Prototyping and Implementation

for

Online Smart Parking system (Web App)

Vellore Institute of Technology

30-April-2021

Table of Contents

Table of Contents **ii**
I) Title of the project **1**
II) Team Size **1**
III)Scope / Objective **1**
IV) Modules **2**
V)Estimation Techniques **3**
1*)Function point calculation* *3*
2*)Intermediate cocomo model* *4*
i*) Parameters* *4*
ii) *Results* *5*
VI) prototype **6**
VII)Use cases **9**
i)Use cases identified *9*
ii)Use cases(completed) screenshot *10*
iii)Tools or platform used *19*
VIII)Challenges **20**

I. Title of the project

Online Smart Parking system (web application)

II. Team Size

4 members

III. Scope or Objective of the project

Our proposed project is an online parking booking system which provides customers with a way of reserving a parking space online. It helps to overcome the problem of finding a place for parking in commercial areas that unnecessarily consumes time. Hence the project will be a web-based reservation system where users can view various parking areas and select the space to view whether the space is available or not. If the booking space is available then he/she can book it for a specific time slot. Users can also make an online payment for booking. After making the payment users are notified about the booking with a unique parking number. The project aims to avoid the rush in parking slots for choosing a space and reduce the number of cars waiting or searching for parking venues. The application shall run on different platforms. The application forms an individual parking space for each vehicle before their entry into that area/zone. The application asks the user to choose their desired space for parking. The application mainly checks this capability. The project is first developed on a Web-based application like a website, but later it can be developed on Android/IOS applications if the customers want.

IV. Modules

1) LOGIN TO ACCOUNT (ALONG WITH HISTORY OF BOOKINGS

- **Admin Login:** The system is under the supervision of the admin who manages the bookings
- **User login/registration:** Users have to first register themselves to login into the system.

2) VIEW PARKING

- **Different Parking areas outlook:** The system will provide three parking areas of different locations.
- **Parking availability check:** The User can click on spaces to view the availability. If space is already booked it'll be marked yellow and therefore the available ones are going to be seen in normal colour.

3) BOOK PARKING

- **The Parking booking online for date and time:** Users can book parking space for their required date and time.
- **Automatic cost calculation:** The system calculates the total cost incurred for parking based on the time that the user has asked for booking.
- **Email on successful parking booking:** When the user is successful in parking the space, the system sends a confirmation and 'thank you' email regarding the space booked.

4) CANCEL BOOKING

- **Parking cancellation:** Users may even cancel their bookings by login into the system anytime.

5) FEEDBACK

- **Feedback:** The system has a feedback form, where the user can provide feedback into the system.

6) LOGOUT

V. Estimation Techniques

1. Function point calculation:

2. Intermediate cocomo model

Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR Simple	Average	Complex
Number of User Input	20	○	◉	○
Number of User Outputs	7	◉	○	○
Number of User Inquiries	6	◉	○	○
Number of Files	5	◉	○	○
Number of External Interfaces	7	○	◉	○

Complexity Adjustment Table | FP Calculation

Complexity Adjustment Table

ITEM	COMPLEXITY ADJUSTMENT QUESTIONS	SCALE No Influence 0	1	2	3	4	Essential 5
1	Does the system require reliable backup and recovery?	○	○	◉	○	○	○
2	Are data communications required?	○	○	◉	○	○	○
3	Are there distributed processing functions?	◉	○	○	○	○	○
4	Is performance critical?	○	○	○	○	◉	○
5	Will the system run in an existing, heavily utilized operational environment?	◉	○	○	○	○	○
6	Does the system require on-line data entry?	○	○	○	○	○	◉
7	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	◉	○	○	○	○	○
8	Are the master files updated on-line?	◉	○	○	○	○	○
9	Are the inputs, outputs, files or inquiries complex?	○	○	◉	○	○	○
10	Is the internal processing complex?	○	○	◉	○	○	○
11	Is the code to be designed reusable?	○	◉	○	○	○	○
12	Are conversion and installation included in the design?	◉	○	○	○	○	○
13	Is the system designed for multiple installations in different organizations?	◉	○	○	○	○	○
14	Is the application designed to facilitate change and ease of use by the user?	○	○	○	◉	○	○

Domain Characteristic Table | FP Calculation

FP Calculation

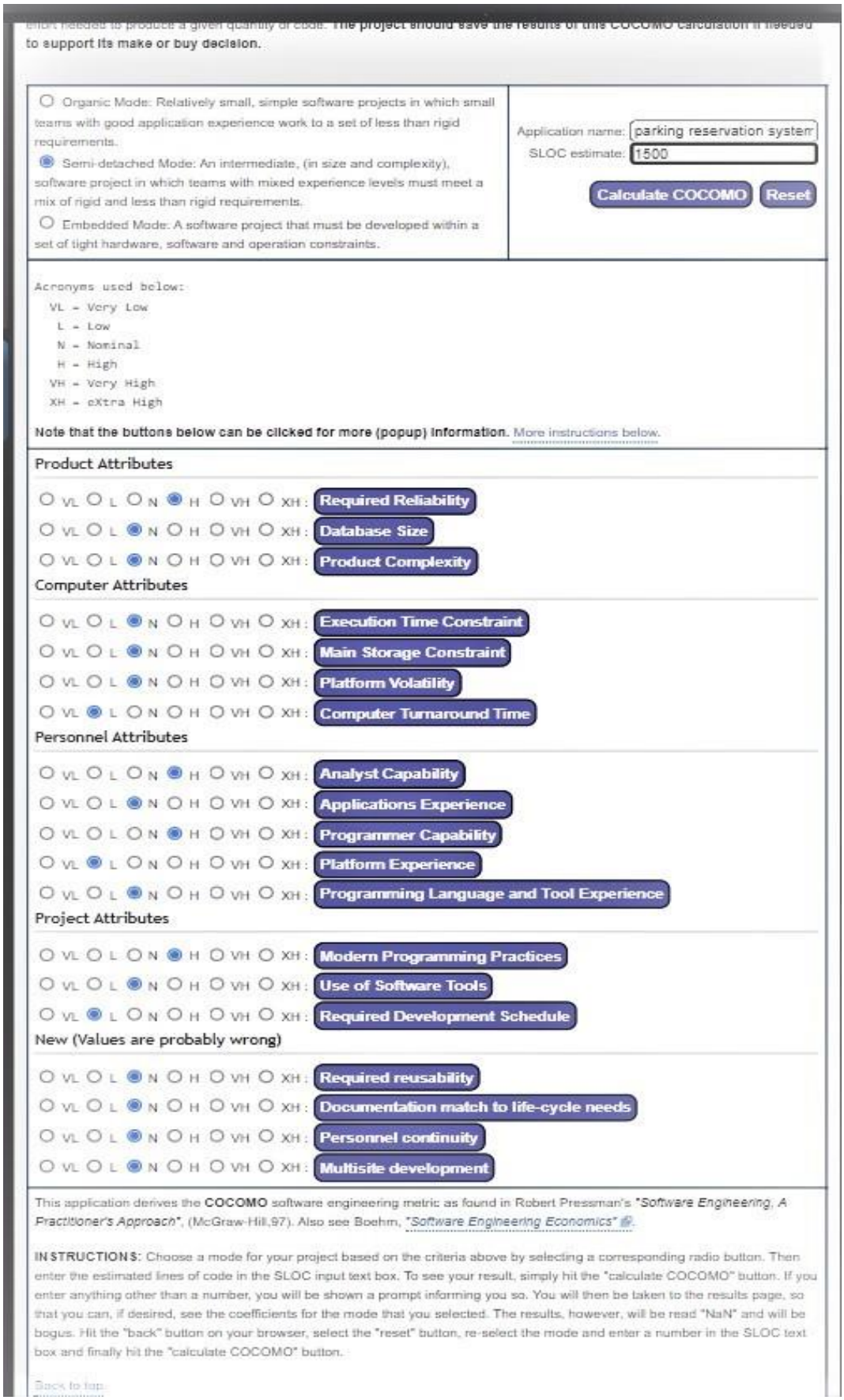
NOTE: For any updates made on any of the entries, always click the 'Calculate Function Points' button to recalculate function points value.

