTEIRO | 8 |
| ARVIND NAIDU | 19 |
| ETHAN D'COSTA | 17 |
| JIYA JUNEJA | 23 |
| RACHEL WAIKAR | 42 |
| SIDDHI NAIK | 50 |
| SUHAIB ANSARI | 48 |
| SUHANA KHAN | 57 |

Mr. Shantanu Arlekar
(Project Guide)

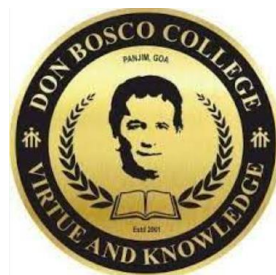
Mr. Rohan Kerkar
(Project Coordinator)

Dr. Cedric Silveira
(Principal)

DECLARATION OF CANDIDATES:

We declare that this project titled “SPARK”- A Smart parking web application aimed to manage all the parking facilities to a user by providing proper security and reducing the manual work.

| SEAT NUMBER | NAME | SIGNATURE |
|--------------------|-----------------------------|------------------|
| | VISHAL PRASAD | |
| | ALDRICH MONTEIRO | |
| | ARVIND NAIDU | |
| | ETHAN D’COSTA | |
| | JIYA JUNEJA | |
| | RACHEL WAIKAR | |
| | SIDDHI NAIK | |
| | SUHAIB ANSARI | |
| | SUHANA KHAN | |



DBSHEP's

Don Bosco College, Panaji

Affiliated to Goa University

CERTIFICATE

This is to certify that a project on “SPARK- A Smart parking web application” has been successfully completed by

VISHAL PRASAD

ALDRICH MONTEIRO

ARVIND NAIDU

ETHAN D'COSTA

JIYA JUNEJA

RACHEL WAIKAR

SIDDHI NAIK

SUHAIB ANSARI

SUHANA KHAN

Studying in **T.Y.B.C.A** during the academic year 2023-2024.

The project has been carried out under the supervision of the Internal Guide.

Mr. Shantanu Arlekar

(Project Guide)

Dr. Cedric Silveria

(Principal)

(External Examiner)

Mr. Rohan Kerkar

(Project Coordinator)

Place: Panaji, Goa

Date:

ACKNOWLEDGEMENT

It gives us immense pleasure to present “Spark”-A Smart parking web application.

We take this opportunity to extend our sincere gratitude to Mr. Shantanu Arlekar, our Internal Guide for this valuable guidance and encouragement throughout the project.

We would especially like to thank all the teachers of the BCA department who helped make our project success.

Last but not the least we would like to extend our gratitude to our parents and friends for their immeasurable guidance and encouragement whenever we needed it the most.

TABLE OF CONTENT

| SR NO. | CONTENT | Pg. No. |
|-------------------|------------------------------------|----------------|
| 1 | INTRODUCTION | 01-02 |
| 2 | EXISTING SYSTEM | 03-04 |
| 3 | PROPOSED SYSTEM | 05-06 |
| 4 | SOFTWARE ENGINEERING MODEL | 07-9 |
| 5 | ENTITY RELATIONSHIP DIAGRAM | 10-11 |
| 6 | CLASS DIAGRAM | 11-12 |
| 7 | USE CASE DIAGRAM | 13-14 |
| 8 | ACTIVITY DIAGRAM | 16-19 |
| 9 | DATABASE DESIGN | 20-28 |
| 10 | SOFTWARE REQUIREMENT SPECIFICATION | 29-42 |
| 11 | VALIDATION TEST REPORT | 43-48 |
| 12 | SYSTEM INTEGRATION REPORT | 49-56 |
| 13 | USER MANUAL | 57-67 |
| 14 | FUTURE ENHANCEMENTS | 68-69 |
| 15 | APPENDICES | 70-73 |
| 16 | GANTT CHART | 74-75 |

INTRODUCTION

Introduction

The project entitled smart parking system is to help manage all the parking facilities for a user. The recent growth in economy and due to the availability of low-price cars in the market, every average middle-class individual can afford a car, however the consequences of this are heavy traffic jams, pollution, less availability of roads and spot to drive car and other vehicles. One of the important concerns, which is to be taken in accounting, is the problem of parking those vehicles. Though, even if there is space for parking the vehicle so much time is squandered in finding that exact parking slot resulting in more fuel intake and which isn't also environment friendly. It will be a great deal if in some way we find out that the parking itself can provide the precise vacant position of a parking slot then it'll be helpful not limited to the drivers also for the environment. Initially when the user is about to enter the location the LCD displays the number of empty and filled spots and when the user is with its vehicle near to the parking detect sensor, he/she would be thrown with a notification on their mobile app of the parking slot number, where they should park their vehicle.

Objectives

Smart Parking involves the use of low-cost sensors, real-time data and applications that allow users to monitor available and unavailable parking spots. The goal is to automate and decrease time spent manually searching for the optimal parking floor, spot and even lot. Some solutions will encompass a complete suite of services such as online payments, parking time notifications and even car searching functionalities for very large lots. A parking solution can greatly benefit both the user and the lot owner.

1. **Optimized parking** – Users find the best spot available, saving time, resources and effort. The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities. Reduced traffic – Traffic flow increases as fewer cars are required to drive around in search of an open parking space.
2. **Reduced pollution** – Searching for parking burns around one million barrels of oil a day. An optimal parking solution will significantly decrease driving time, thus lowering the amount of daily vehicle emissions and ultimately reducing the global environmental footprint.
3. **Increased Safety** – Parking lot employees and security guards contain real-time lot data that can help prevent parking violations and suspicious activity. License plate recognition cameras can gather pertinent footage. Also, decreased spot-searching traffic on the streets can reduce accidents caused by the distraction of searching for parking.
4. **Decreased Management Costs** – More automation and less manual activity saves on labour cost and resource exhaustion.
5. **Enhanced User Experience** – A smart parking solution will integrate the entire user experience into a unified action. Driver's payment, spot identification, location search and time notifications all seamlessly become part of the destination arrival process.

EXISTING SYSTEM

The existing system for parking is a very time-consuming as well as draining endeavour.

The driver often has to search around blocks for a parking spot which can lead to a buildup in traffic and also ruin the driver's driving experience

Common Aspects of Traditional Parking Systems are:

Time-Consuming

Firstly, traditional parking is notorious for its time-taking factor. In this method, it is purely a matter of luck finding the parking spot first to save time and all the hassle.

Hassle of Finding Parking Spots

In traditional car parking, it is always filled with the hassle of moving your car here and there while locating an empty space.

Reduced Parking Spaces

In traditional car parking system, it was disoriented in nature as drivers used to follow their own free will in parking their cars, etc. This decreases the overall parking spaces at any place by making things congested owing to asymmetrical car parking queues

Build-up of Traffic on the Road

The traditional parking system was the primal cause of inviting extra traffic on the road. It is physically challenging for the driver to look for a parking spot at locations like beaches. This further increases the overall traffic on the road as the parking-search traffic gets intermingled with them. This ultimately leads to jammed roads and huge traffic congestion.

Poor Driving Experience

The traditional method of parking can severely affect the driver's journey and experience. As aforementioned, things like finding a parking space can be difficult owing to the physical limitation and unavailability of parking spaces. This can completely kill the drivers' experience as they may get late for their meetings, interviews, classes, etc.

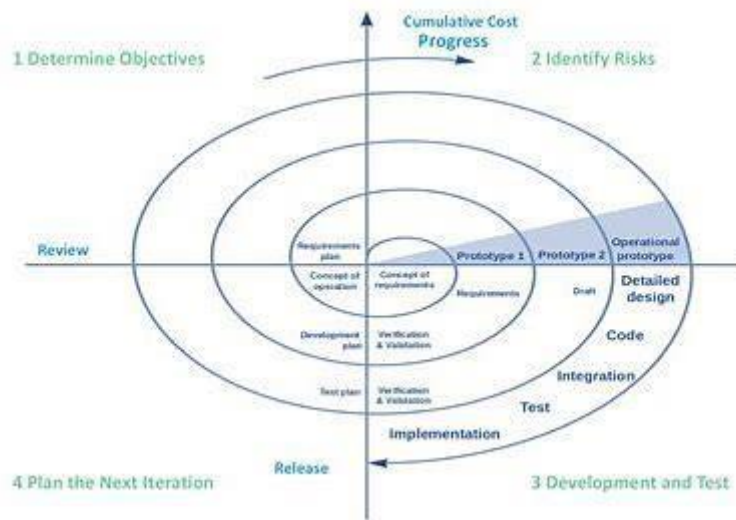
PROPOSED **SYSTEM**

The aim of our proposed system is to develop a system of improved facilities for parking. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

- In this system, users must keep a minimum amount of balance on their entry card in order to gain access to the parking system; otherwise, the system will deny access.
- As a result of using automation technology, this smart parking system will help reduce human effort and time.
- This IOT smart parking system provides the following advantages:
 - Automatic Parking System with Zero Human Intervention
 - RFID Scan for Access Verification
 - Automatic Gate Barriers for Entry-Exit
 - Parking Slot Sensors for Empty Slot Detection
 - Online Parking Slot availability on Phone
 - Easy to Use System
- To achieve smart parking, the system employs a number of technologies.
- RFID scanner, Esp8266 Wi-Fi Module, module motors, IR sensors, Microcontroller, and electronics components are used to create this system.
- Here are some points on the working of proposed system of RFID-based smart parking system:
 1. The proposed car parking system uses RFID reader at the entrance to sense the authorized vehicle and allocates the available parking slots to the vehicle.
 2. The proposed system incorporated an infrared sensor in each slot for getting information about the vacancy position of the parking slot.
 3. The user can book-parking slot well in advance, all the necessary information is available on the server.
 4. Every user has an exclusive username and password.
 5. The modified car parking system is RFID based system in which wireless communication is established for authentic car and gate of parking hence any vehicle passing through parking automatically displays an availability of parking space to user and there's no need to stop at gate.

SOFTWARE **ENGINEERING** **MODEL**

Spiral Model



Advantages and disadvantages of our chosen model

Advantages

- Software is produced early in the software life cycle.
- Risk handling is one of important advantages of the Spiral model, it is best development model to follow due to the risk analysis and risk handling at every phase.
- Flexibility in requirements. In this model, we can easily change requirements at later phases and can be incorporated accurately. Also, additional Functionality can be added at a later date.
- It is good for large and complex projects.
- It is good for customer satisfaction. We can involve customers in the development of products at early phase of the software development. Also, software is produced early in the software life cycle.
- Strong approval and documentation control.
- It is suitable for high-risk projects, where business needs may be unstable. A highly customized product can be developed using this

Disadvantages

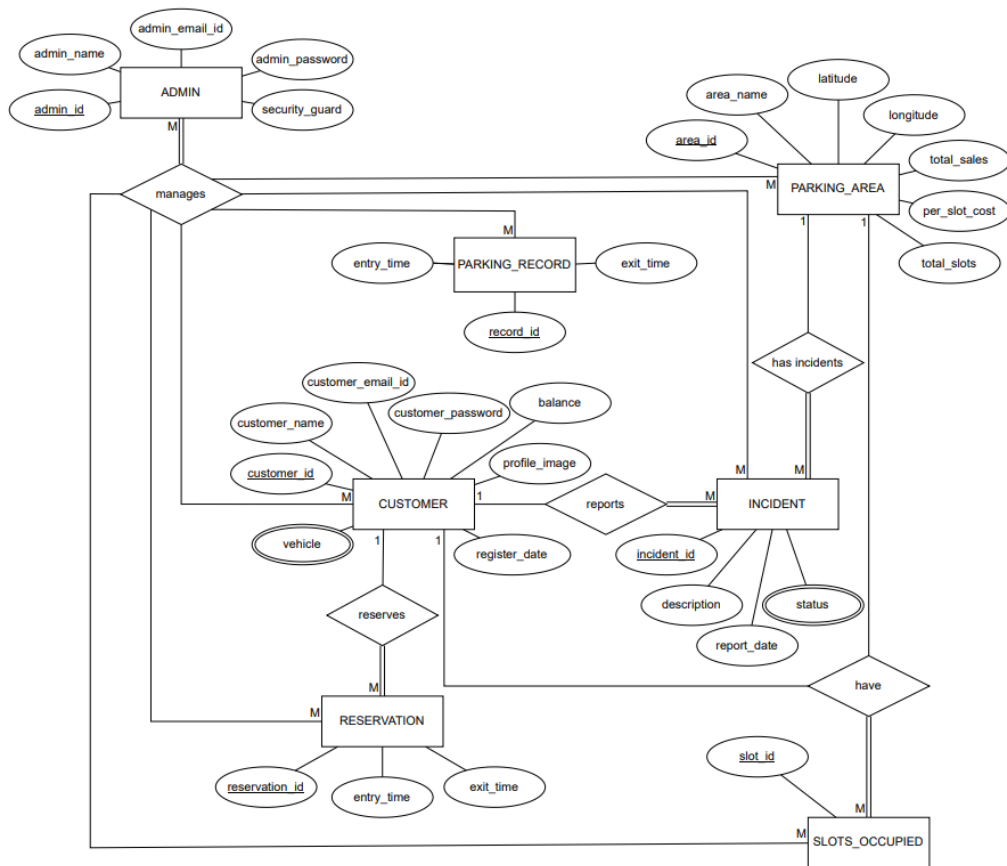
- It is not suitable for small projects as it is expensive • Process is complex.
- Too much dependable on Risk Analysis and requires highly specific expertise.
- Difficulty in time management. As the number of phases is unknown at the start of the project, so time estimation is very difficult.
- Spiral may go on indefinitely hence end of the project may not be known early.

- Hard to define objective, verifiable milestones. Large numbers of intermediate stages require excessive documentation

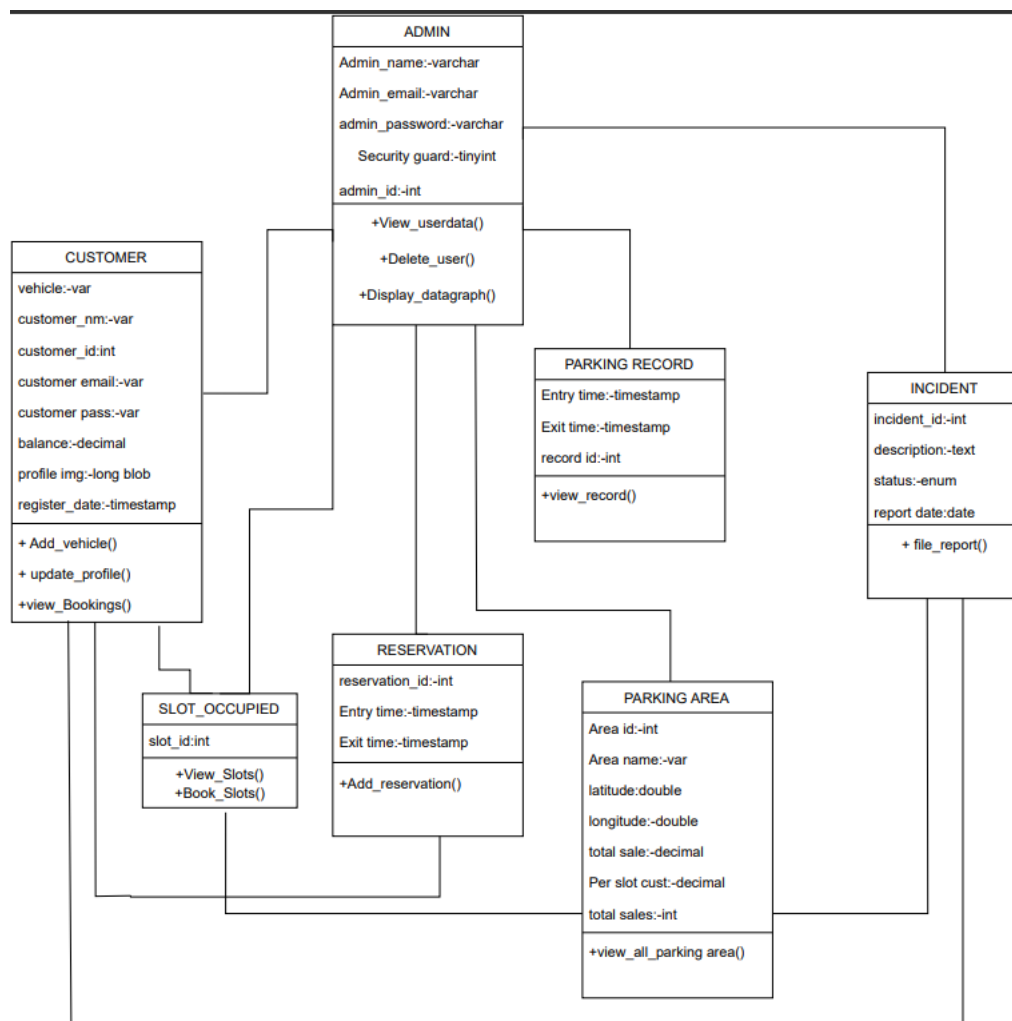
Why the chosen model suits our project best

1. **Risk assessment:** Spiral Model focuses on assessing risk and identifying solutions to care of those risks. Since a prototype is produced at the end of the risk analysis phase, it allows us to handle it early in the development cycle if risks are found by implementing alternative solutions.
2. **Flexibility in Requirements:** The spiral model affords us the ability to be flexible by allowing us to change requirements at later phases and incorporate those changes in the project,
3. **Plan-driven and highly structured:** This model results in a structured approach for creating medium to large projects and thus is best suited for our project.