Reset / Clear all form entries Calculate Function Points

RESULT	
PROJECT FUNCTION POINTS	180.6

Top of Page | Domain Characteristic Table | Complexity Adjustment Table

- **Parameters :**


to support its make or buy decision.
Organic Mode: Relatively small, simple software projects in which small teams with good application experience work to a set of less than rigid requirements.
Semi-detached Mode: An intermediate, (in size and complexity), software project in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.
Embedded Mode: A software project that must be developed within a set of tight hardware, software and operation constraints.
Application name: parking reservation system
SLOC estimate: 1500
Calculate COCOMO
Reset
Acronyms used below:
VL = Very Low
L = Low
N = Nominal
H = High
VH = Very High
XH = eXtra High
Note that the buttons below can be clicked for more (popup) information. More instructions below.
Product Attributes
Required Reliability
Database Size
Product Complexity
Computer Attributes
Execution Time Constraint
Main Storage Constraint
Platform Volatility
Computer Turnaround Time
Personnel Attributes
Analyst Capability
Applications Experience
Programmer Capability
Platform Experience
Programming Language and Tool Experience
Project Attributes
Modern Programming Practices
Use of Software Tools
Required Development Schedule
New (Values are probably wrong)
Required reusability
Documentation match to life-cycle needs
Personnel continuity
Multisite development
This application derives the COCOMO software engineering metric as found in Robert Pressman's "Software Engineering, A Practitioner's Approach", (McGraw-Hill,97). Also see Boehm, "Software Engineering Economics".
INSTRUCTIONS: Choose a mode for your project based on the criteria above by selecting a corresponding radio button. Then enter the estimated lines of code in the SLOC input text box. To see your result, simply hit the "calculate COCOMO" button. If you enter anything other than a number, you will be shown a prompt informing you so. You will then be taken to the results page, so that you can, if desired, see the coefficients for the mode that you selected. The results, however, will be read "NaN" and will be bogus. Hit the "back" button on your browser, select the "reset" button, re-select the mode and enter a number in the SLOC text box and finally hit the "calculate COCOMO" button.
Back to top

- **Result(estimated time for intermediate cocomo using KLOC):**

COCOMO RESULTS for parking reservation system								
MODE	**"A" variable**	**"B" variable**	**"C" variable**	**"D" variable**	**KLOC**	**EFFORT, (in person-months)**	**DURATION, (in months)**	**STAFFING, (recommended)**
semi-detached	2.3998996541520006	1.12	2.5	0.35	1.500	3.779	3.981	0.949

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm). Note: the decimal separator is a period.

The final estimates are determined in the following manner:

effort = a*KLOC^b, in person-months, with KLOC = lines of code, (in thousands), and:

staffing = effort/duration

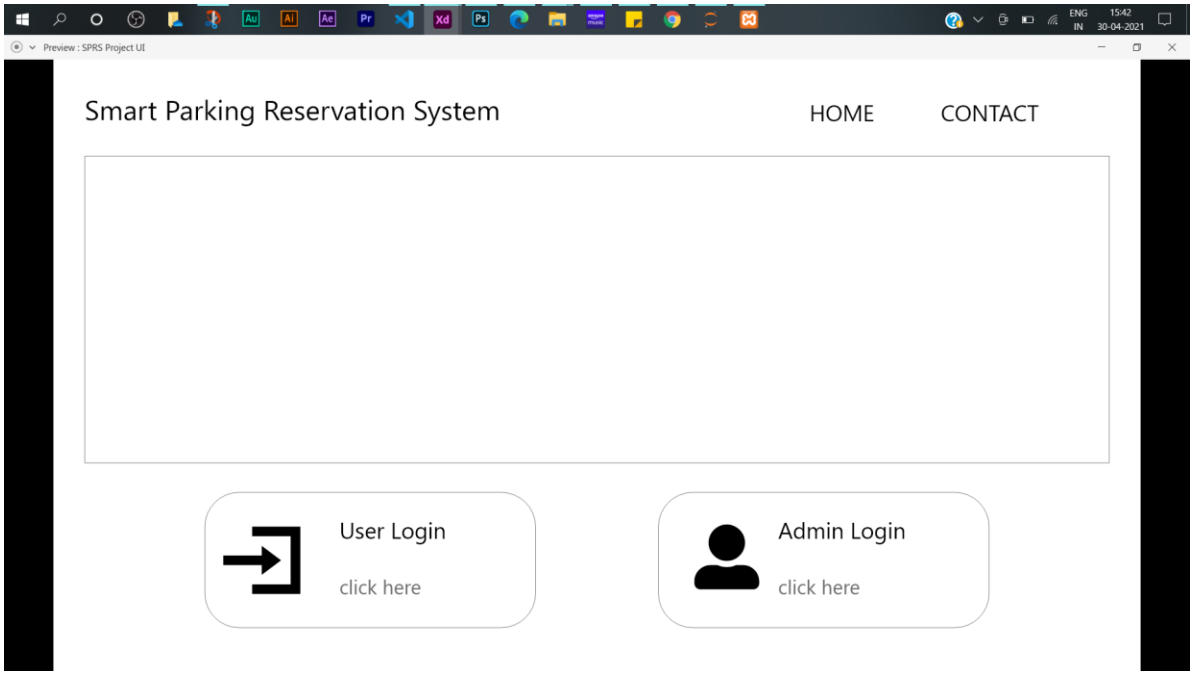
where a has been adjusted by the factors:

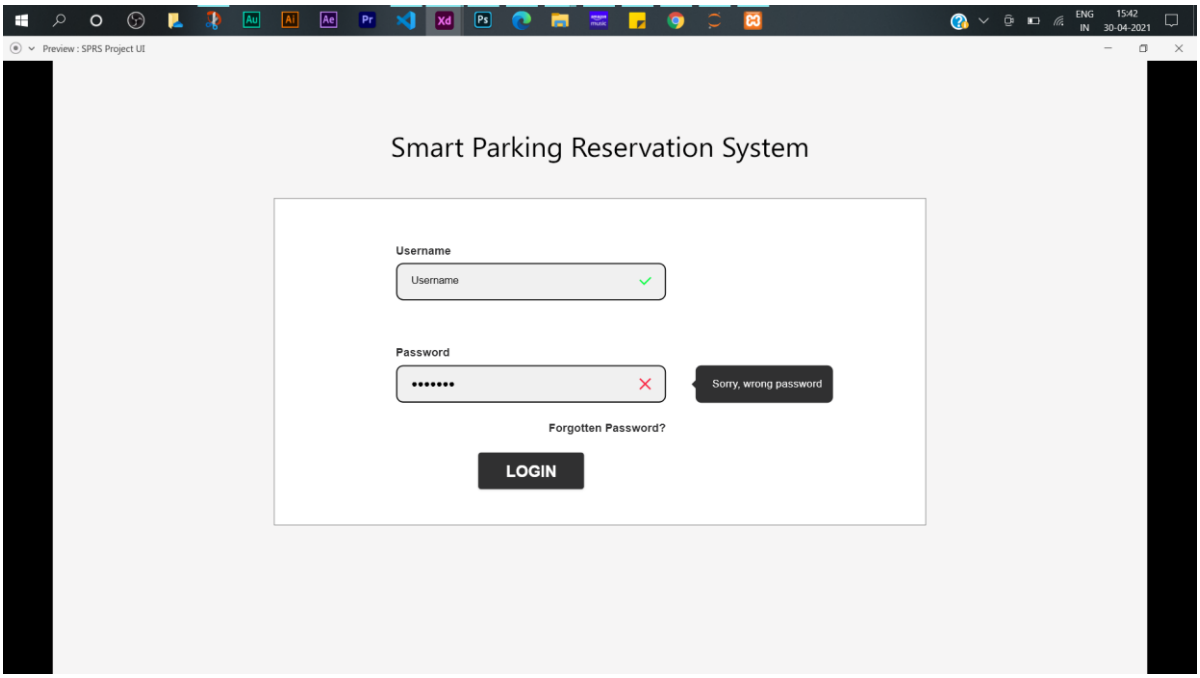
Product Attributes	
Required Reliability	1.15 (H)
Database Size	1.00 (N)
Product Complexity	1.00 (N)
Computer Attributes	
Execution Time Constraint	1.00 (N)
Main Storage Constraint	1.00 (N)
Platform Volatility	1.00 (N)
Computer Turnaround Time	0.87 (L)
Personnel Attributes	
Analyst Capability	0.86 (H)
Applications Experience	1.00 (N)
Programmer Capability	0.86 (H)
Platform Experience	1.10 (L)
Programming Language and Tool Experience	1.00 (N)
Project Attributes	
Modern Programming Practices	0.91 (H)
Use of Software Tools	1.00 (N)
Required Development Schedule	1.08 (L)
New (Values are probably wrong)	
Required reusability	1.00 (N)
Documentation match to life-cycle needs	1.00 (N)
Personnel continuity	1.00 (N)
Multisite development	1.00 (N)

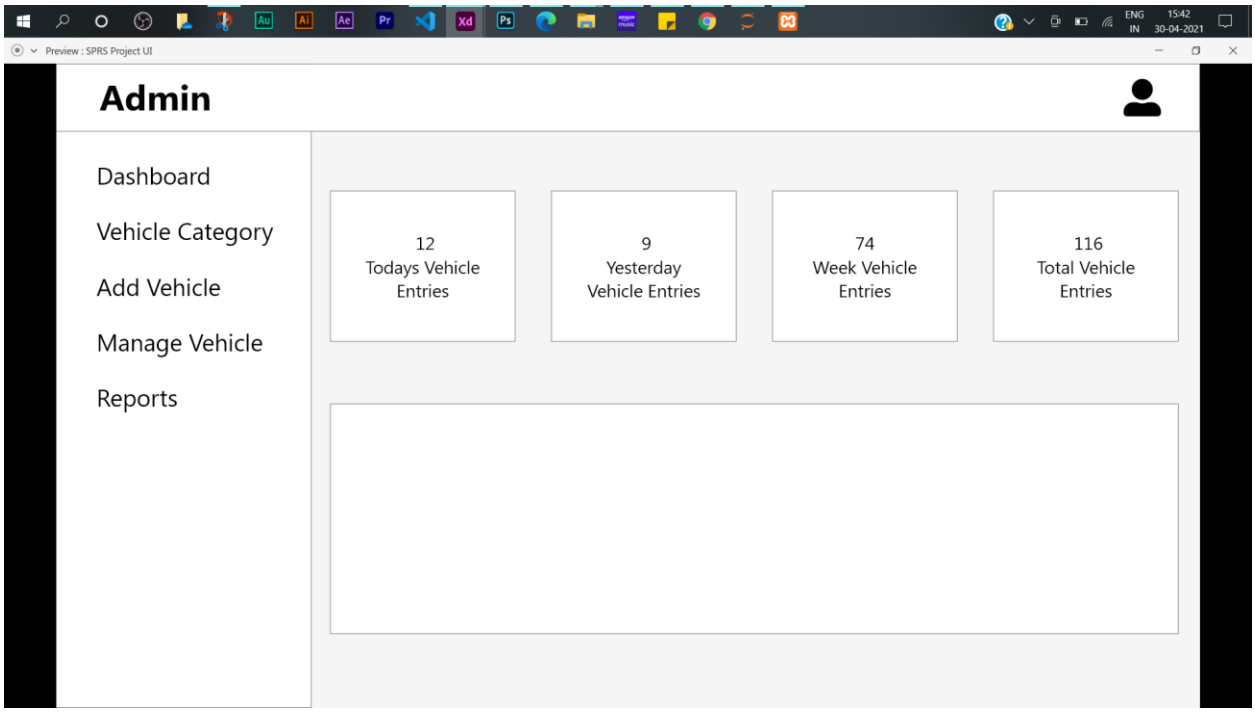
For further reading, see Boehm, "Software Engineering Economics"