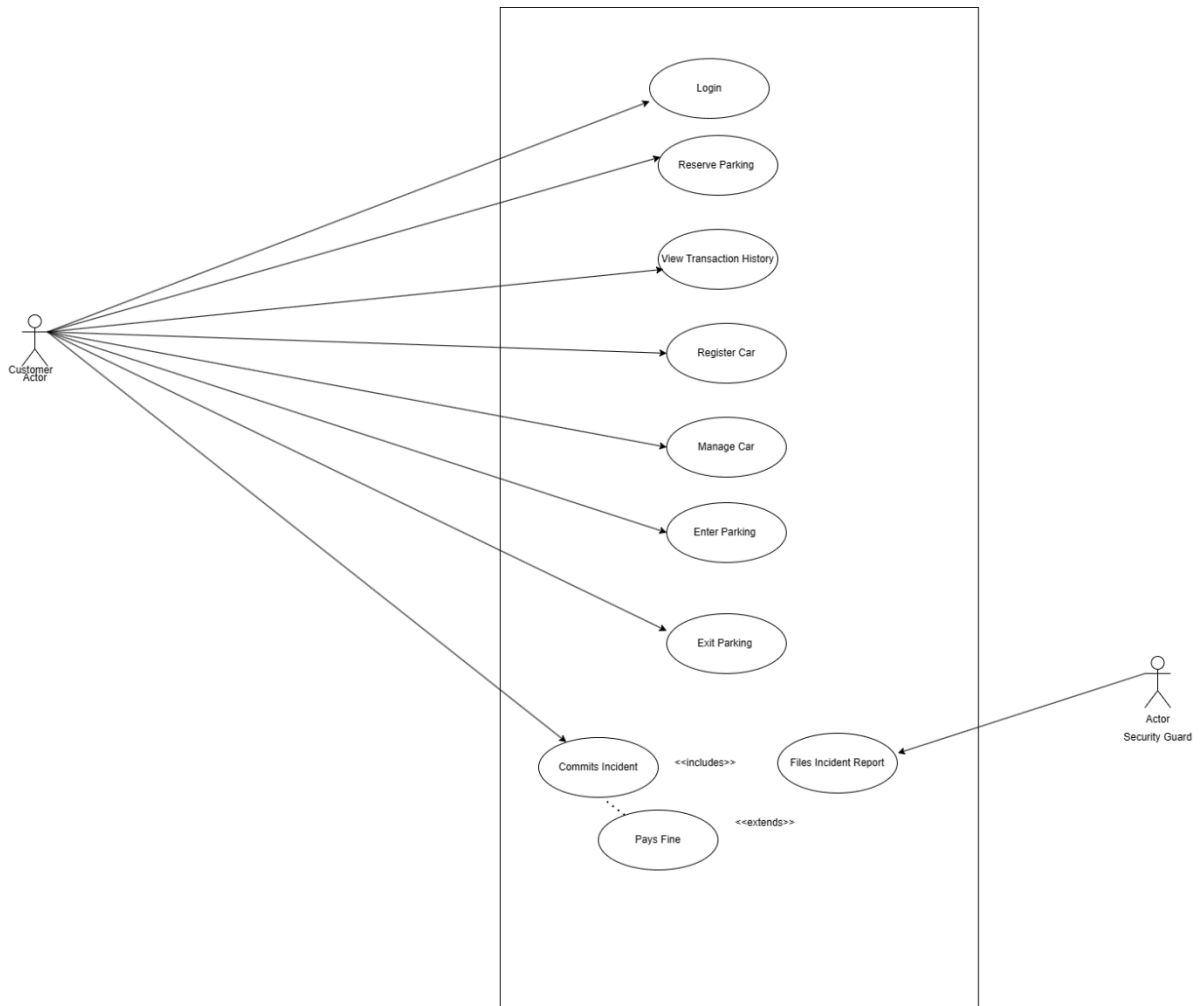
ENTITY **RELATIONSHIP** **DIAGRAM**



CLASS DIAGRAM

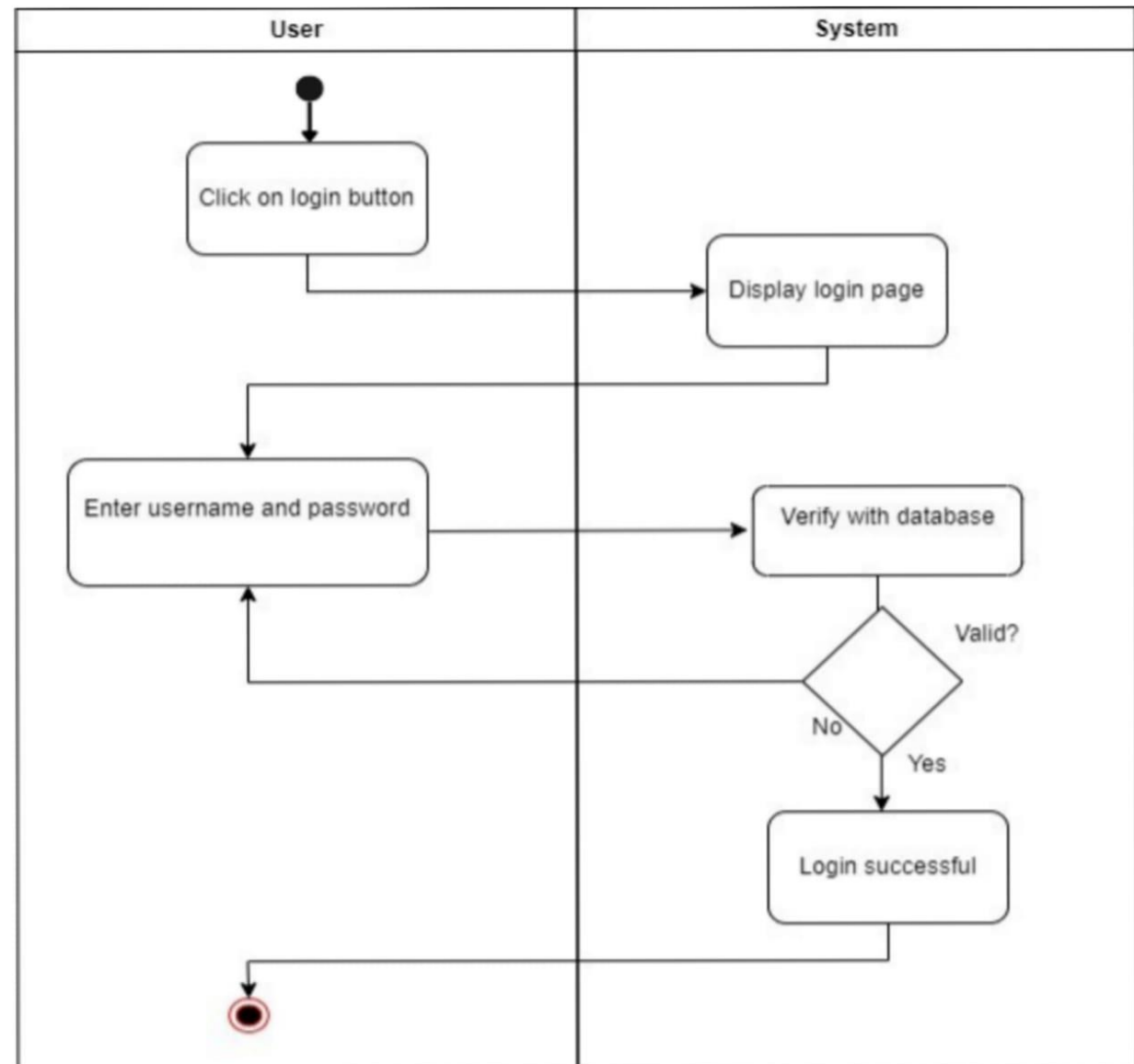


USE-CASE **DIAGRAM**

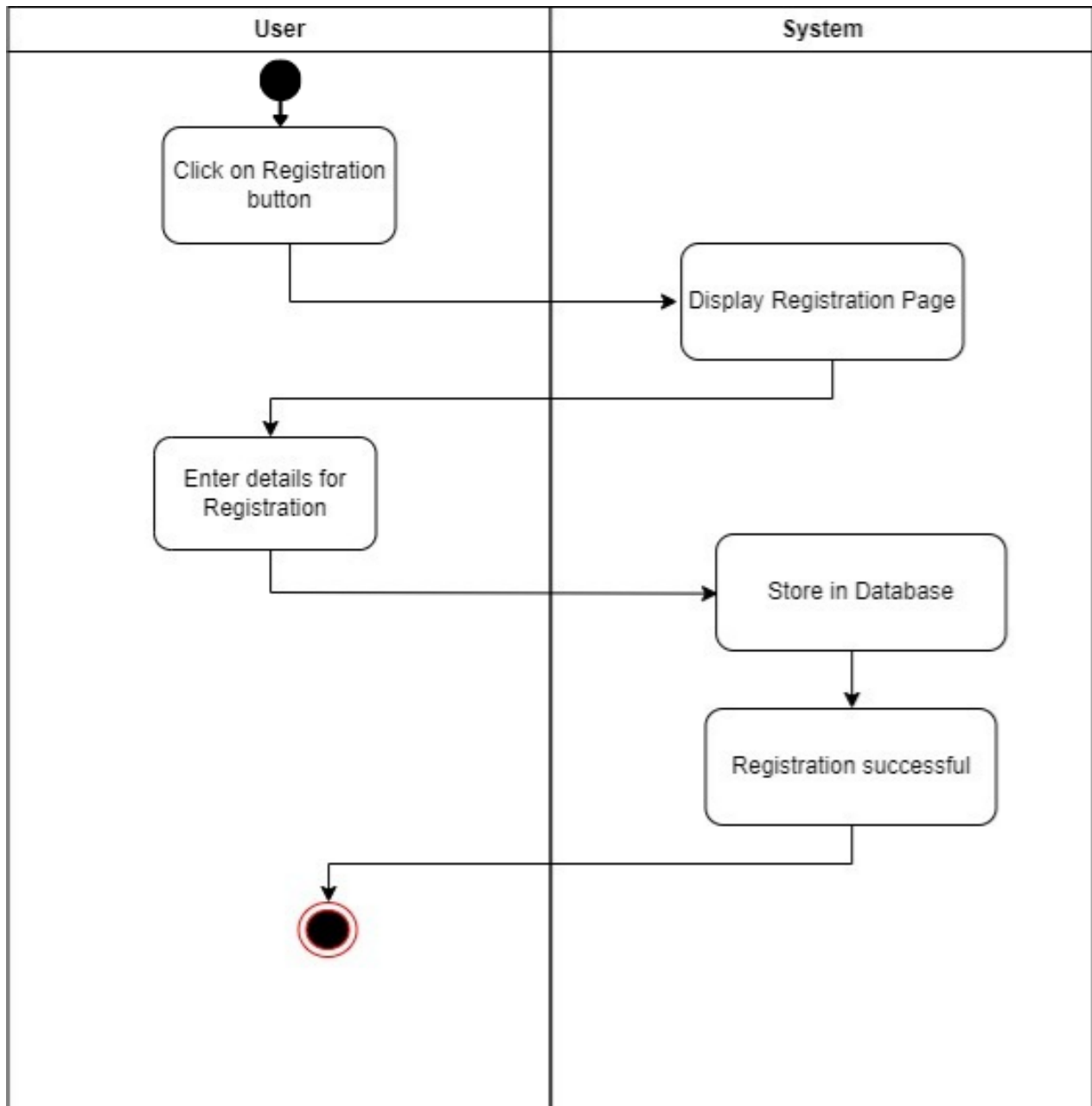


ACTIVITY **DIAGRAMS**

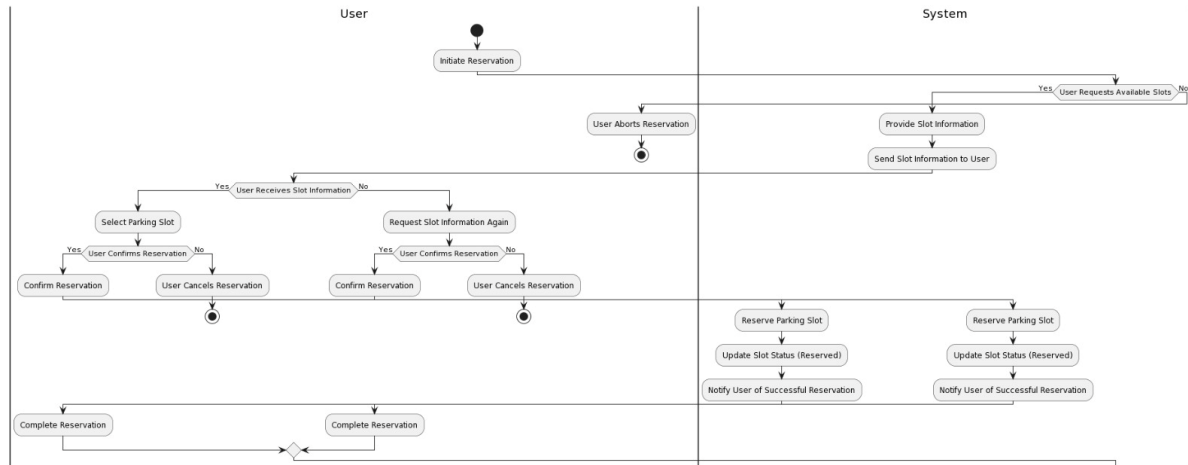
LOGIN PAGE:



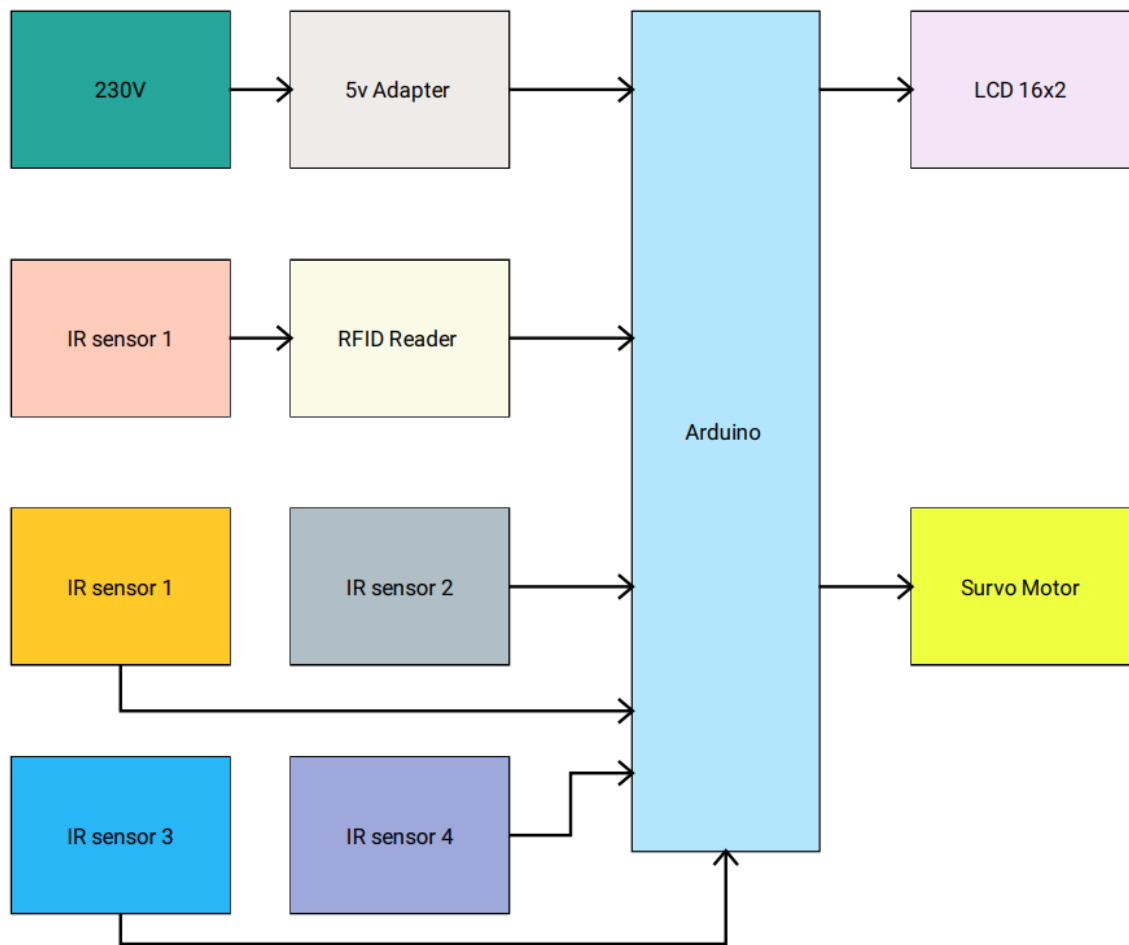
REGISTRATION PAGE:



RESERVATION PAGE

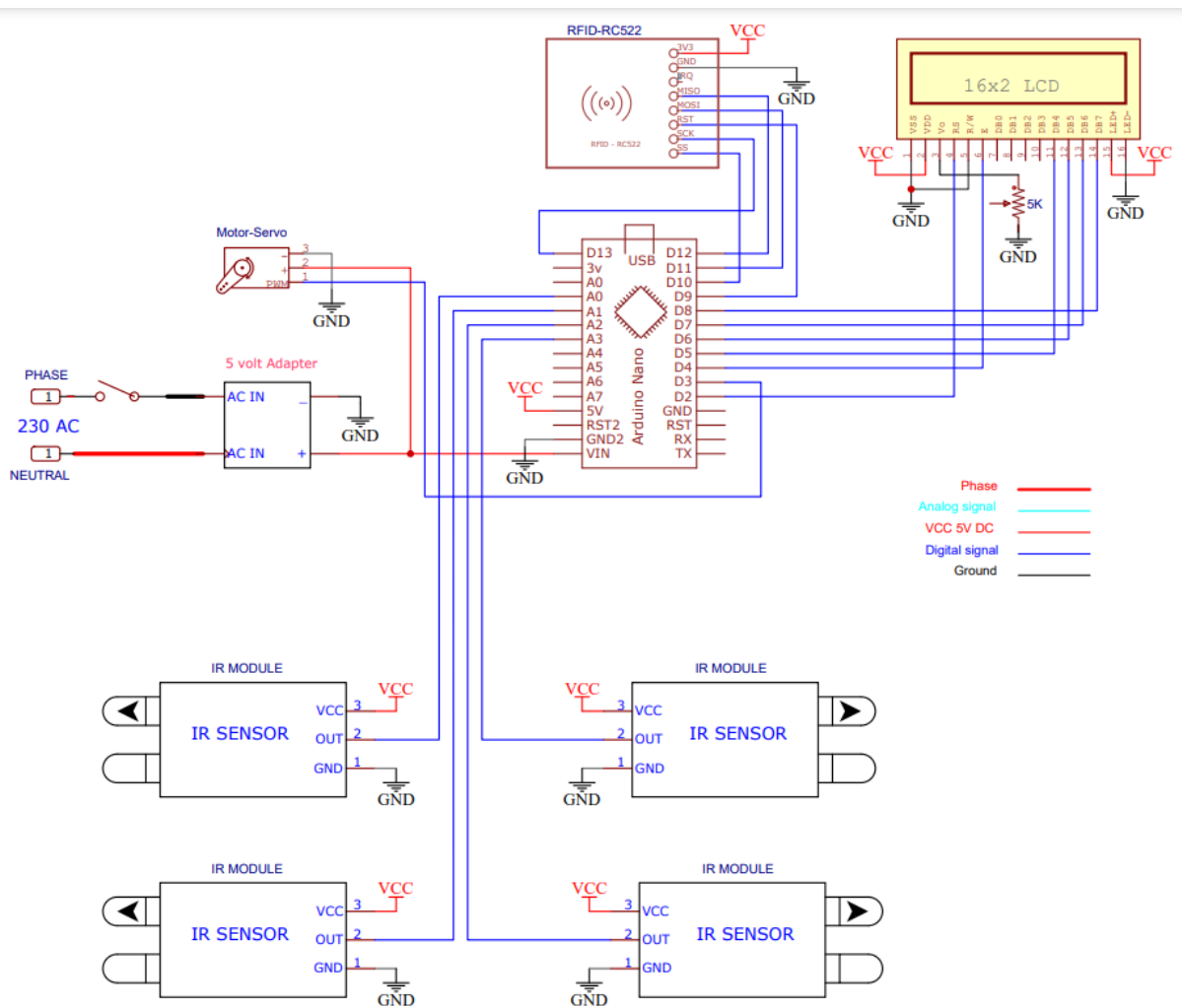


BLOCK DIAGRAM



SCHEMATIC

DIAGRAM



DATABASE DESIGN

Table name: Admin

Description: This table stores the Admin details

Primary key: admin_id

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|----------------|-----------|-----------------------|---------------------------------------|
| admin_id | int | Admin ID | PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| admin_name | varchar | Name of the admin | NOT NULL |
| admin_email_id | varchar | Password of the admin | unique_admin_email_id |
| admin_password | varchar | Email of the admin | NOT NULL |
| security_guard | tinyint | No of Security guards | NOT NULL |

Table name: Admin_control

Description: This table stores the primary keys of all tables which the admin can handle.

Primary key:

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|----------------|-----------|----------------|-------------|
| admin_id | int | Admin ID | FOREIGN KEY |
| customer_id | int | Customer ID | FOREIGN KEY |
| area_id | int | Area ID | FOREIGN KEY |
| incident_id | int | Incident ID | FOREIGN KEY |
| reservation_id | int | Reservation ID | FOREIGN KEY |
| slot_id | int | Slot ID | FOREIGN KEY |
| record_id | int | Record ID | FOREIGN KEY |

Table name: Customer

Description: This table stores the customers details.

Primary key: customer_id

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|--------------------|-----------|-----------------------------------|---------------------------------------|
| customer_id | int | Customer ID | PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| customer_name | varchar | Customer Name | NOT NULL |
| customer_email_id | varchar | Customer Email ID | NOT NULL |
| customer_password | varchar | Customer Password | NOT NULL |
| balance | decimal | Wallet Balance | NOT NULL |
| profile_img | longblob | Profile picture image of the user | NOT NULL |
| register_timestamp | timestamp | Timestamp of user registration | NOT NULL |

Table name: Customer_vehicle

Description: This table stores the details of the customer's vehicle.

Primary key: customer_id

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|-------------|-----------|--------------------|---|
| customer_id | int | Customer ID | PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| vehicle | char | Customer's Vehicle | NOT NULL |

Table name: incident

Description: This table stores the details of the incidents reported by customers.

Primary key: incident_id

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|------------------|-----------|---|---|
| incident_id | int | Incident ID | PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| description | text | Description of Incident | NOT NULL |
| report_timestamp | timestamp | Time of the report | NOT NULL |
| customer_id | int | ID of the Customer | FOREIGN KEY |
| area_id | int | Parking area where incident occurred | FOREIGN KEY |

Table name: incident_status

Description: This table relates to the status of incidents reported by customers.

Primary key: incident_id

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|-------------|-----------|--|---|
| incident_id | int | Incident ID | PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| status | enum | Status of the incident whether solved,unsolved or pending | NOT NULL |

Table name: parking_area

Description: This table stores the parking area details.

Primary key: area_id

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|---------------|-----------|--|---|
| area_id | int | Area id | PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| area_name | varchar | Area name | NOT NULL |
| latitude | double | Latitude of area | NOT NULL |
| longitude | double | Longitude of area | NOT NULL |
| total_sales | decimal | Total sales of that particular area | NOT NULL |
| per_slot_cost | decimal | Per slot cost | NOT NULL |
| total_slots | int | Total slots | NOT NULL |

Table name: parking_record

Description: This table stores the record of each customers including entry and exit time.

Primary key: record_id

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|------------|-----------|-------------|---|
| record_id | int | Record id | PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| Entry_time | timestamp | Entry time | NOT NULL |
| Exit_time | timestamp | Exit time | NOT NULL |

Table name: reservation

Description: This table stores the slot reserved details of each customer.

Primary key: reservation_id

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|----------------|-----------|--|---|
| reservation_id | int | Record id | PRIMARY KEY, NOT NULL, AUTO INCREMENT |
| Entry_time | timestamp | Entry time | NOT NULL |
| Exit_time | timestamp | Exit time | NOT NULL |
| Customer_id | int | Customer id of the one who reserved | FOREIGN KEY |

Table name: slot_occupied

Description: This table stores the foreign keys of to link them.

| ATTRIBUTE | DATA TYPE | DESCRIPTION | CONSTRAINTS |
|-------------|-----------|-------------|-------------|
| slot_id | int | Slot ID | FOREIGN KEY |
| customer_id | int | Customer ID | FOREIGN KEY |
| area_id | int | Area ID | FOREIGN KEY |

SOFTWARE

REQUIREMENT

SPECIFICATION

HARDWARE AND SOFTWARE:

Hardware:

- Arduino Nano
- Capacitor
- Resistor
- Sensor IR
- SMPS Adapter
- Servo motor
- Jumper Wires
- Buzzer
- Breadboard
- RFID Reader
- RFID Card
- PCB
- Soldering Wires

Software:

- Web Browser (Google Chrome, Edge)
- Database-MySQL 8.0.27
- Operating System- Windows 7 or latest
- Python
- HTML, CSS, JavaScript, C++
- PHP

SOFTWARE:

1) FRONT-END TOOLS:

a) HTML:

It is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as CSS and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a webpage semantically and originally included cues for the appearance of the document.

b) CSS:

CSS (Cascading Style Sheets) is used to style and layout web pages — for example, to alter the font, colour, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features.

c) JavaScript

JavaScript, often abbreviated as JS, is a programming language and core technology of the World Wide Web, alongside HTML and CSS. As of 2024, 98.8% of websites use JavaScript on the client side for webpage behaviour, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

2) **BACK-END TOOLS:**

d) **Python:**

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

e) **MySQL:**

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by **Oracle Company**.

f) **PHP:**

PHP is a general-purpose scripting language geared towards web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. The PHP reference implementation is now produced by the PHP Group. PHP was originally an abbreviation of Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

f) **C++**

Arduino code is written in C++ with an addition of special methods and functions, which we'll mention later on. C++ is a human-readable programming language. When you create a 'sketch' (the name given to Arduino code files), it is processed and compiled to machine language.

PLATFORMS:

a) VISUAL STUDIO:

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer.

b) ARDUNIO IDE:

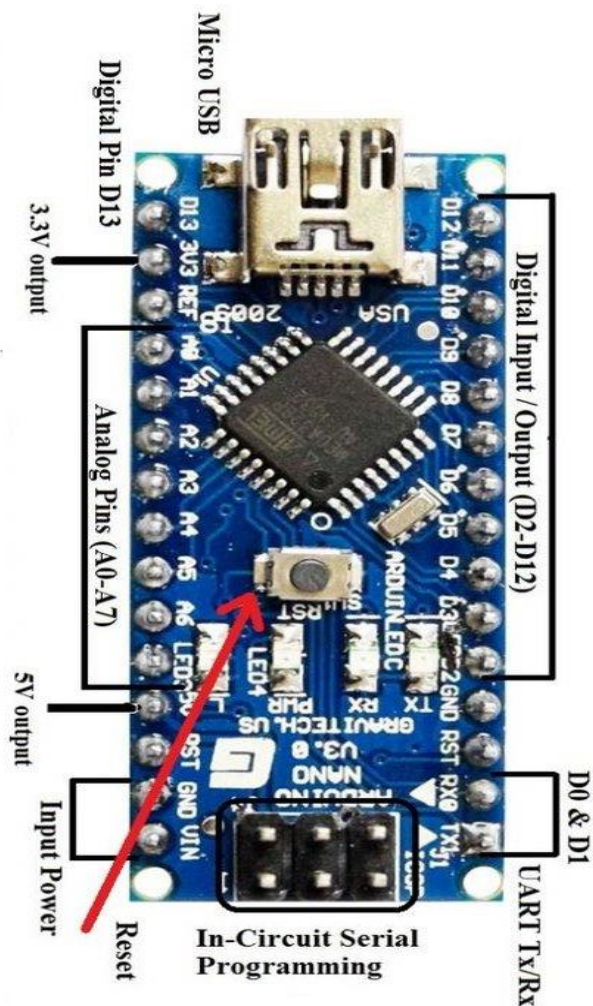
Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software.

c) MySQL WORK-BENCH:

MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system.

HARDWARE:

ARDUINO NANO:



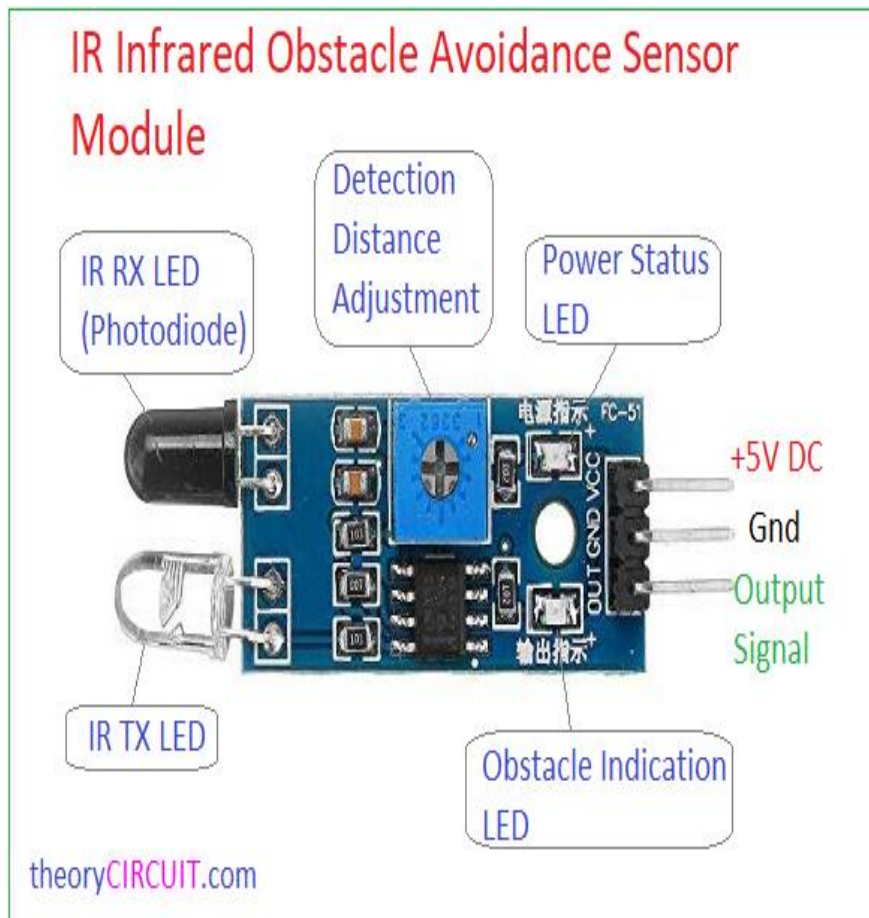
- The Arduino Nano is a small, compact microcontroller board based on the ATmega328P microcontroller.
- It has 14 digital input/output pins, 6 analog input pins, and 6 PWM pins.
- It also has a 16 MHz quartz crystal oscillator, a mini USB port, and a power jack for external power supply.
- The Nano can be programmed using the Arduino software, which is free and easy to use.
- The board can be powered either by USB connection or an external power supply (7-12V DC).
- The size of the Nano is 45mm x 18mm, making it one of the smallest Arduino boards available.
- The board can be used for a wide range of applications, including robotics, automation, sensing, and control.
- The Nano is compatible with a wide range of shields and sensors, making it a versatile choice for DIY electronics projects.
- It has a built-in voltage regulator, which allows it to operate from 5V to 12V.

LCD 16X2:



- An LCD 16x2 display is a type of alphanumeric display that consists of two rows of 16 characters each, hence the name 16x2.
- It is a commonly used display in various electronic devices such as calculators, digital clocks, and programmable controllers.
- The display uses a liquid crystal material that changes its polarization when a voltage is applied to it, allowing light to pass through and create characters.
- The display is controlled using a microcontroller or other digital circuitry that sends commands and data to the display via a parallel or serial interface.
- Each character on the display is typically made up of five by eight pixels, although some displays may use a larger font.
- The display can display a variety of characters, including letters, numbers, symbols, and even simple graphics.
- The backlight of the display can be turned on or off depending on the lighting conditions of the environment.
- The display can be customized by programming it to display specific characters or graphics, making it useful for creating custom displays.
- The display is generally low power and can operate on a voltage range of 4.5V to 5.5V.

IR SENSOR:



- A moisture sensor is an electronic device that detects the level of moisture or water content in a material or the surrounding environment.
- Moisture sensors can be used in various applications, such as agriculture, construction, and environmental monitoring, to measure humidity, soil moisture, water content in concrete, and more.
- The most commonly used types of moisture sensors are resistive, capacitive, and conductive sensors.
- Resistive sensors measure moisture by detecting changes in resistance caused by changes in moisture levels. They are inexpensive and easy to use but may not be as accurate as other types of sensors.
- Conductive sensors measure moisture by detecting changes in the conductivity of the material. They are often used in agriculture to measure soil moisture and are less expensive than capacitive sensors.
- Moisture sensors can be used in combination with other sensors to provide more comprehensive data, such as temperature and humidity sensors.
- Moisture sensors can be wired or wireless and can be integrated with other systems for automatic data collection and analysis.

SERVO MOTOR:



- Compact and lightweight design, making it suitable for various applications.
- Operates on low voltage (typically 4.8V to 6V), making it compatible with common power sources.
- Controlled by sending PWM (Pulse Width Modulation) signals from a microcontroller or other control devices.
- Offers a 180-degree rotation (90 degrees in each direction) with a limited range of motion.
- Gear-driven mechanism for precise and controlled movement.
- Generally, comes with three wires for power (VCC), ground (GND), and signal (PWM).
- Can be easily interfaced with Arduino, Raspberry Pi, or other microcontrollers.
- Used in robotics, model airplanes, remote-controlled cars, and other projects requiring controlled movement.
- Affordable and readily available in the market.
- Limited torque capability, suitable for lightweight applications.
- May not be suitable for heavy-duty or high-torque tasks.
- Provides a cost-effective solution for hobbyists and DIY projects.
- Requires careful power management to avoid overloading and damaging the servo motor.

BUZZER:



- A buzzer is an electronic device that produces a buzzing or beeping sound when activated.
- Buzzer devices are typically composed of a coil of wire, a diaphragm, and a contactor mechanism that creates the buzzing or beeping sound.
- When an electric current is applied to the coil, it generates a magnetic field that attracts the contactor mechanism to the diaphragm, causing it to vibrate and create the sound.
- Buzzer devices can be either active or passive. Active buzzers have a built-in oscillator circuit and can produce sound with just a DC voltage supply, while passive buzzers require an external oscillator circuit or pulse generator to produce sound.
- Buzzer devices can be classified based on their frequency, with some devices producing a single tone and others producing multiple tones or melodies.
- Buzzer devices are commonly used in electronic devices, alarms, timers, and other applications where an audible signal is required.
- Buzzer devices are available in a wide range of sizes and shapes, including surface-mount, through-hole, and panel-mounted types.

RFID MODULE:



- RFID module facilitates wireless communication for data exchange
- Uses radio waves for identification and tracking of objects with RFID tags
- Comprises an RFID reader and antenna for communication
- Operates in different frequency ranges: LF, HF, UHF
- RFID tags available in various form factors: cards, labels, embedded chips
- Enables contactless data transfer for applications like access control, inventory management
- Supports read-only or read-write functionality based on tag type
- Integrates easily with microcontrollers or microprocessors
- Range varies with frequency (LF shorter, UHF longer)
- Commonly employed in logistics, retail, healthcare, and security
- Automates identification and tracking processes for increased efficiency
- Offers real-time data capture, enhancing system accuracy
- Security features, including encryption, can be implemented

RFID CARDS:



- Designed for contactless data exchange with RFID readers or other compatible devices.
- Commonly used in access control systems, public transportation, payment cards, and identification badges.
- Incorporate an RFID chip and antenna for communication purposes.
- Offer convenience and speed in applications like access control, where users can simply tap their card to gain entry.
- RFID cards can be passive (powered by the reader's signal) or active (contain a built-in power source for longer-range communication).
- Utilize various frequency bands, such as LF, HF, and UHF, depending on the application requirements.
- Can store and transmit a unique identifier or additional data, depending on the card's memory capacity.
- Provide a cost-effective and efficient solution for secure identification and authentication.
- Resistant to wear and tear, ensuring durability for everyday use.
- RFID card technology has widespread adoption in sectors like transportation, hospitality, and corporate environments.
- Integration with existing systems is relatively straightforward, making them versatile for different applications.

FILTER:



- A filter is an electrical circuit that is used to remove unwanted signals from a signal source.
- In voltage measurement circuits, filters are used to smooth the rectifier output voltage and reduce its ripple.
- There are two types of filters: passive filters and active filters.
- Passive filters use only resistors, capacitors, and inductors to filter the signal, while active filters use additional amplifiers.
- The most commonly used filter in voltage measurement circuits is the capacitor filter, which uses a capacitor to smooth the rectifier output voltage and reduce its ripple.
- The filter capacitor value is chosen based on the desired level of ripple reduction and the maximum allowable output voltage drop.
- The filter cut off frequency can be calculated using the following formula: $f = 1 / (2 * \pi * R * C)$, where R is the filter resistance and C is the filter capacitance.