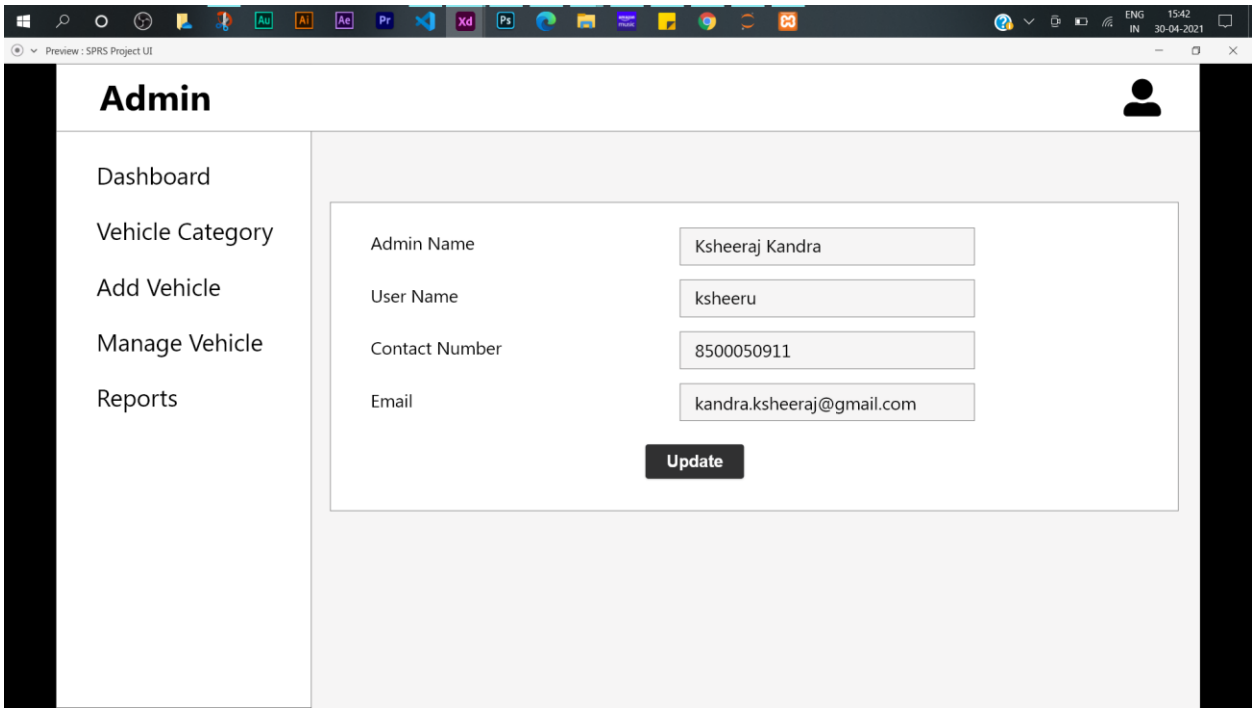
VI. Specify the appropriate prototype for the project