General purpose PCB:



Here are some points about General Purpose PCBs:

- General Purpose PCBs (Printed Circuit Boards) are used to create electronic circuits for a wide range of applications.
- They are made of a non-conductive material, such as fiberglass or epoxy, with conductive tracks printed onto the surface to create a circuit.
- The conductive tracks are made of copper and are laid out in a specific pattern to connect electronic components together.
- General Purpose PCBs come in a variety of sizes, shapes, and designs, and can be custom-designed to fit specific applications.
- They can be single-sided, double-sided, or multi-layered, depending on the complexity of the circuit.
- General Purpose PCBs are cost-effective and widely available, making them a popular choice for hobbyists, students, and professionals.
- They are easy to assemble and modify, allowing for quick prototyping and testing of electronic circuits.
- General Purpose PCBs can be used for both through-hole and surface-mount components, making them versatile and suitable for a wide range of applications.

VALIDATION

TEST REPORT

Validation is the process of evaluating a system of components during or at the end of the development process to determine whether it is satisfied requirements.

The output produced by given activity represents the goal to be satisfied by that activity hence it is necessary to have validation for each output. Test validation is a procedure that demonstrates that a test is job-related and correlates to on-the-job performs.

So, validations involved actual testing that can be done after the verification is completed. Hence verification is a process evaluating a component or system to determine whether the products of given development stages satisfying the condition imposed at the start of the phase.

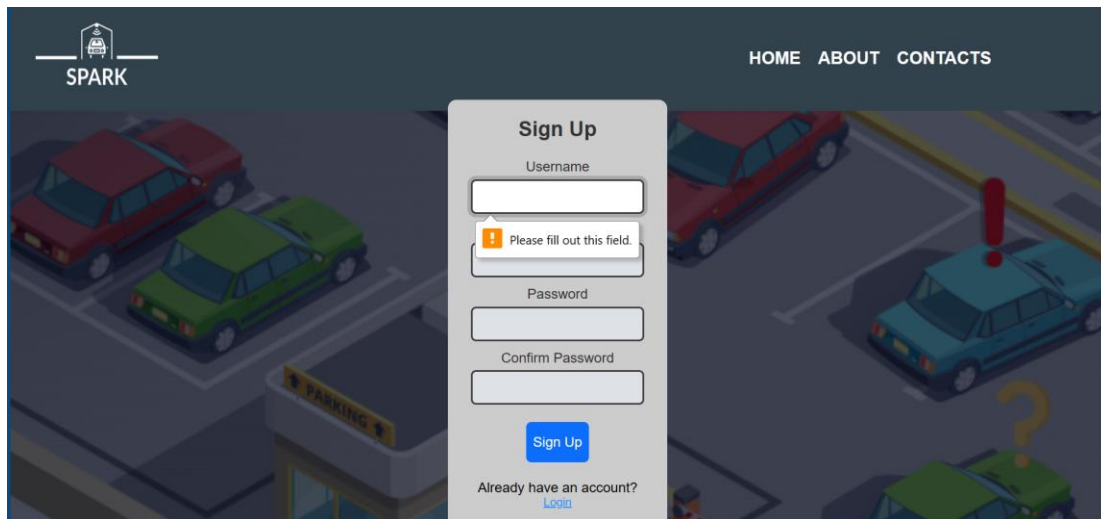
Validation is a process of checking in order to check whether they are working according to the requirements or not in the validations they give preference.

Validation is correctness of a component or system with respect to their requirements.

For this purpose, the validation testing is the responsibility of the tester and all the tests done in the box testing will be treating as validation tests.

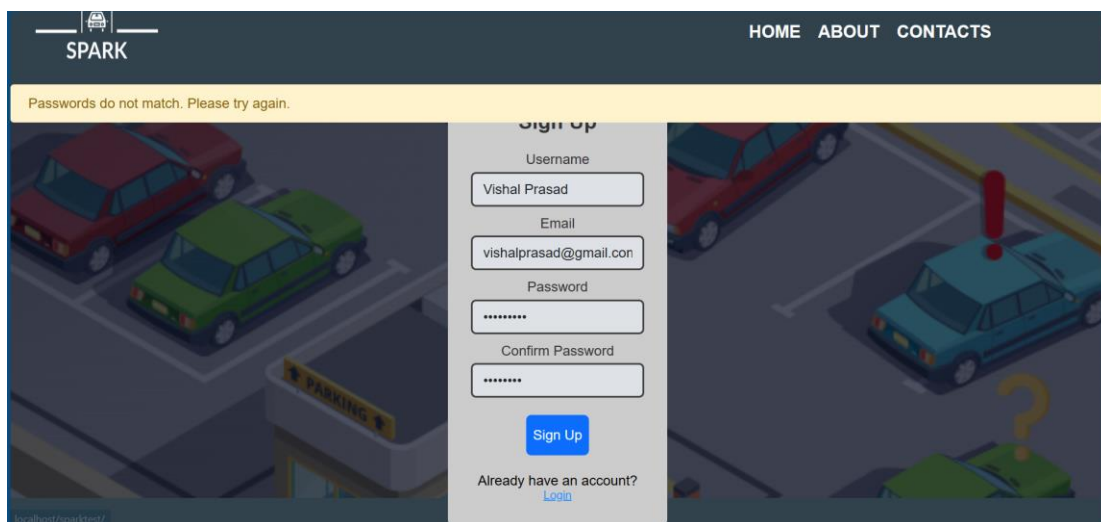
SIGN-UP PAGE FOR CUSTOMER:

The user must fill in all the required details asked in the registration form.



The screenshot shows the SPARK website's Sign Up page. The header includes the SPARK logo and navigation links: HOME, ABOUT, and CONTACTS. The background features an isometric illustration of a parking lot with several cars. The Sign Up form is centered and contains the following fields and elements:

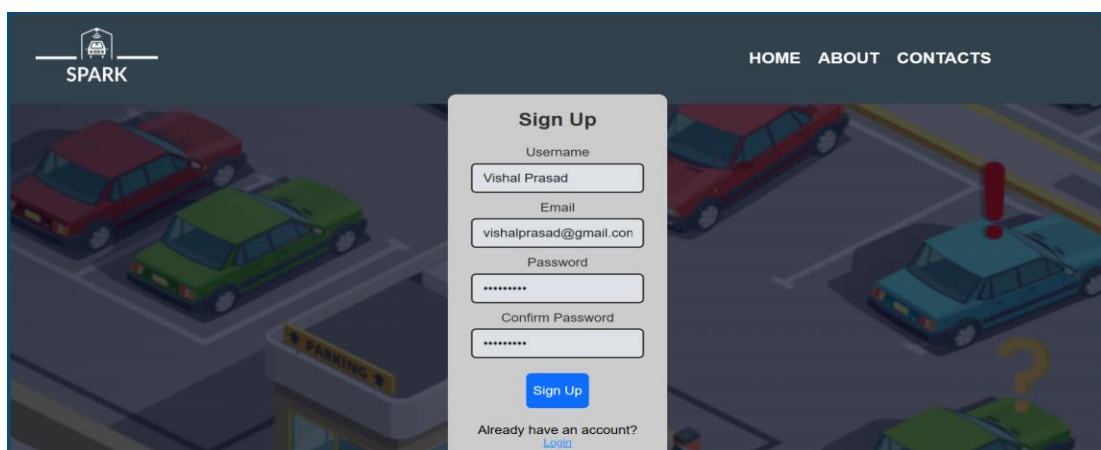
- Username:** A text input field with a validation error message: "Please fill out this field." (indicated by an orange exclamation mark icon).
- Password:** A text input field.
- Confirm Password:** A text input field.
- Sign Up:** A blue button.
- Already have an account?** A link to the Login page.



This screenshot shows the SPARK Sign Up page after a password mismatch error. A yellow banner at the top of the form area displays the message: "Passwords do not match. Please try again." The form fields are now populated with the following data:

- Username:** Vishal Prasad
- Email:** vishalprasad@gmail.com
- Password:** (masked with asterisks)
- Confirm Password:** (masked with asterisks)

The Sign Up button and the "Already have an account? Login" link remain visible at the bottom of the form.



This screenshot shows the SPARK Sign Up page with all form fields filled out. The data entered is:

- Username:** Vishal Prasad
- Email:** vishalprasad@gmail.com
- Password:** (masked with asterisks)
- Confirm Password:** (masked with asterisks)

The Sign Up button and the "Already have an account? Login" link are still present at the bottom of the form.

REPORT-01:

Project: Spark

Module: Sign Up Page

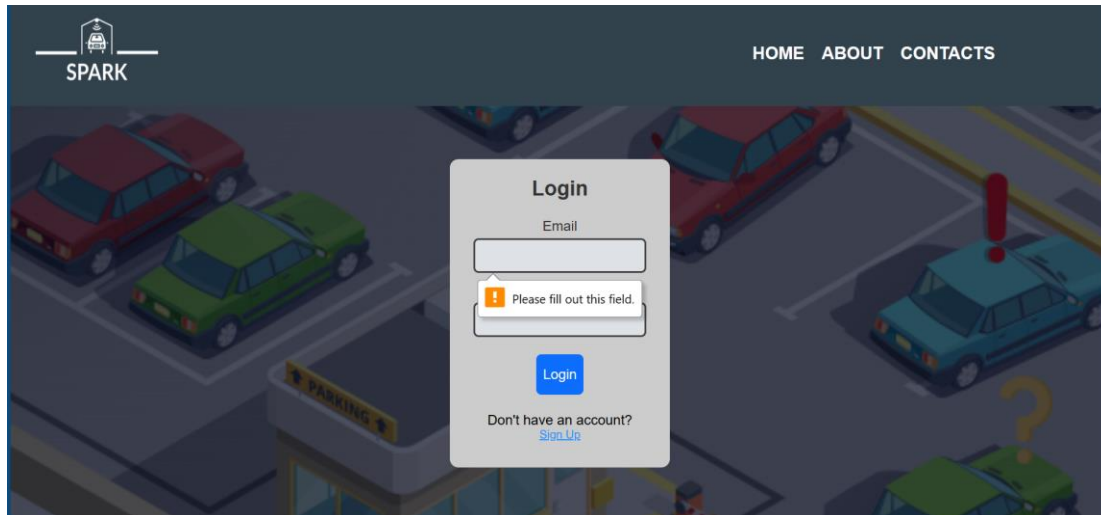
Function Specification: Sign Up for Customer

Test Objective: To Validate the Sign Up Page

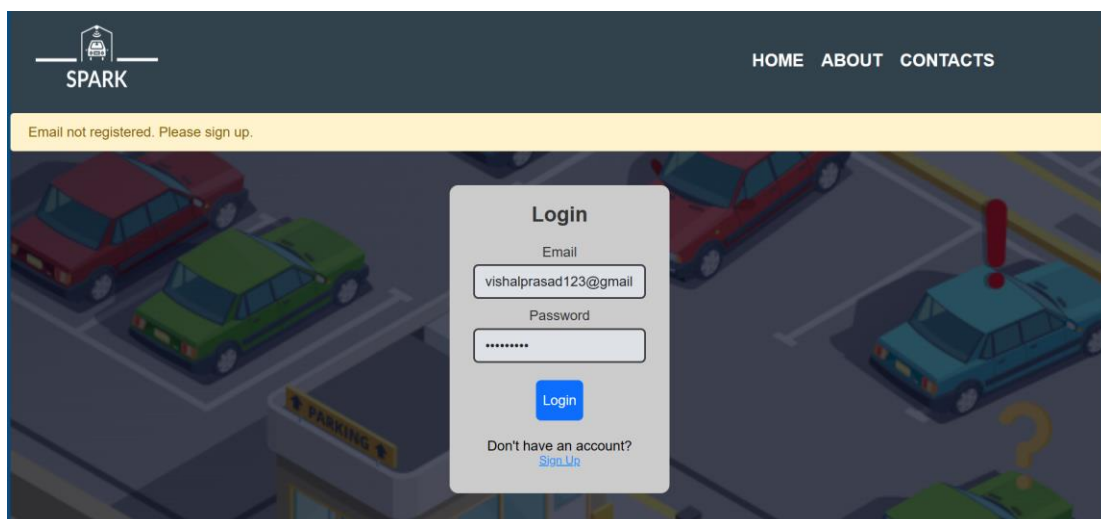
| Test Case No. | Event | Input Data | Expected Output | Actual Output | Result |
|---------------|---------------------|---------------|------------------------------------|---|---------|
| 01 | Enter User name | User name | Should Display Warning Message Box | Display Warning Message Box "Please fill out this field" | Success |
| 02 | Enter Email Address | Email Address | Should Display Warning Message Box | Display Warning Message Box "Please fill out this field" | Success |
| 03 | Enter Password | Password | Should Display Warning Message Box | Display Warning Message Box "Please fill out this field" | Success |
| 04 | Confirm Password | Password | Should Display Warning Message Box | Display Warning Message Box "Passwords do not match. Please try again." | Success |
| 05 | Confirm Password | Password | Password Match | Login Successful | Success |

LOGIN PAGE FOR CUSTOMER:

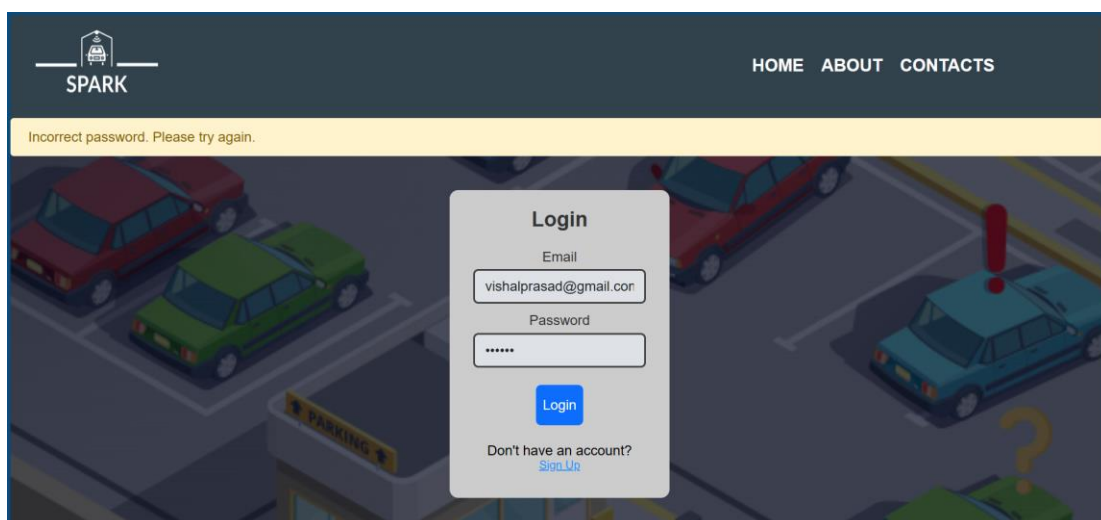
The user must fill in all the required details asked in the login form. In case either is missing or incorrect the error message shows up.



The screenshot shows the SPARK login page. The header includes the SPARK logo and navigation links: HOME, ABOUT, CONTACTS. The background is a stylized illustration of a parking lot with cars. A central login form is displayed with the title "Login". It has an "Email" label and an empty input field. Below the input field, a red error message box states "Please fill out this field." with an exclamation mark icon. A blue "Login" button is positioned below the input field. At the bottom of the form, it says "Don't have an account?" followed by a blue link "Sign Up".



The screenshot shows the SPARK login page with a yellow error banner at the top that reads "Email not registered. Please sign up." The login form is identical to the previous one, but the "Email" input field now contains the text "vishalprasad123@gmail". The "Password" input field is empty and has a label "Password" above it. The "Login" button and "Sign Up" link remain the same.



The screenshot shows the SPARK login page with a yellow error banner at the top that reads "Incorrect password. Please try again." The login form is identical to the previous ones, but the "Email" input field contains "vishalprasad@gmail.com" and the "Password" input field contains "*****". The "Login" button and "Sign Up" link remain the same.

REPORT-02:

Project: Spark

Module: Login Page

Function Specification: Login for Customer

Test Objective: To Validate the Login Page

| Test Case No. | Event | Input Data | Expected Output | Actual Output | Result |
|---------------|-----------------------------|---------------|------------------------------------|---|---------|
| 01 | Enter Email Address | Email Address | Should Display Warning Message Box | Display Warning Message Box “Please fill out this field | Success |
| 02 | Enter Invalid Email Address | Email Address | Should Display Warning Message Box | Display Warning Message Box “Email not registered. Please sign up.” | Success |
| 03 | Enter Password | Password | Should Display Warning Message Box | Display Warning Message Box “Incorrect password. Please try again.” | Success |
| 04 | Enter Valid Password | Password | Should Login the user | Successful Login | Success |

SYSTEM **INTEGRATION** **REPORT**

System integration means that all components of the system are integrated and tested as a single unit. Integration testing is testing of the interface. It can be divided into 2 types, namely:

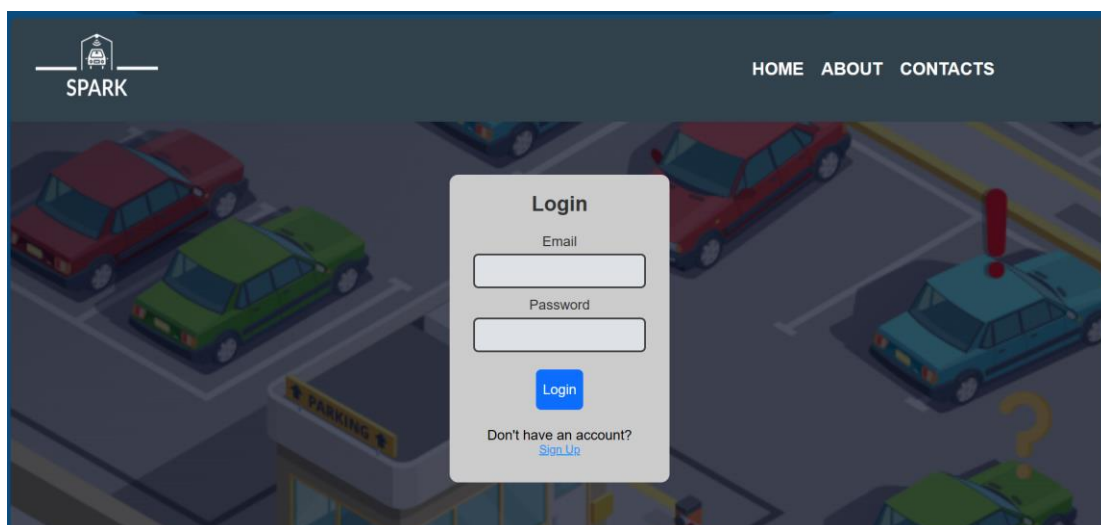
- Component or subsystem integration
- Final integration testing or system integration

The main aim is that optimization of integrating components and testing this approach is also called big bang integration. It reduces testing efforts and removes the duplication in testing.

System integration using the big bang approach is well suited in the product development scenario where the majority of the components are already available and stable and very few components are added or modified.

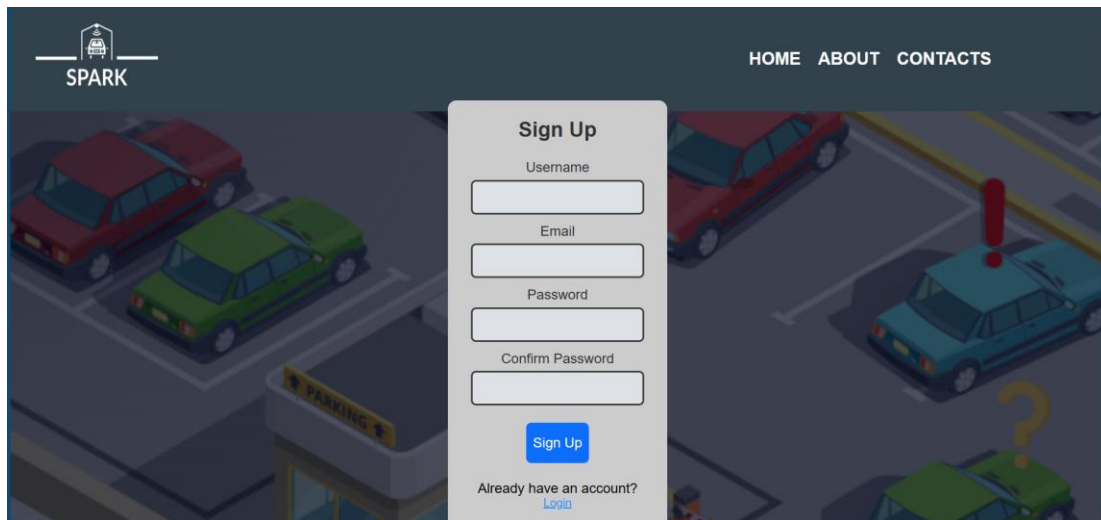
LOGIN PAGE:

This Page provides users with options to either Login with an already existing or Sign Up for a new account.



SIGN-UP PAGE:

Here the User gets to create new account by filling up the Registration details.

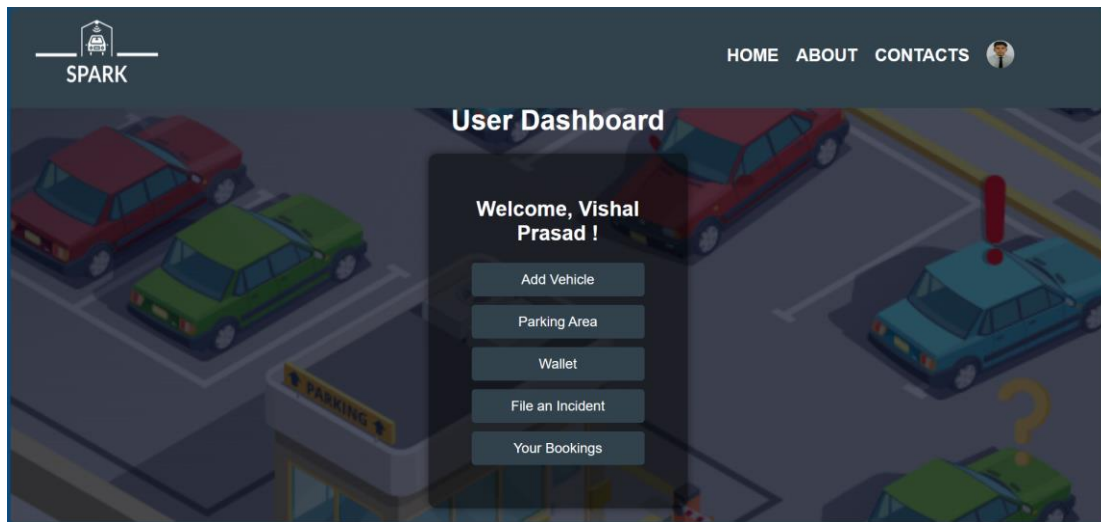


The image shows a web page for 'SPARK' with a dark blue header. The header contains the 'SPARK' logo on the left and navigation links 'HOME', 'ABOUT', and 'CONTACTS' on the right. The background of the page features an isometric illustration of a parking lot with several cars (red, green, blue) and a building with a 'PARKING' sign. A central white modal box titled 'Sign Up' contains the following elements:

- Sign Up** (Section Title)
- Username** (Label) with a text input field.
- Email** (Label) with a text input field.
- Password** (Label) with a text input field.
- Confirm Password** (Label) with a text input field.
- Sign Up** (Blue button).
- Already have an account?** (Text) with a [Login](#) link.

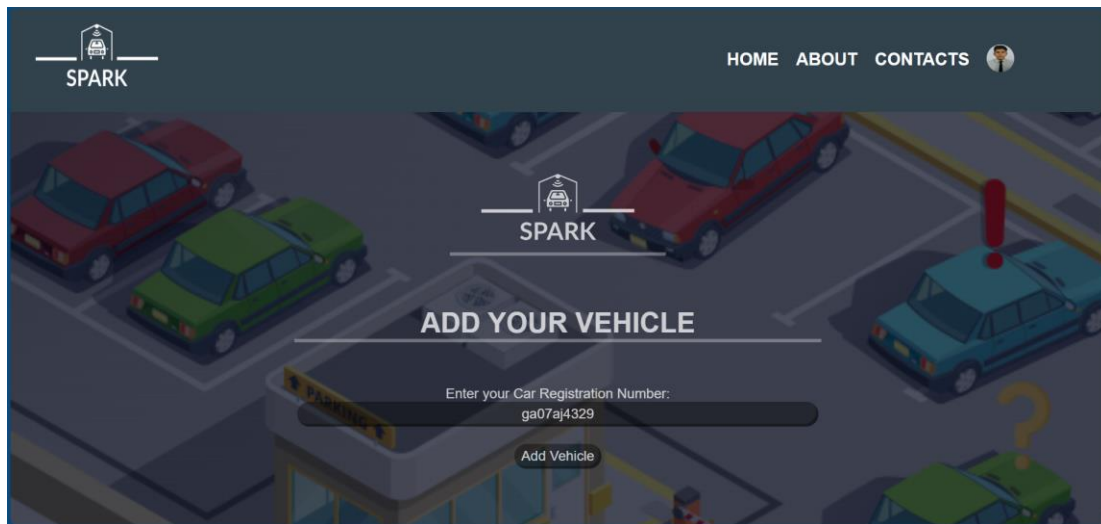
HOME PAGE:

The Home-page contains a dashboard for the user with options to add vehicle, view parking areas affiliated with Spark, check wallet balance, file an incident report and view your bookings.

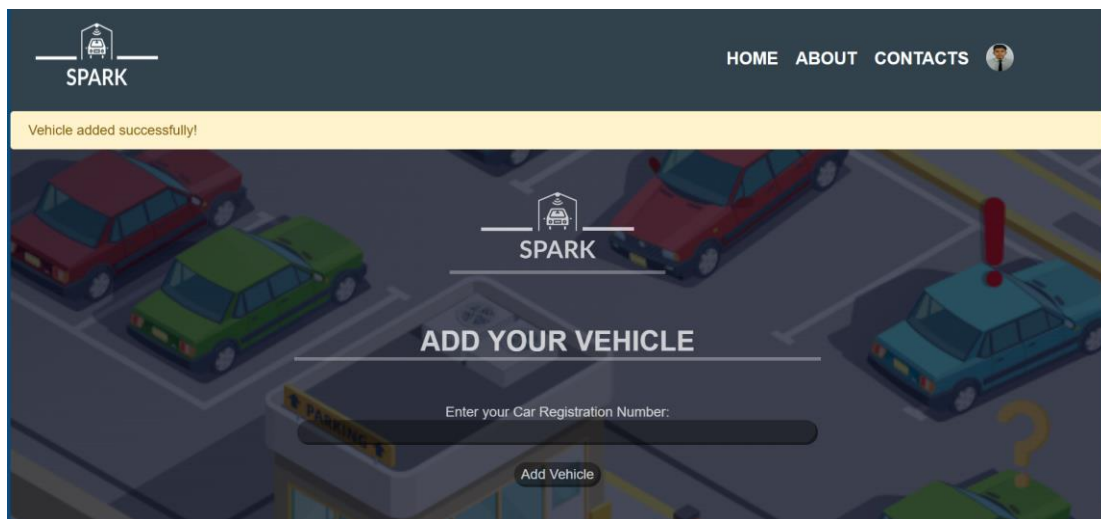


ADD VEHICLE:

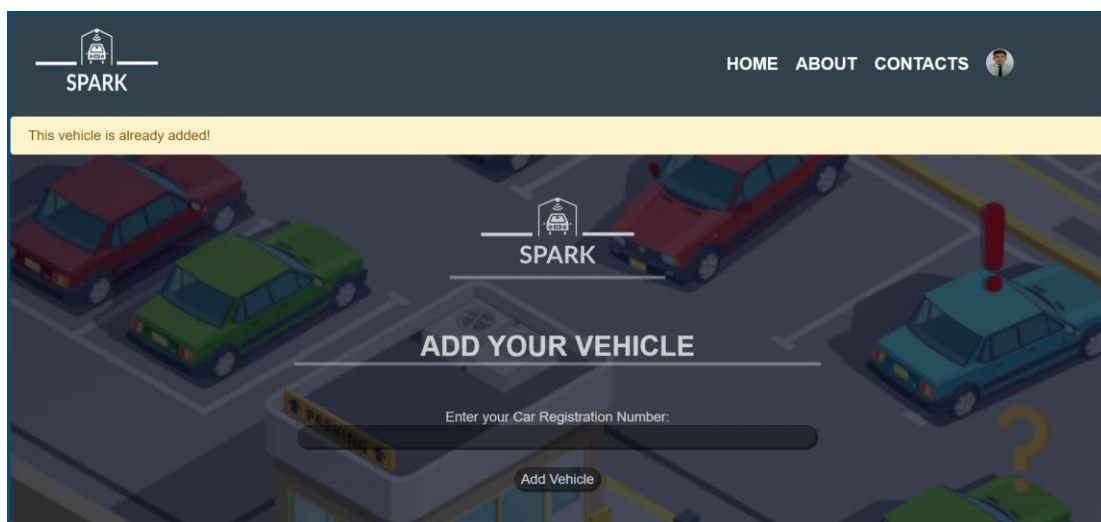
Here the user can add their products to the cart to buy them all together.



The screenshot shows the 'ADD YOUR VEHICLE' form on the SPARK website. The background is a dark blue isometric illustration of a parking lot with several cars. The form is centered and consists of a dark grey input field for the car registration number, which contains the text 'ga07aj4329'. Below the input field is a dark grey button labeled 'Add Vehicle'. The SPARK logo is at the top left, and the navigation menu (HOME, ABOUT, CONTACTS) with a user profile icon is at the top right.



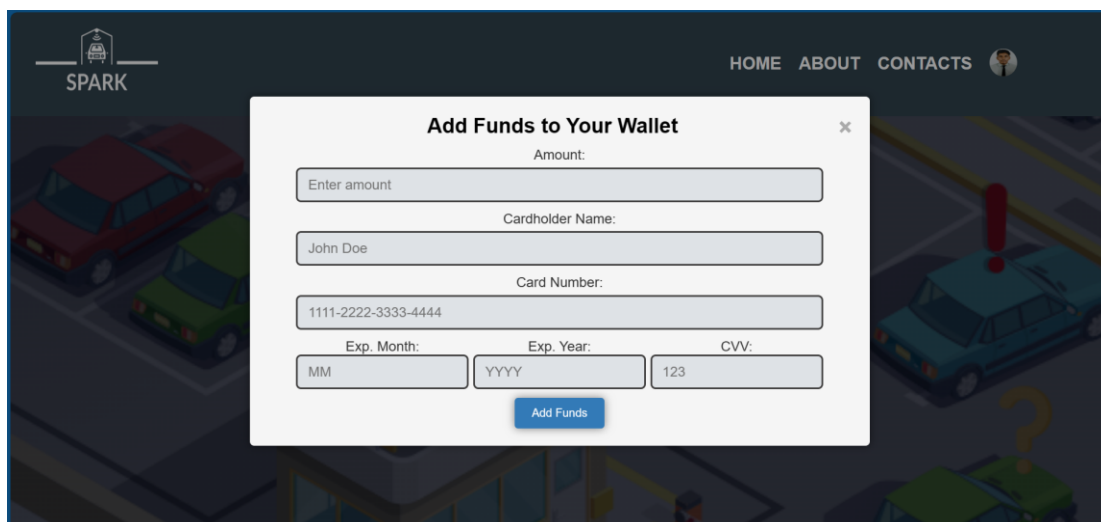
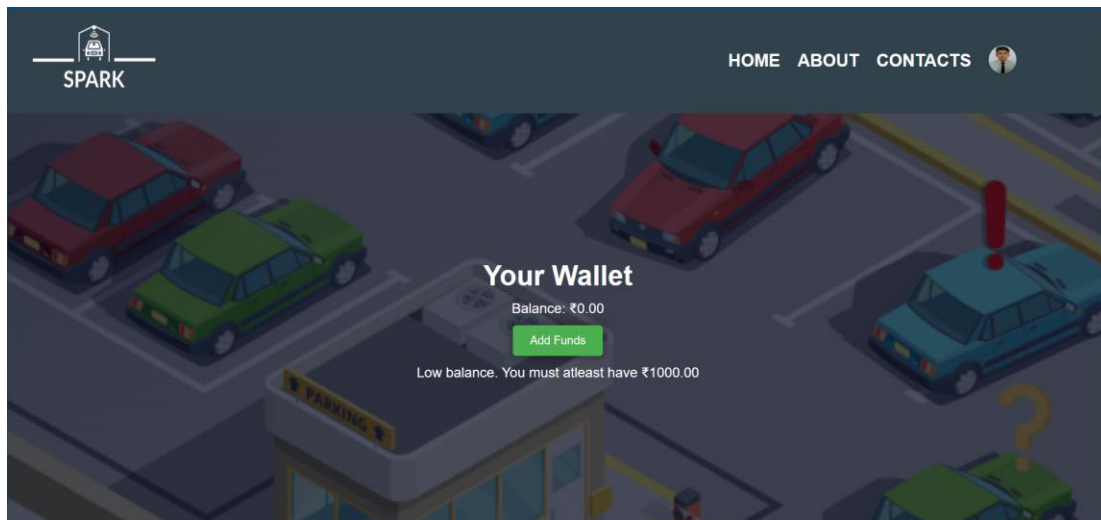
This screenshot shows the 'ADD YOUR VEHICLE' form with a success message. A yellow banner at the top of the form area displays the text 'Vehicle added successfully!'. The rest of the form, including the input field and the 'Add Vehicle' button, remains the same as in the previous screenshot.



This screenshot shows the 'ADD YOUR VEHICLE' form with an error message. A yellow banner at the top of the form area displays the text 'This vehicle is already added!'. The rest of the form, including the input field and the 'Add Vehicle' button, remains the same as in the previous screenshots.




WALLET:

Here the Customer can view their wallet balance and funds.



INCIDENT REPORT:

Here the customer can file an incident report if anything occurred in the parking area selected.




HOME ABOUT CONTACTS 


File An Incident

Type your problem and mention the entry and exit date and time and the slot number.

Select Area ▾

Submit



HOME ABOUT CONTACTS 


File An Incident


Someone parked in my reserved parking slot- slot 3.

Sunset Haven Parking Lot ▾

Submit

HOME ABOUT CONTACTS 

Incident reported successfully.



File An Incident

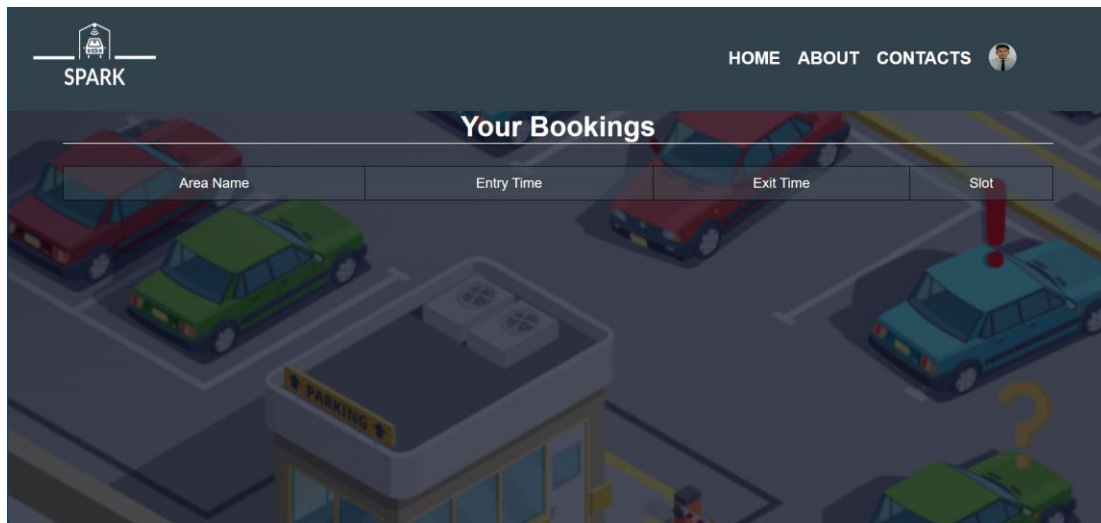
Type your problem and mention the entry and exit date and time and the slot number.