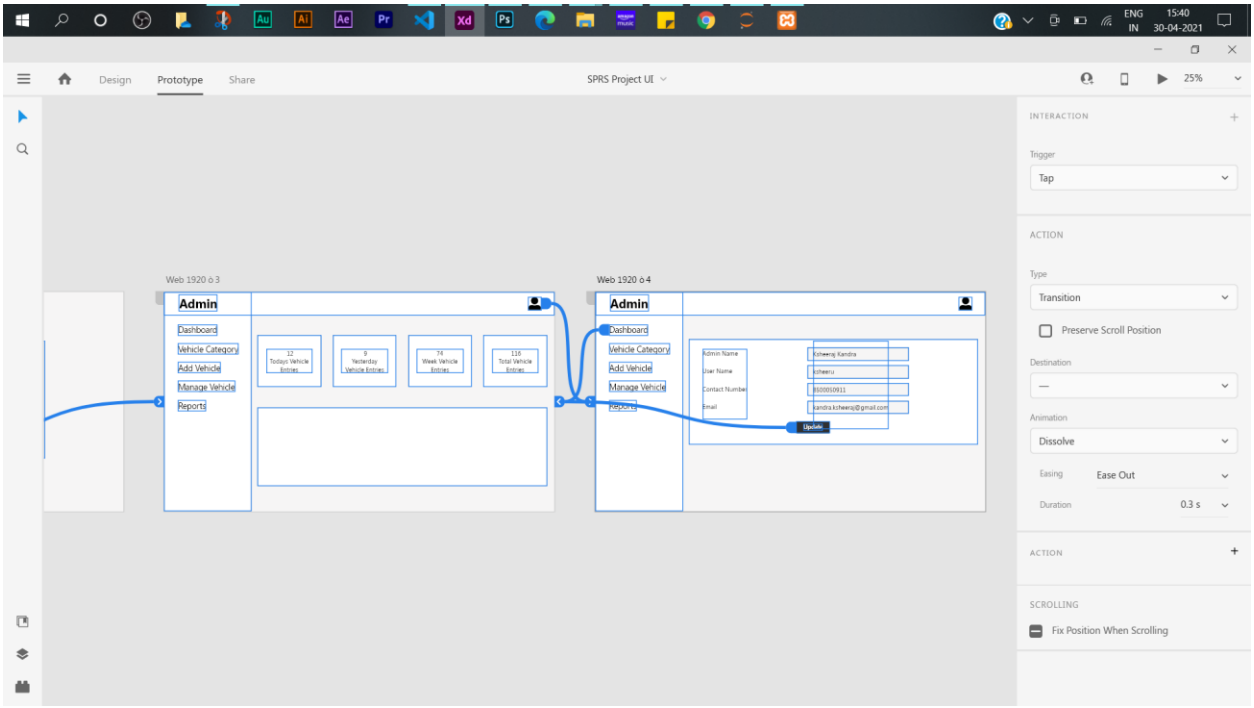
Recommended prototype: Dynamic

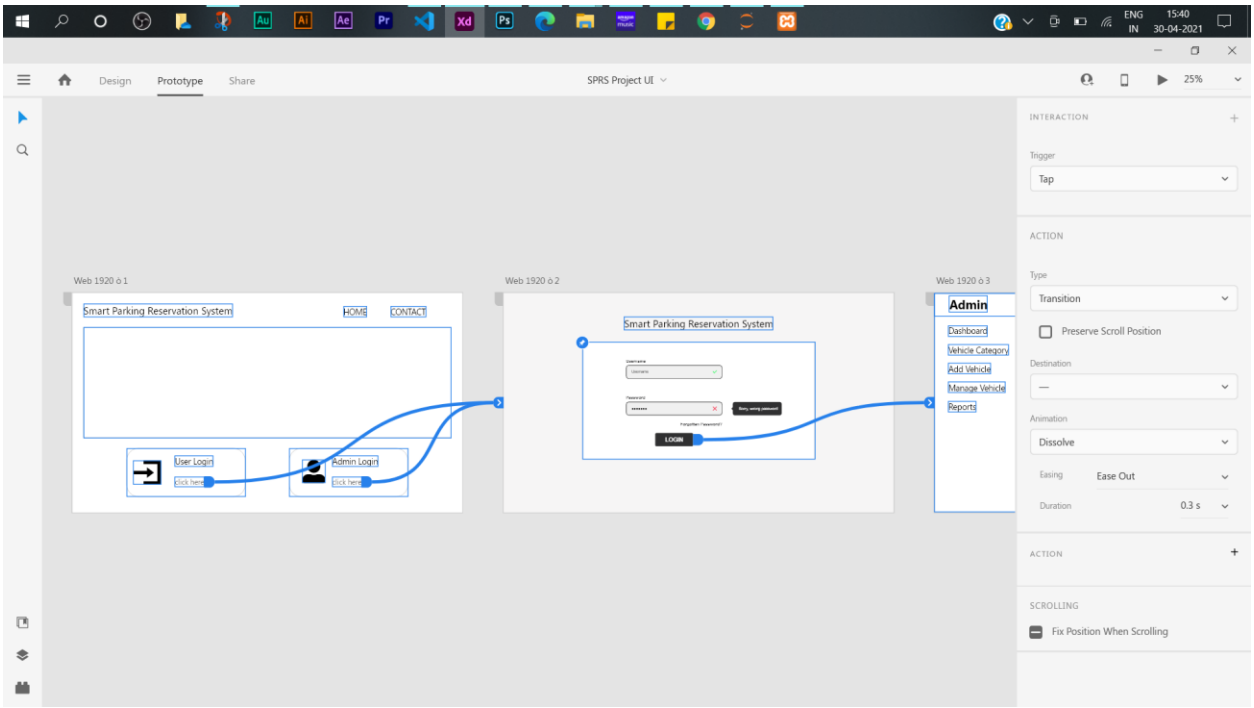



Preview : SPRS Project UI
Smart Parking Reservation System
Username
Username
Password
Sorry, wrong password
Forgotten Password?
LOGIN


Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
12
Todays Vehicle Entries
9
Yesterday Vehicle Entries
74
Week Vehicle Entries
116
Total Vehicle Entries


Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Admin Name
Ksheeraj Kandra
User Name
ksheeru
Contact Number
8500050911
Email
kandra.ksheeraj@gmail.com
Update


Design
Prototype
Share
SPRS Project UI
25%
INTERACTION
Trigger
Tap
ACTION
Type
Transition
Preserve Scroll Position
Destination
Animation
Dissolve
Easing
Ease Out
Duration
0.3 s
ACTION
SCROLLING
Fix Position When Scrolling
Web 1920 ó 3
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Web 1920 ó 4
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Update


Design
Prototype
Share
SPRS Project UI
25%
INTERACTION
Trigger
Tap
ACTION
Type
Transition
Preserve Scroll Position
Destination
Animation
Dissolve
Easing
Ease Out
Duration
0.3 s
ACTION
SCROLLING
Fix Position When Scrolling
Web 1920 ó 1
Smart Parking Reservation System
HOME
CONTACT
User Login
Admin Login
Web 1920 ó 2
Smart Parking Reservation System
LOGIN
Web 1920 ó 3
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports

VII. USE CASES

i) Use cases identified :

User

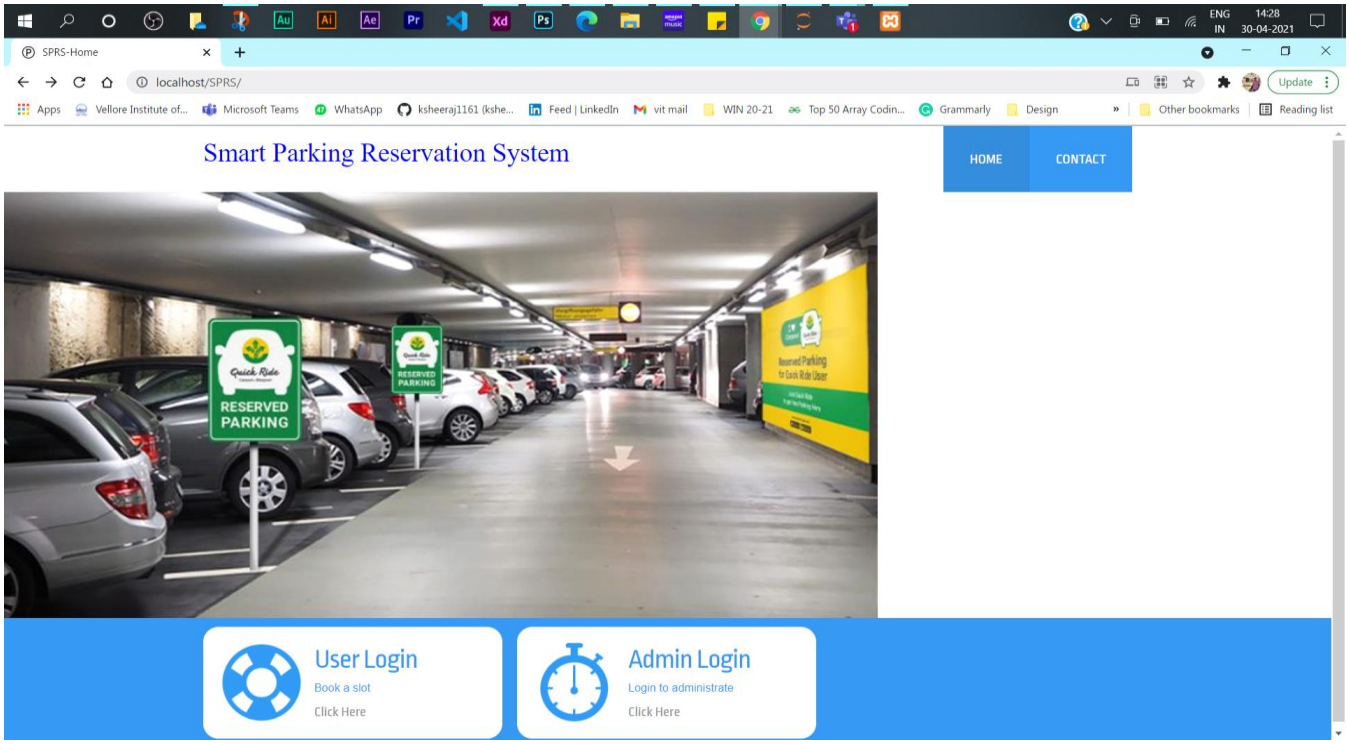
1) login to account
2) view parking
3) book parking
4) cancel booking
5) feedback
6) logout