Select Area ▾

Submit

YOUR BOOKINGS:

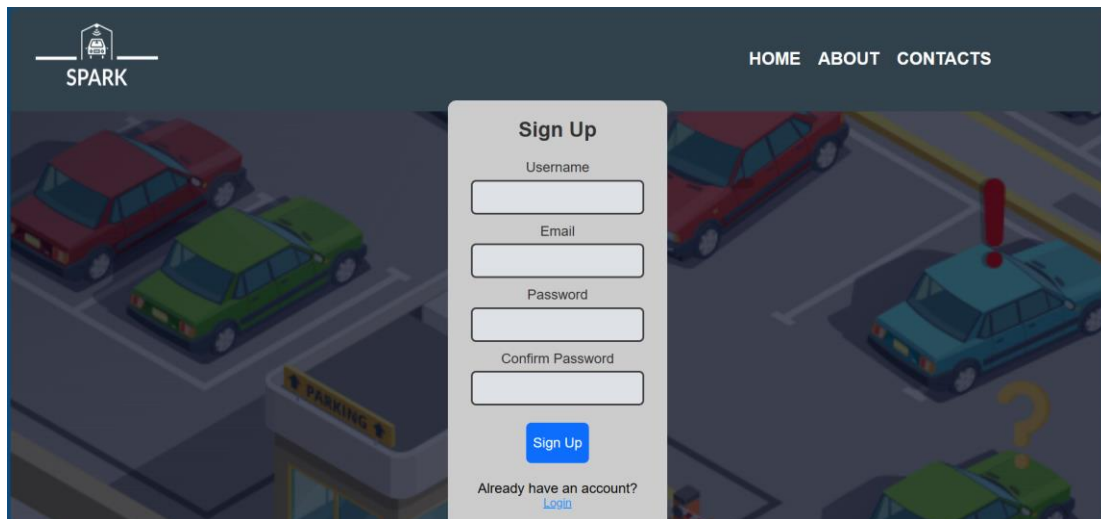
Here the customer can view all their bookings.



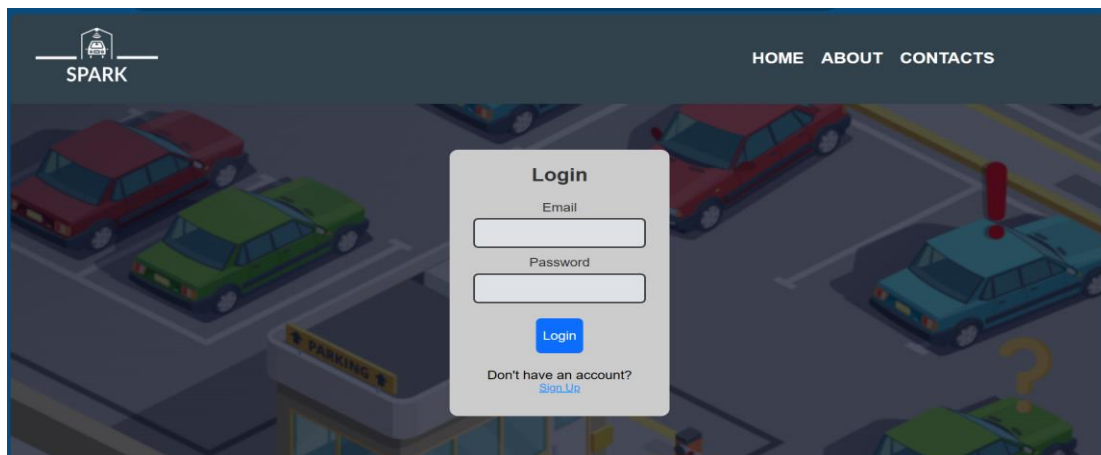
USER MANUAL

MANUAL FOR CUSTOMER:

Here the User gets to create new account by filling up the Registration details.

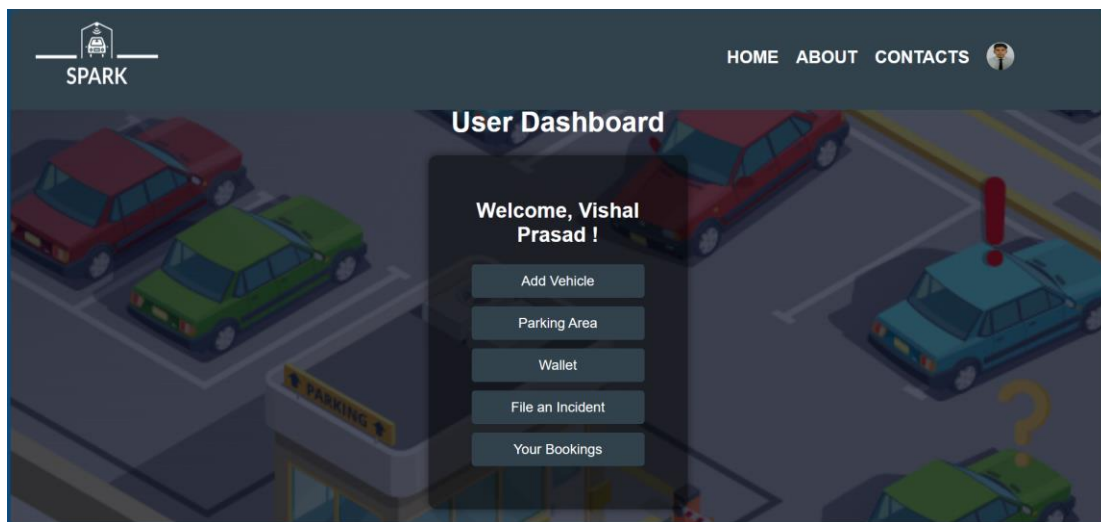


The screenshot displays the SPARK website's Sign Up page. The background features an isometric illustration of a parking lot with several cars. A dark blue header at the top contains the SPARK logo on the left and navigation links 'HOME', 'ABOUT', and 'CONTACTS' on the right. A central white modal box titled 'Sign Up' contains the following elements: a 'Username' label above a text input field, an 'Email' label above a text input field, a 'Password' label above a text input field, and a 'Confirm Password' label above a text input field. Below these fields is a blue 'Sign Up' button. At the bottom of the modal, the text 'Already have an account?' is followed by a blue 'Login' link.



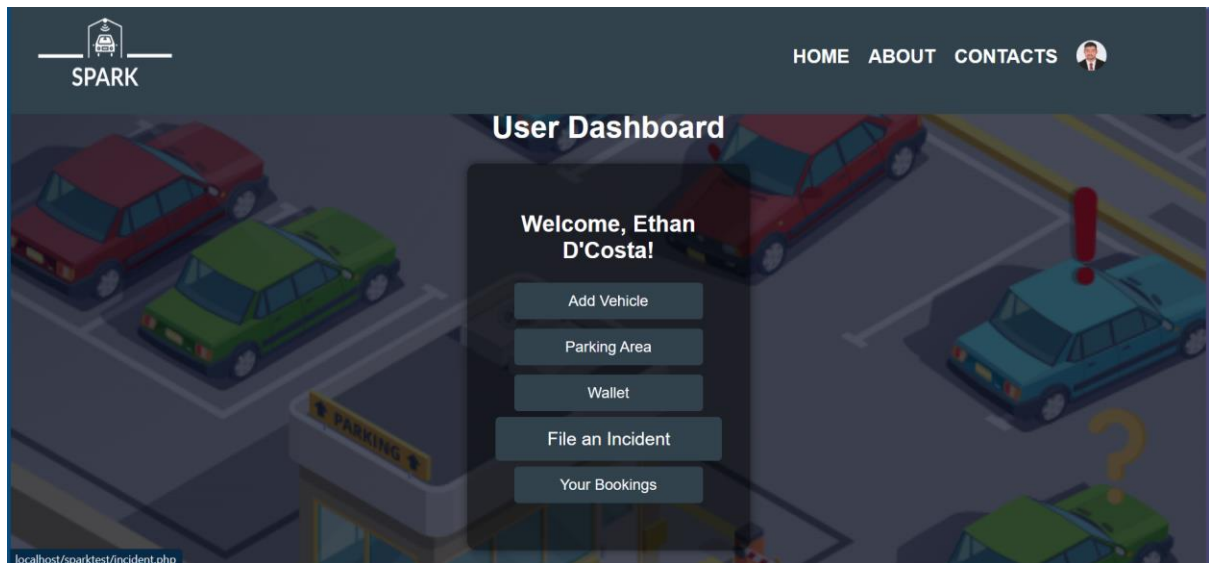
The screenshot displays the SPARK website's Login page. The background features an isometric illustration of a parking lot with several cars. A dark blue header at the top contains the SPARK logo on the left and navigation links 'HOME', 'ABOUT', and 'CONTACTS' on the right. A central white modal box titled 'Login' contains the following elements: an 'Email' label above a text input field, and a 'Password' label above a text input field. Below these fields is a blue 'Login' button. At the bottom of the modal, the text 'Don't have an account?' is followed by a blue 'Sign Up' link.

This Page provides users with options to either Login with an already existing or Sign Up for a new account. After logging-in, the user is taken to the home page which features a user dashboard containing various options.

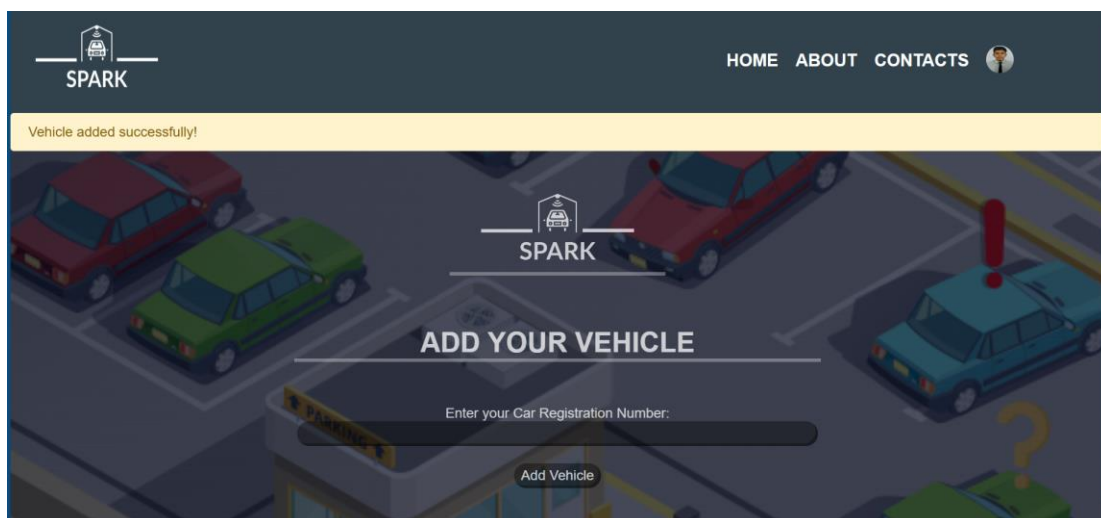
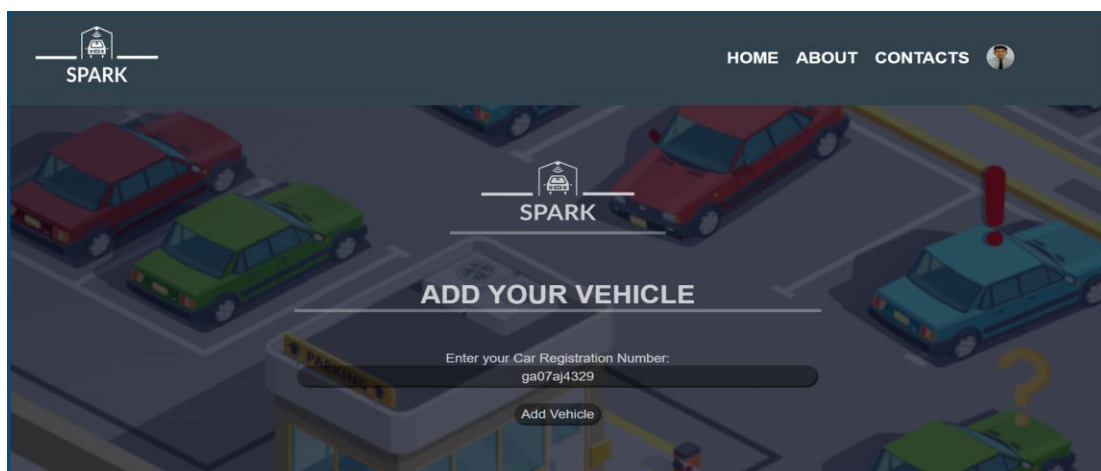


The Home-page contains a dashboard for the user with options to add vehicle, view parking areas affiliated with Spark, check wallet balance, file an incident report and view your bookings.

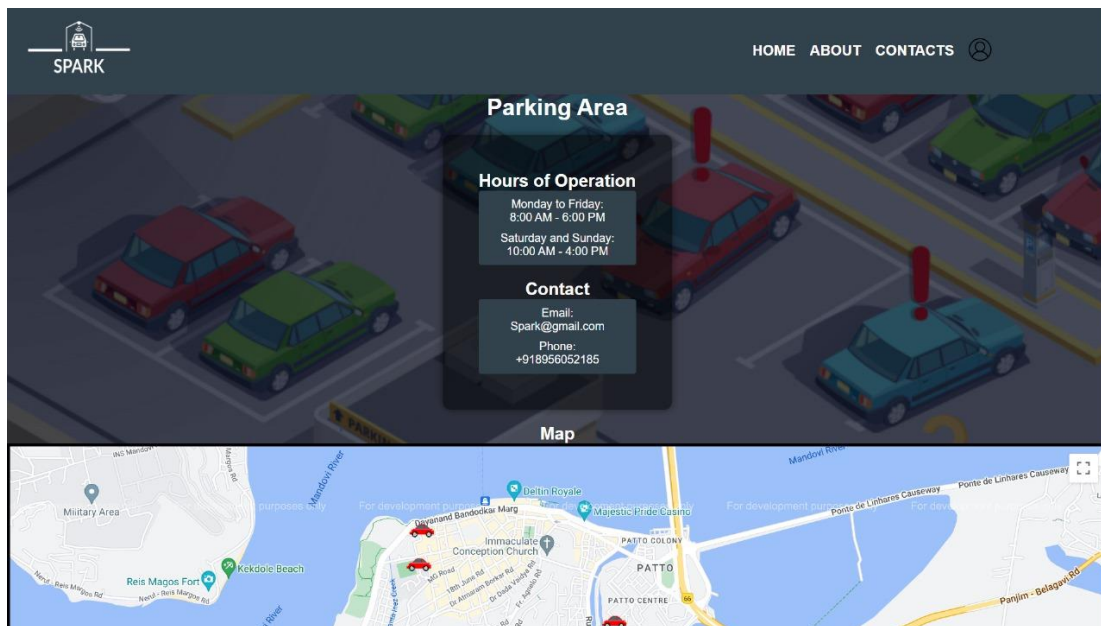
The user can navigate the dashboard and choose the different options.



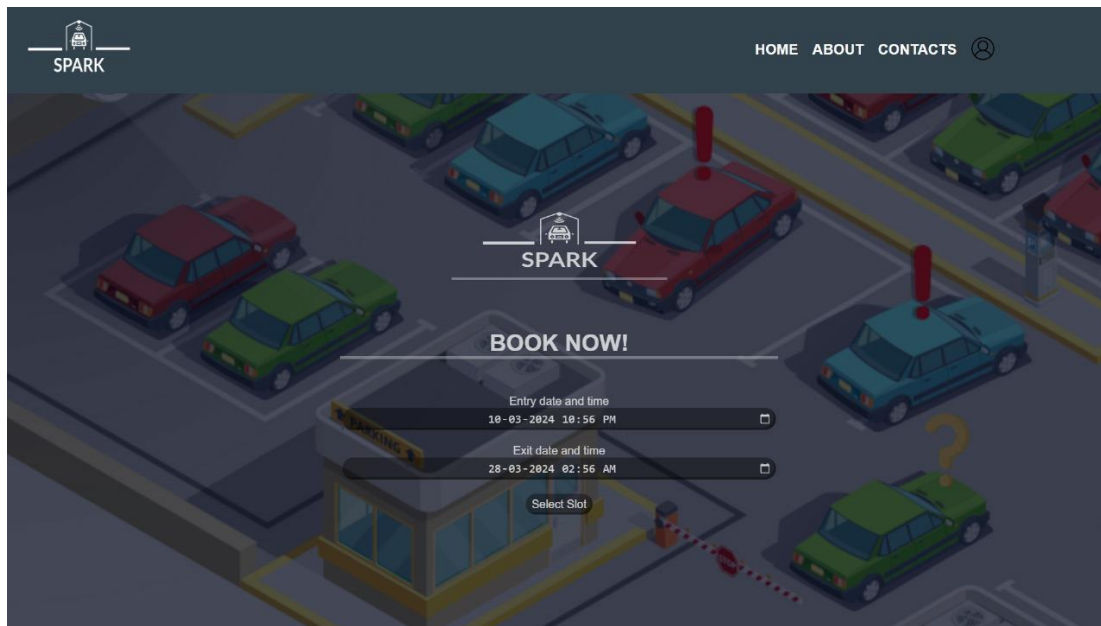
The user selects the add vehicle option in the dashboard so that they can add their vehicles by typing their license plate. This allows us to track the user.



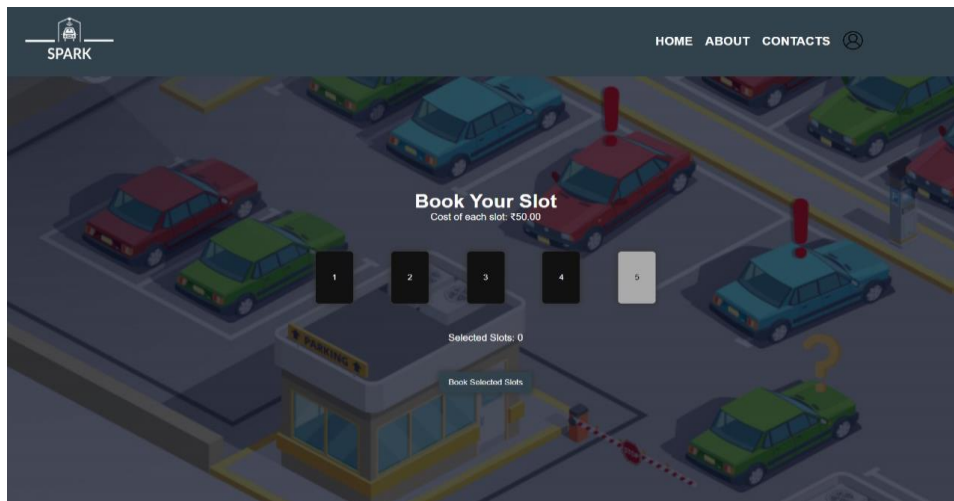
After this the user selects a parking area where he wants to reserve a slot in by clicking on the red car icon on the map.



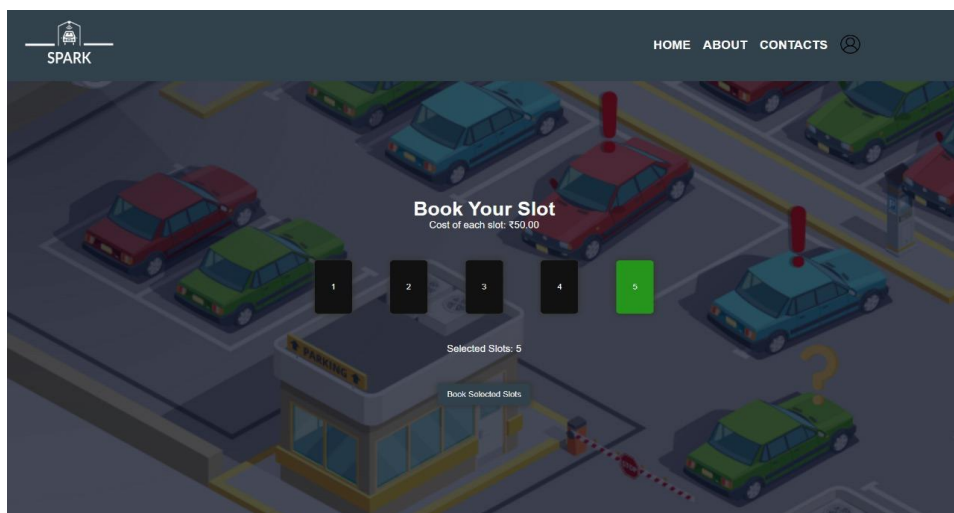
After clicking on a parking area, the user is redirected to the reservation page.



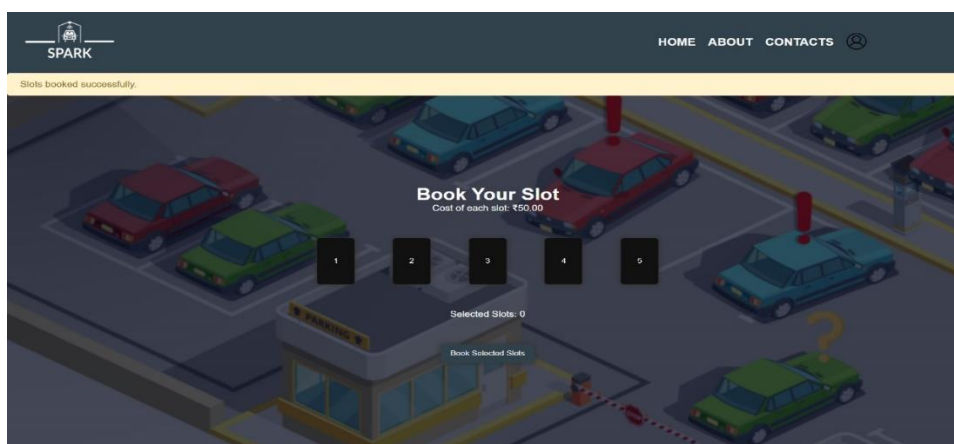
After filling the booking form, they are then redirected to the slot booking page.



The black slots indicate slots which have been occupied and the grey slots indicate unoccupied slots.

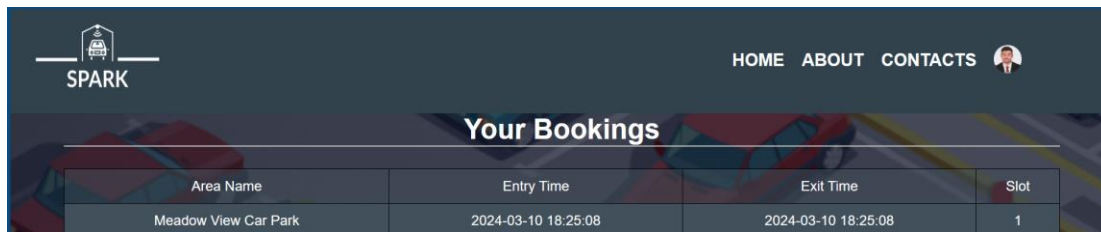


The green slot indicates the slot or slots that the user has selected.



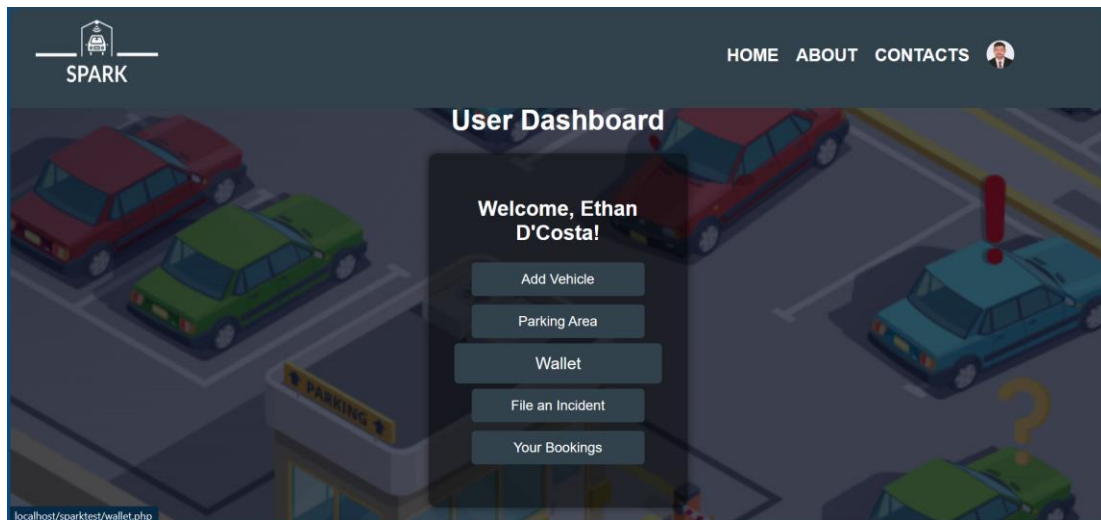
The user clicks on the book selected slots button and the slot is successfully booked with money deducted from the wallet.

The recent bookings are displayed in the ‘Your Booking’ option on the dashboard.

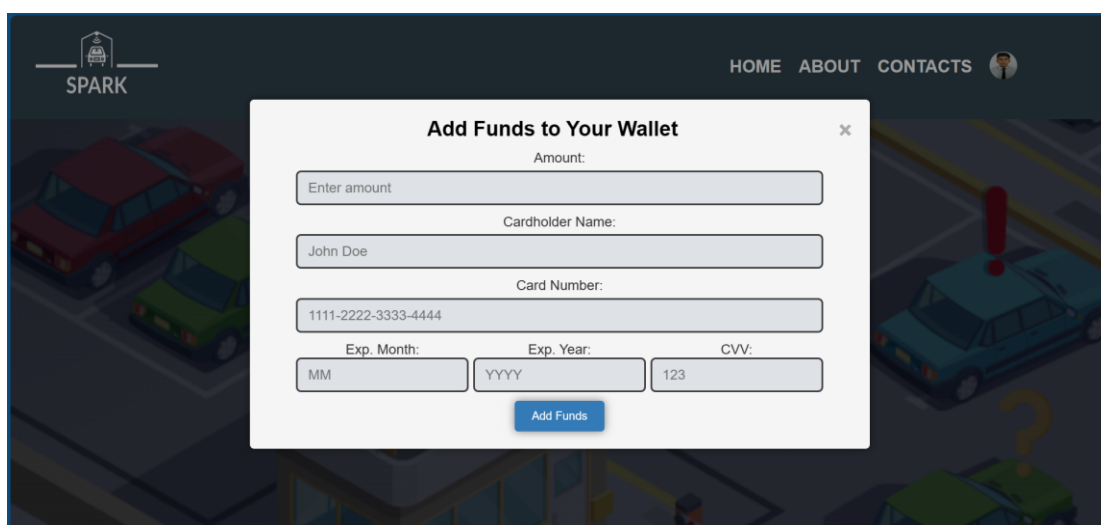
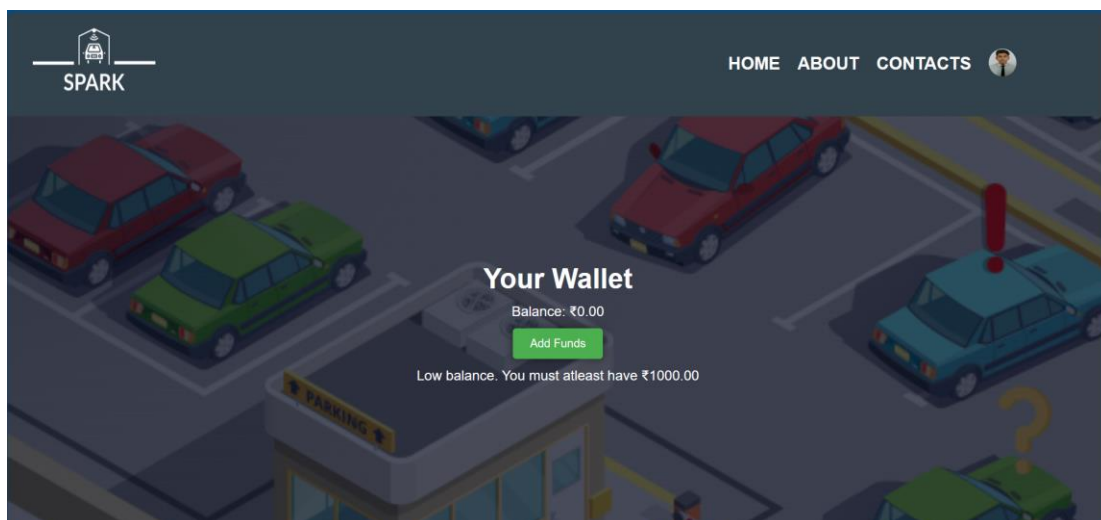


| Your Bookings | | | |
|----------------------|---------------------|---------------------|------|
| Area Name | Entry Time | Exit Time | Slot |
| Meadow View Car Park | 2024-03-10 18:25:08 | 2024-03-10 18:25:08 | 1 |

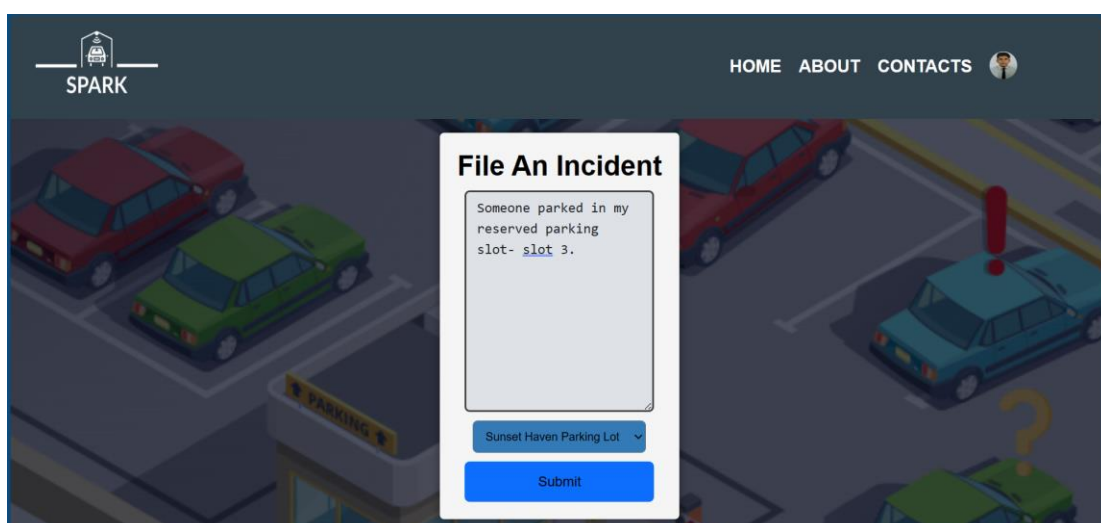
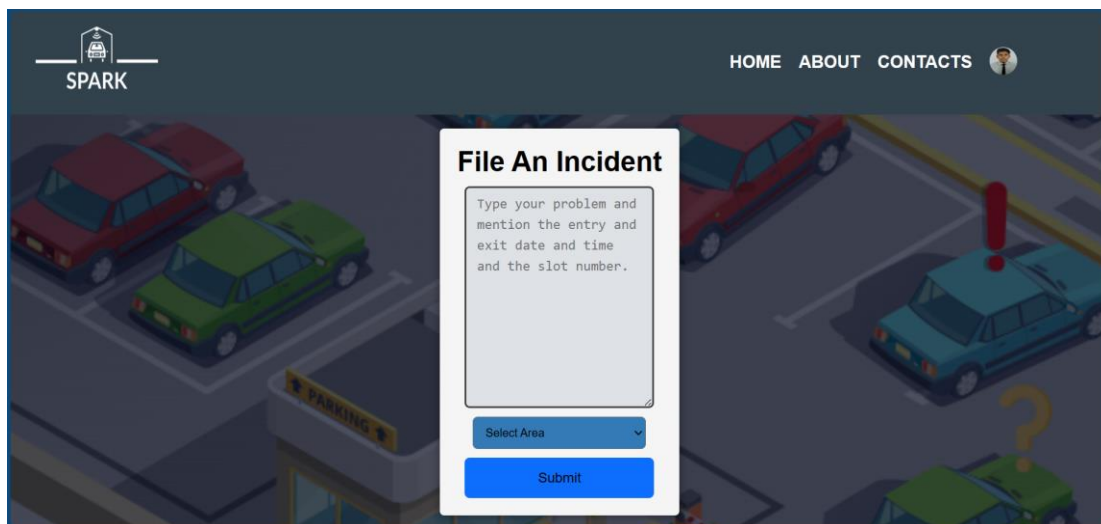
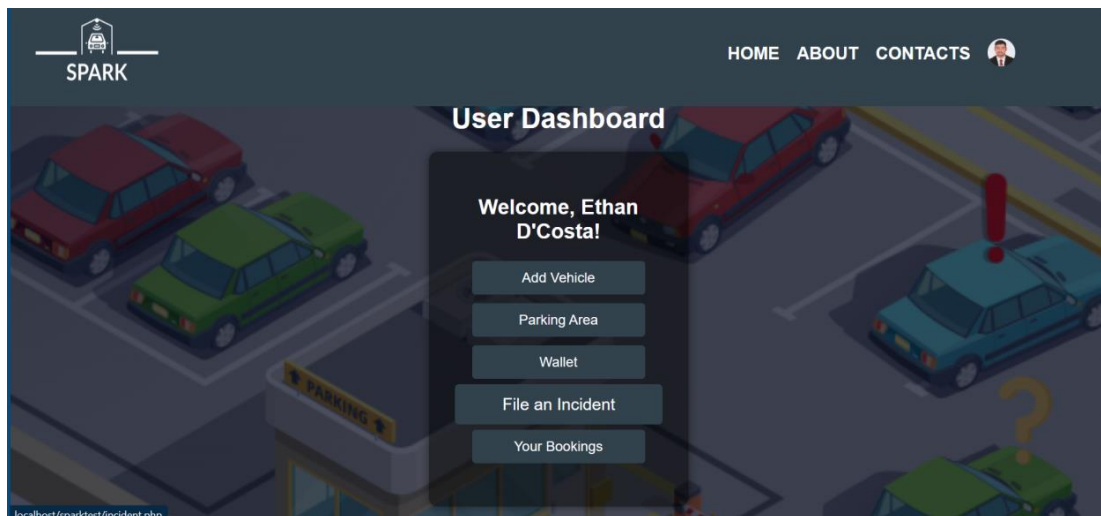
The user can check his wallet balance in the 'Wallet' option on the dashboard.



Here the Customer can view their wallet balance and add funds to it. A minimum of 1000 is required to enter the parking area and reserve the slot.



If any incident has occurred in a parking area the customer can file an incident report in the parking area selected in the 'File an incident' option on the dashboard.



Incident reported successfully.

File An Incident

Type your problem and mention the entry and exit date and time and the slot number.