Admin

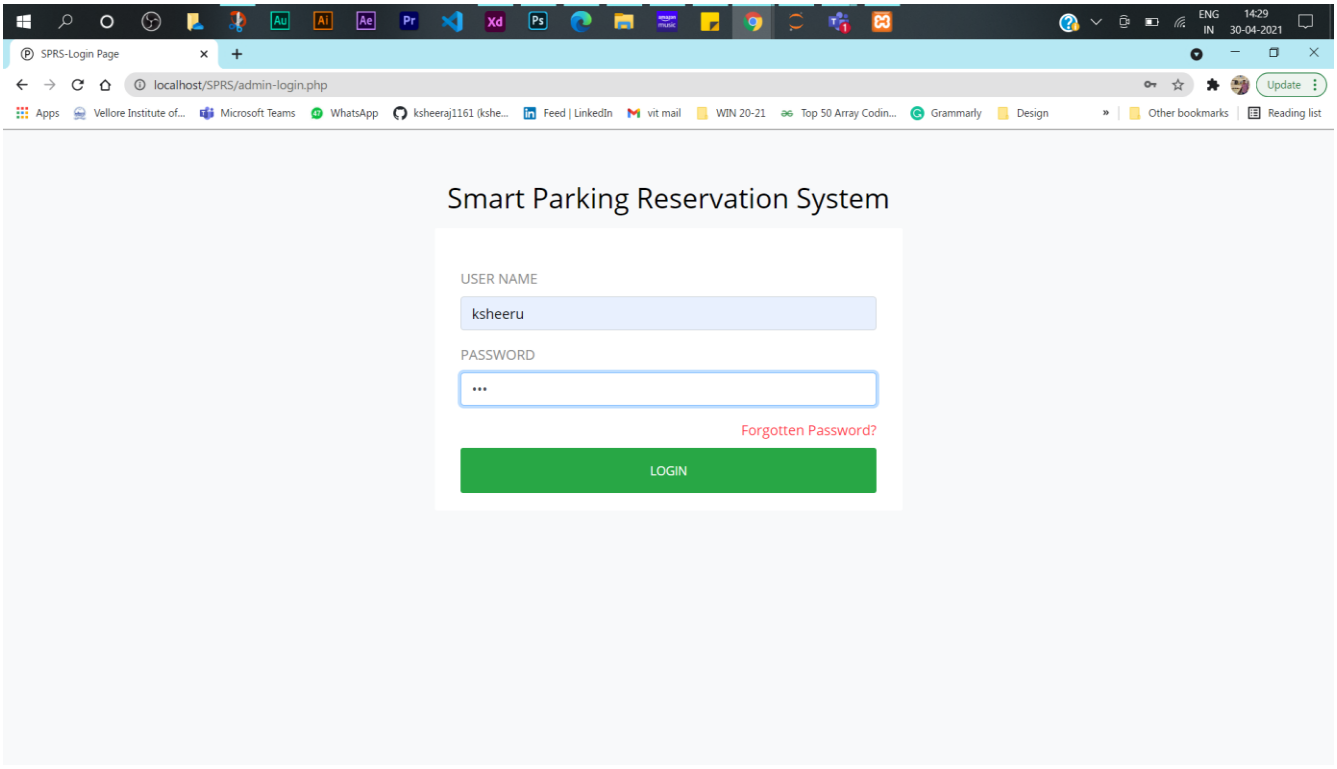
1) admin login
2) search vehicle
3) add vehicle category
4) manage vehicle category
5) add vehicle
6) manage vehicle
7) incoming and outgoing vehicles
8) reports
9) dashboard
10) Profile update
11) print receipt
12) logout

ii) Use cases completed :