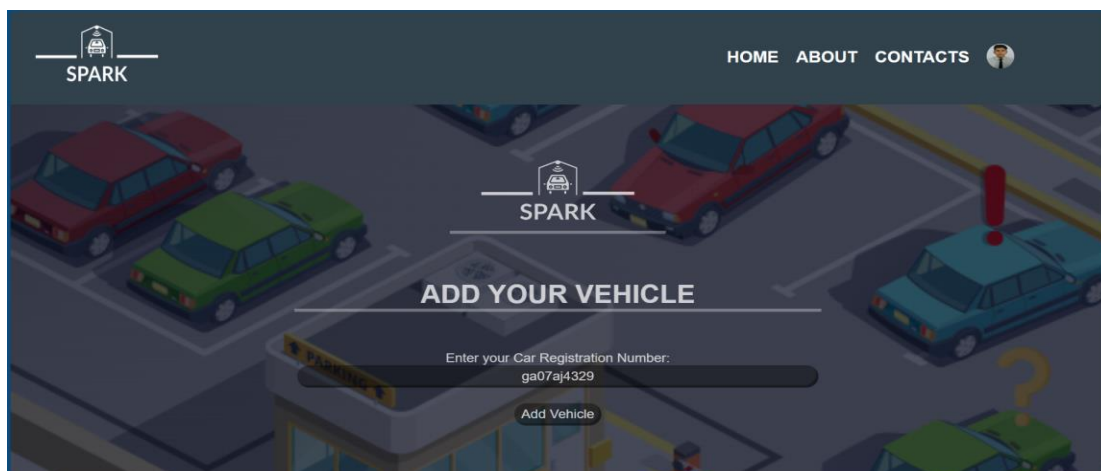
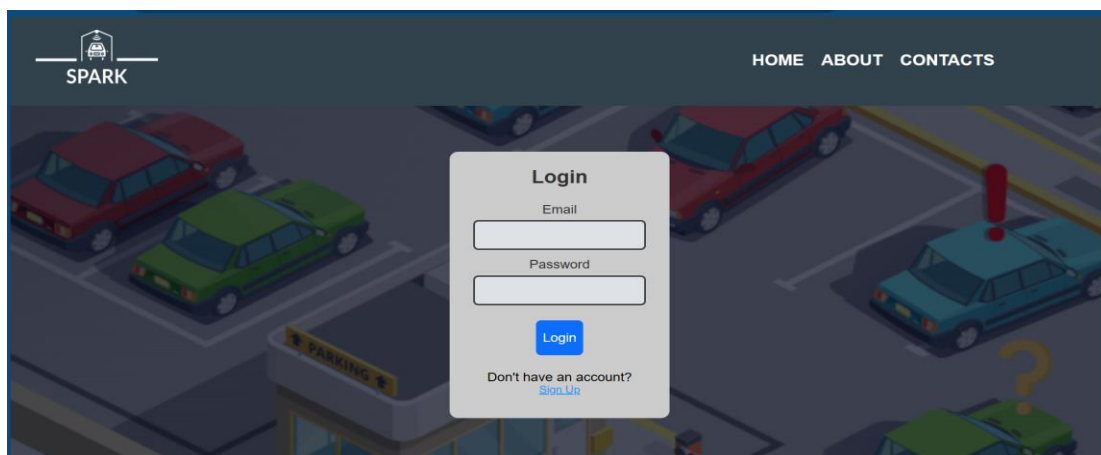
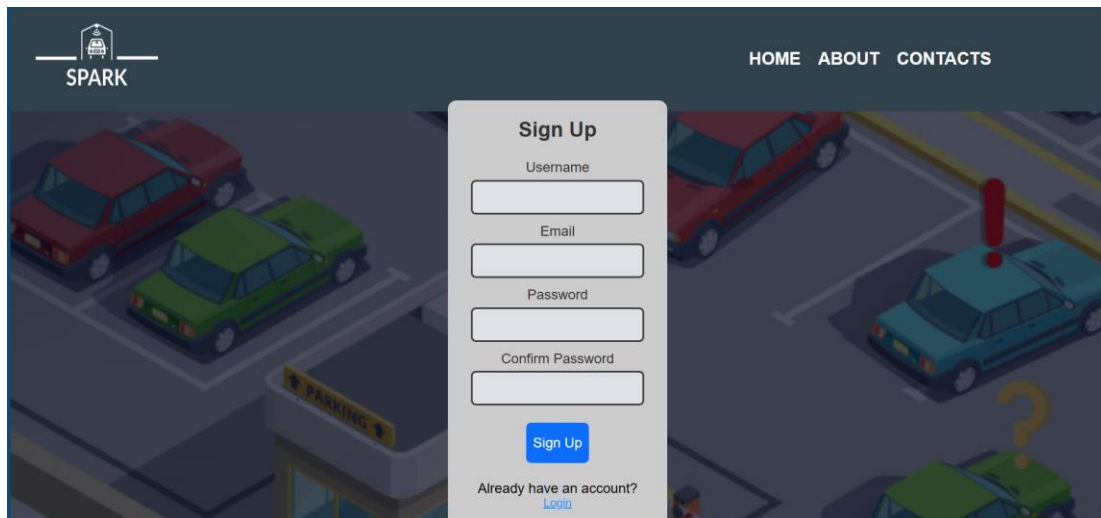
Select Area

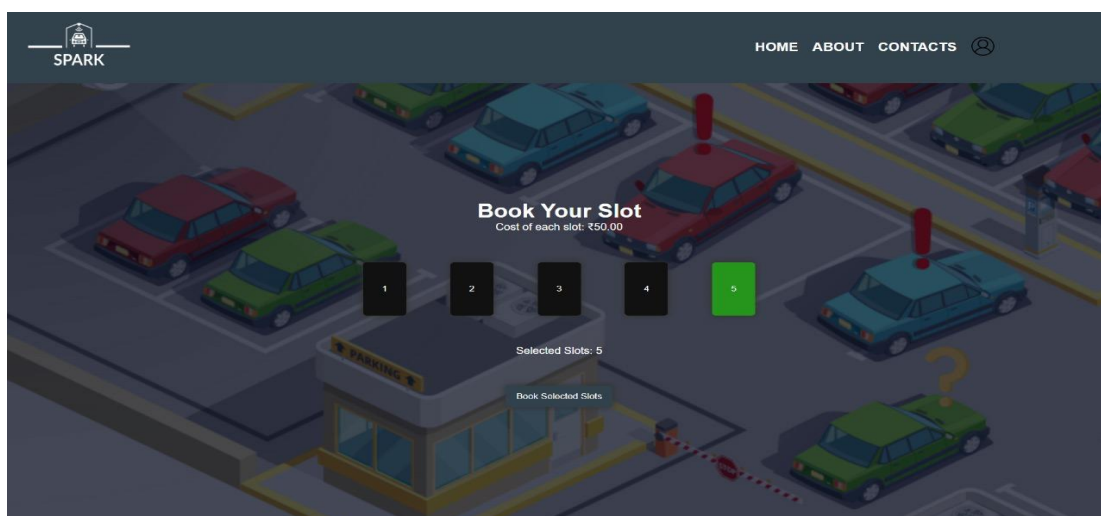
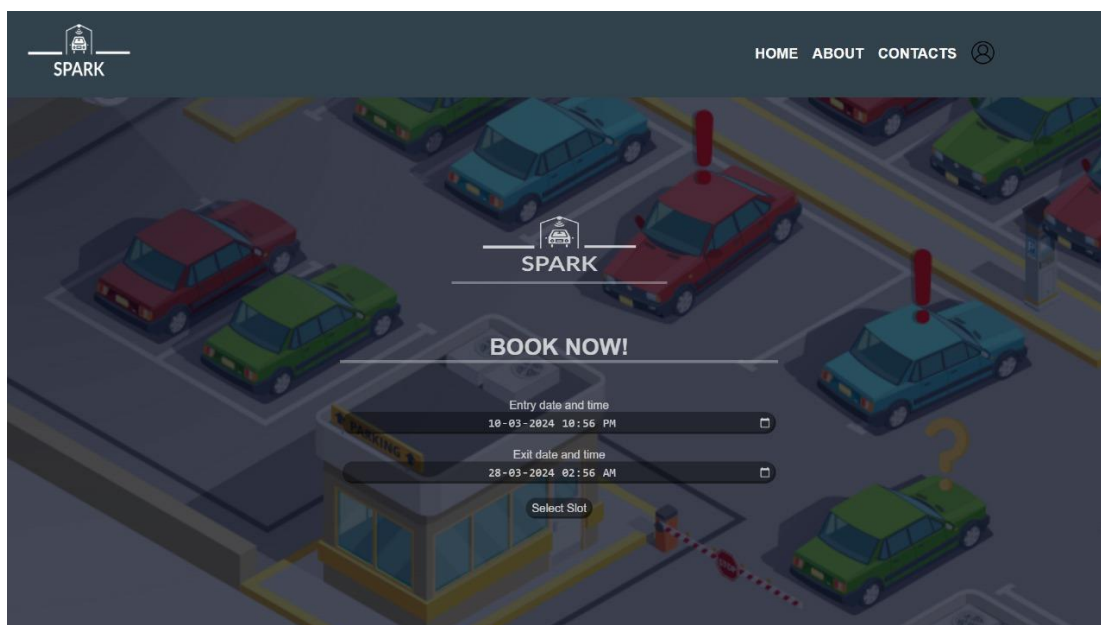
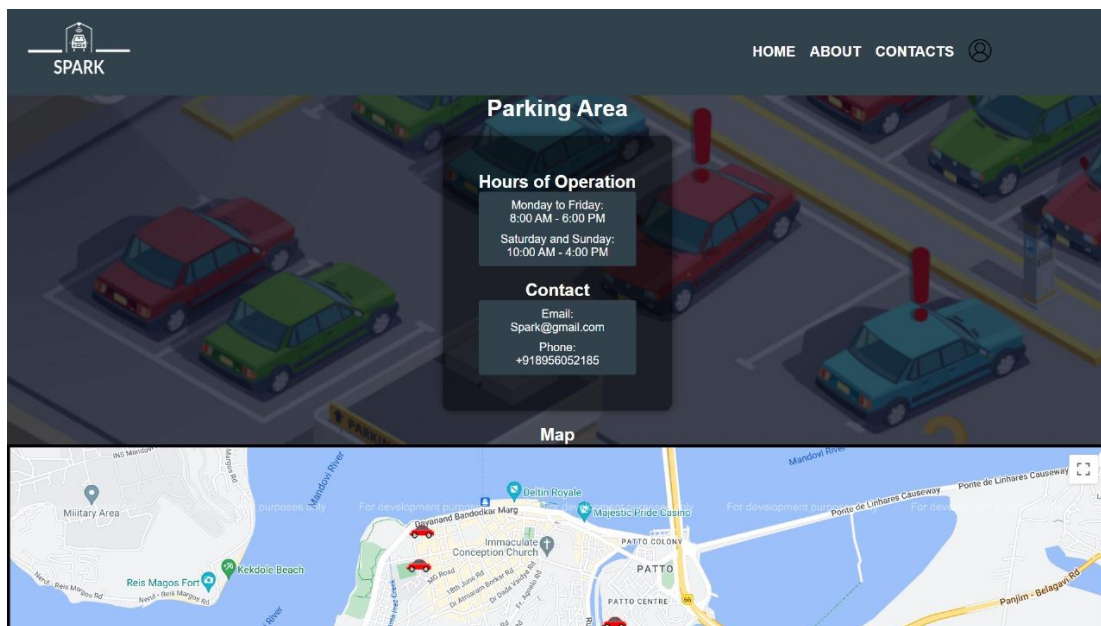
Submit

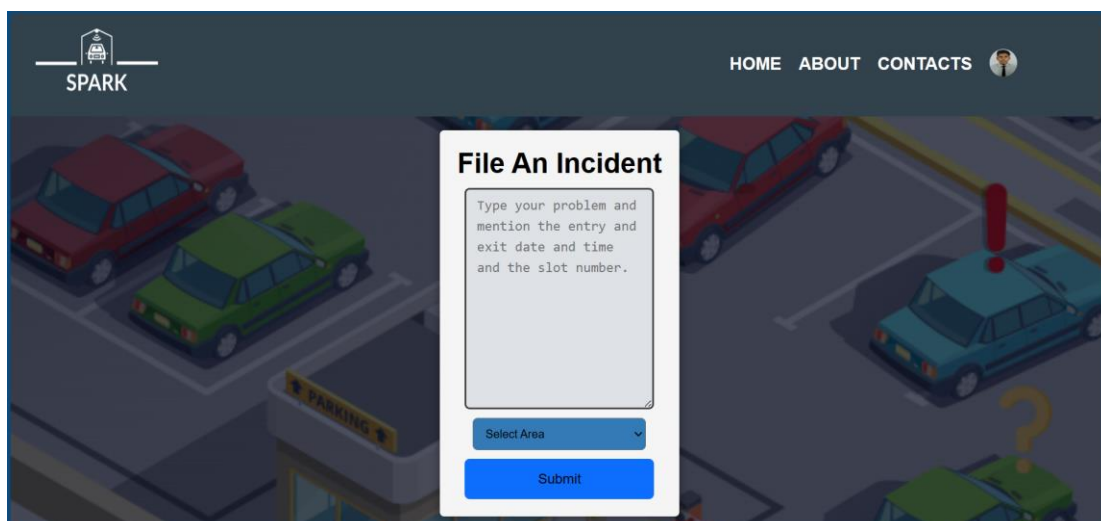
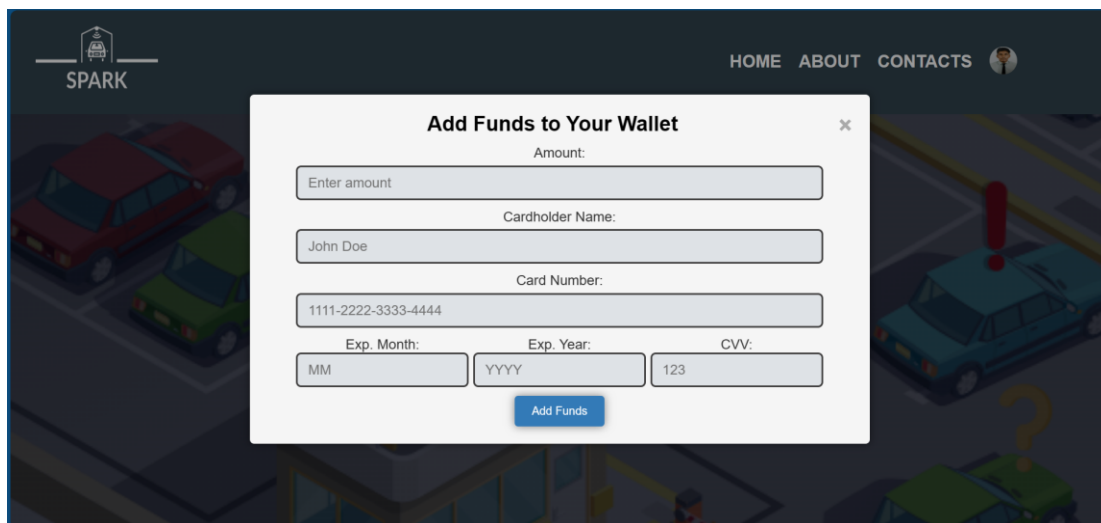
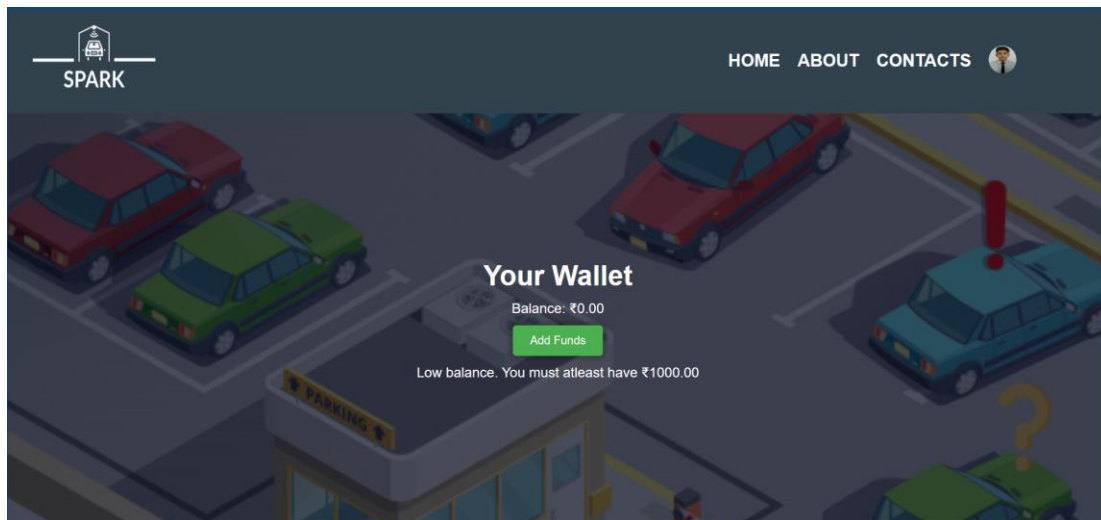
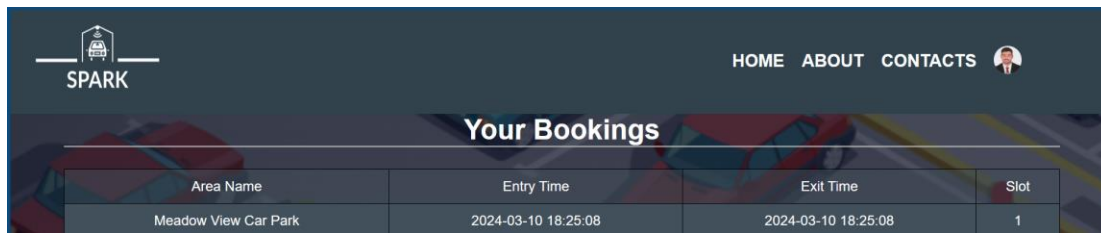
FUTURE **ENHANCEMENTS**

- To make an app so that it's much easier for the users to use.
- To add the following features:
 - 1) Rate the quality of the services offered at each parking area.
 - 2) Provide the customer with Online Payment Methods like GPAY which will aid in easing the transactions.
 - 3) Add more parking areas across the city.

APPENDICES







GANTT CHART

| TASK: | START: | END: | DA VS: | J U L | A U G | S E P | O C T | N O V | D E C | J A N | F E B | M A R |
|-------------------------------------|-------------------|-------------------|-----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| “SPARK” | 16/07/2023 | 10/03/2024 | 238 | | | | | | | | | |
| ANALYSIS: | 15/07/2023 | 10/08/2023 | 27 | | | | | | | | | |
| Scope of Project | 15/07/2023 | 28/08/2023 | 14 | | | | | | | | | |
| Software Requirement Specifications | 18/07/2023 | 05/08/2023 | 19 | | | | | | | | | |
| Feasibility Study | 21/07/2023 | 05/08/2023 | 16 | | | | | | | | | |
| Analysis Complete | 25-07-2022 | 10-08-2022 | 17 | | | | | | | | | |
| DESIGN: | 20/08/2023 | 22/09/2023 | 64 | | | | | | | | | |
| Entity Relation Diagram | 20/08/2023 | 12/09/2023 | 24 | | | | | | | | | |
| Class Diagram | 24/08/2023 | 01/09/2023 | 09 | | | | | | | | | |
| Use-Case Diagram | 22/08/2023 | 05/09/2023 | 15 | | | | | | | | | |
| Activity Diagram | 26/08/2023 | 09/09/2023 | 15 | | | | | | | | | |
| Design complete | 26/08/2023 | 22/10/2023 | 57 | | | | | | | | | |
| IMPLEMENTATION: | 06/10/2023 | 15/12/2024 | 70 | | | | | | | | | |
| User Interface | 06/10/2023 | 05/12/2023 | 60 | | | | | | | | | |
| Coding | 08/10/2023 | 15/12/2023 | 35 | | | | | | | | | |
| Database | 16/10/2023 | 20/11/2023 | 68 | | | | | | | | | |
| Hardware Assembly | 06/10/2024 | 05/12/2024 | 60 | | | | | | | | | |
| Software Setup | 06/10/2024 | 15/12/2024 | 70 | | | | | | | | | |
| TESTING: | 15/01/2024 | 04/03/2024 | 48 | | | | | | | | | |
| Unit Testing | 15/01/2024 | 02-02-2023 | 18 | | | | | | | | | |
| Validation Testing | 07/02/2024 | 27/02/2024 | 20 | | | | | | | | | |
| Integration Testing | 05/02/2024 | 04/03/2024 | 27 | | | | | | | | | |
| Arduino Response | 15/01/2024 | 04/03/2024 | 48 | | | | | | | | | |