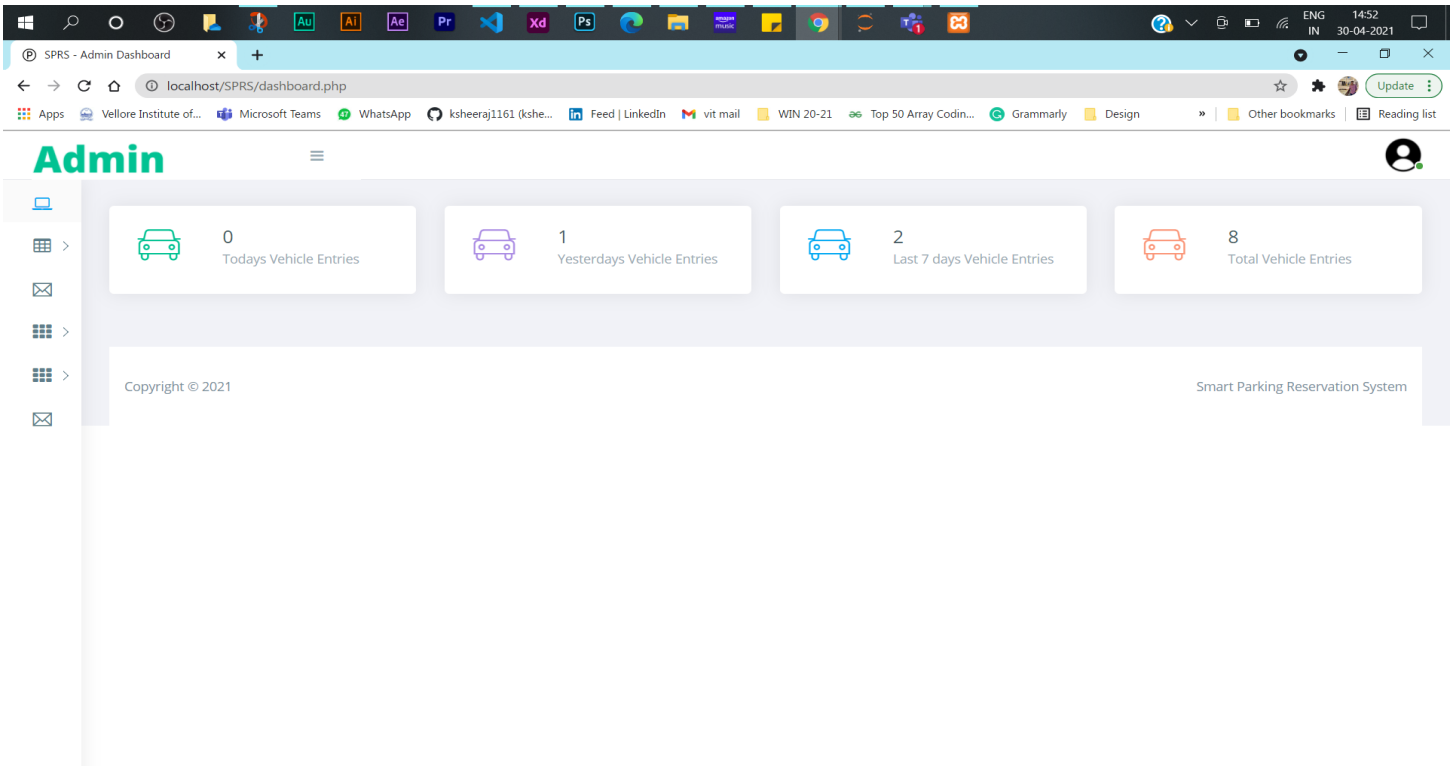
- **Home page**

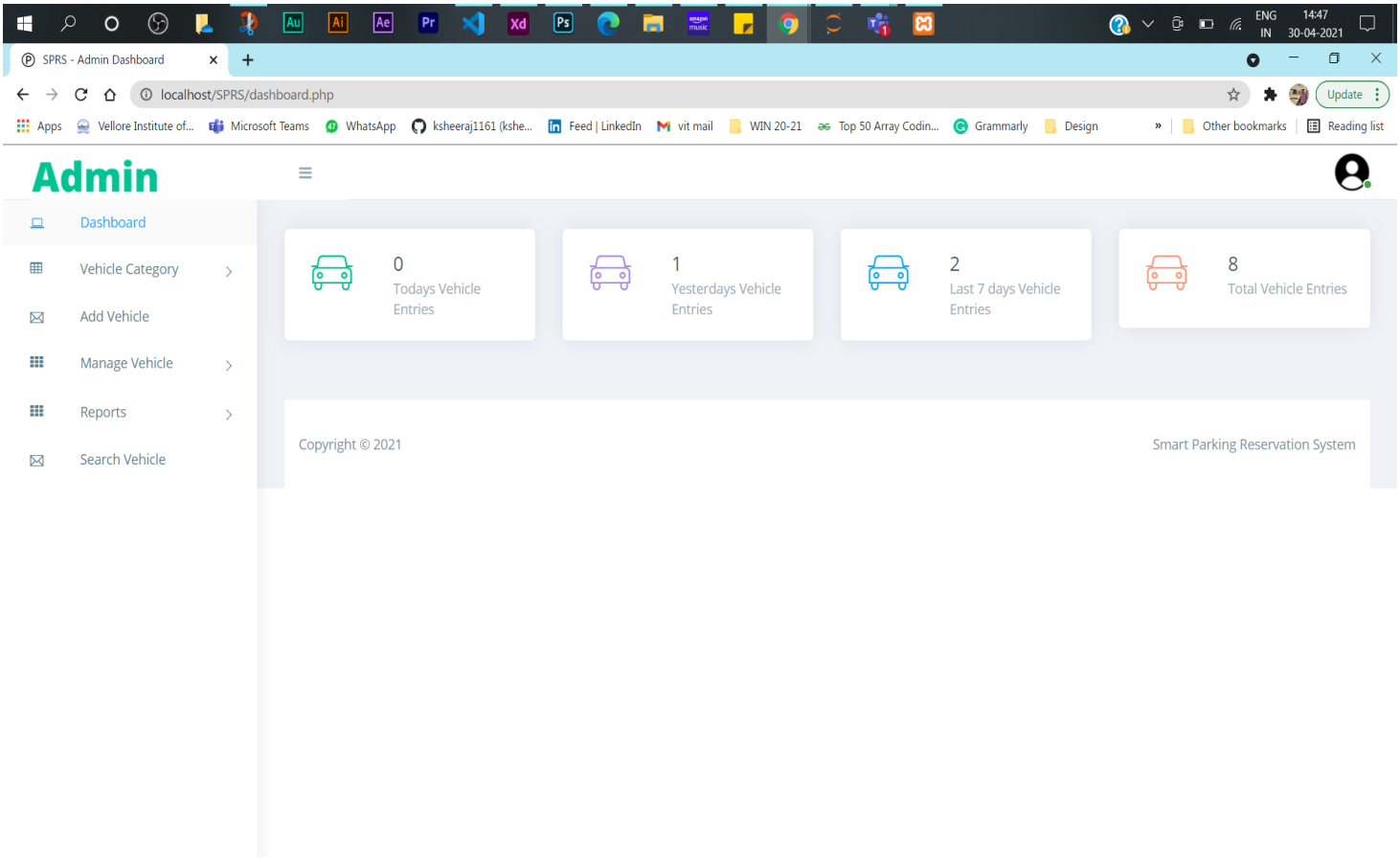


- **Admin Login :**

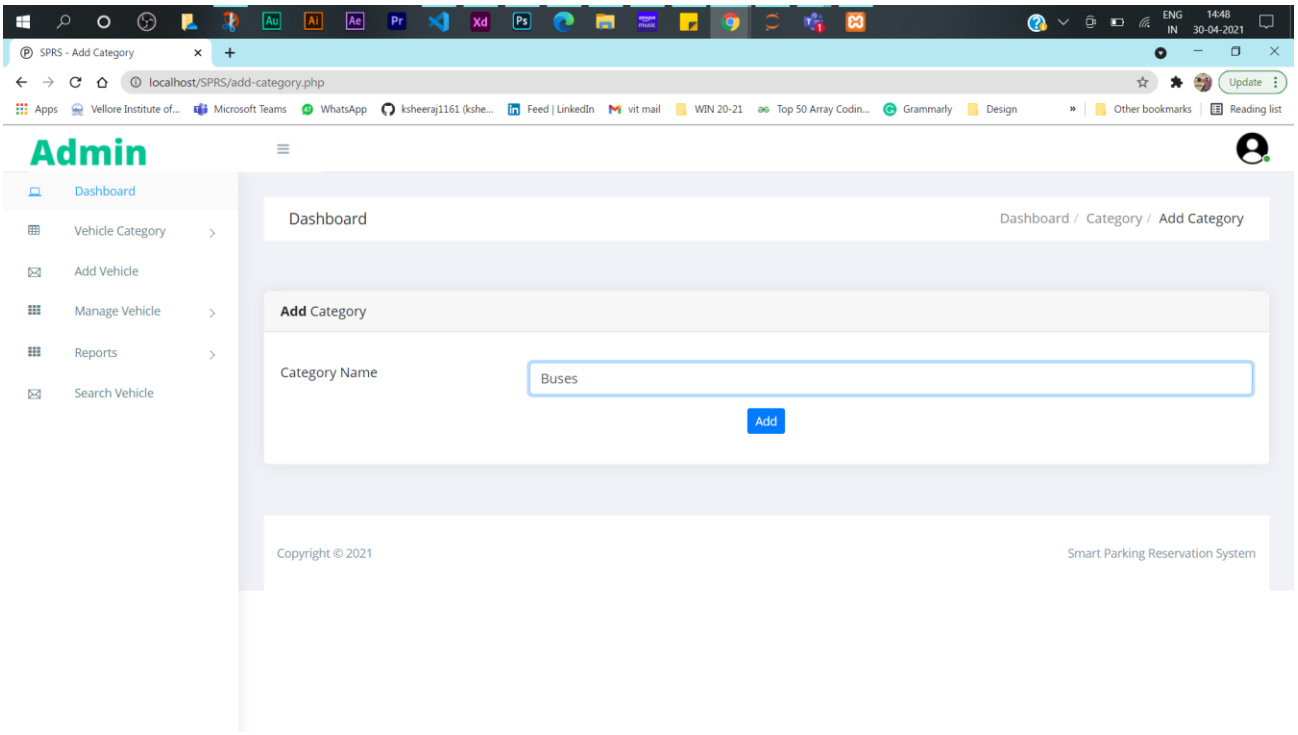


- **Dashboard**

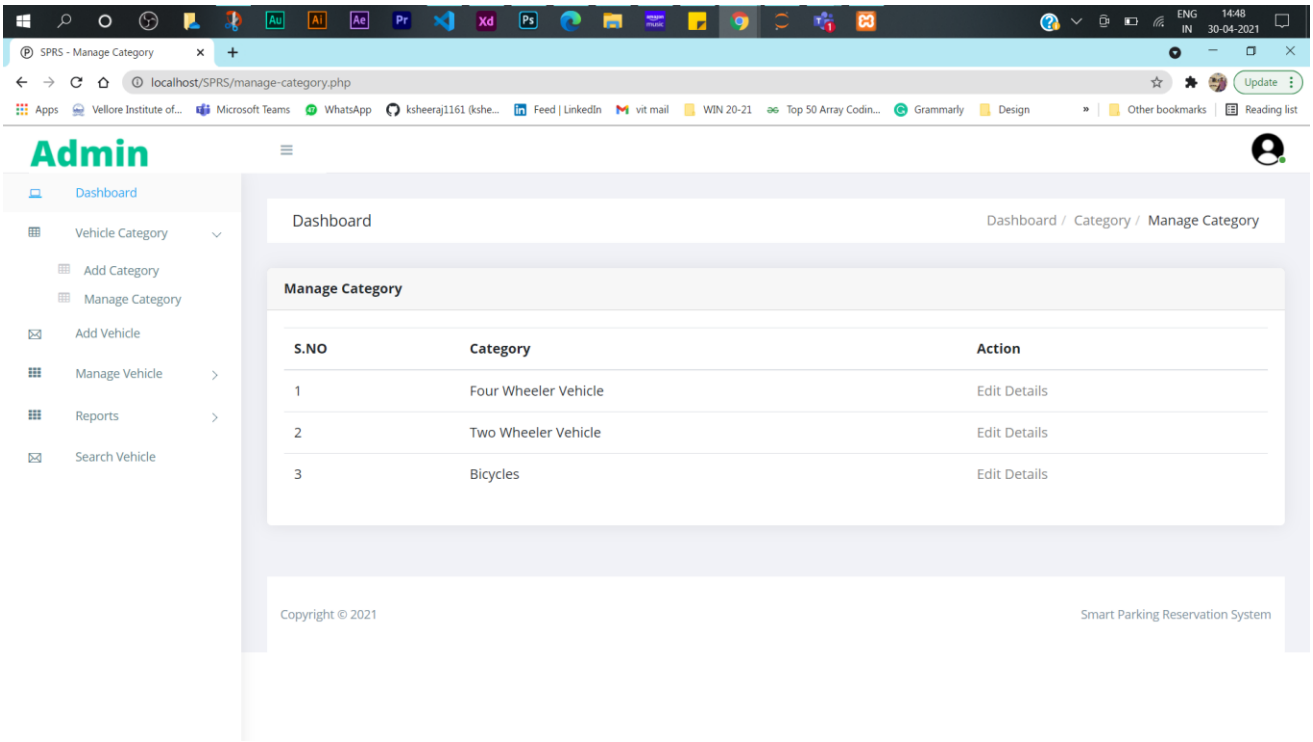

SPRS - Admin Dashboard
localhost/SPRS/dashboard.php
Admin
0
Todays Vehicle Entries
1
Yesterdays Vehicle Entries
2
Last 7 days Vehicle Entries
8
Total Vehicle Entries
Copyright © 2021
Smart Parking Reservation System

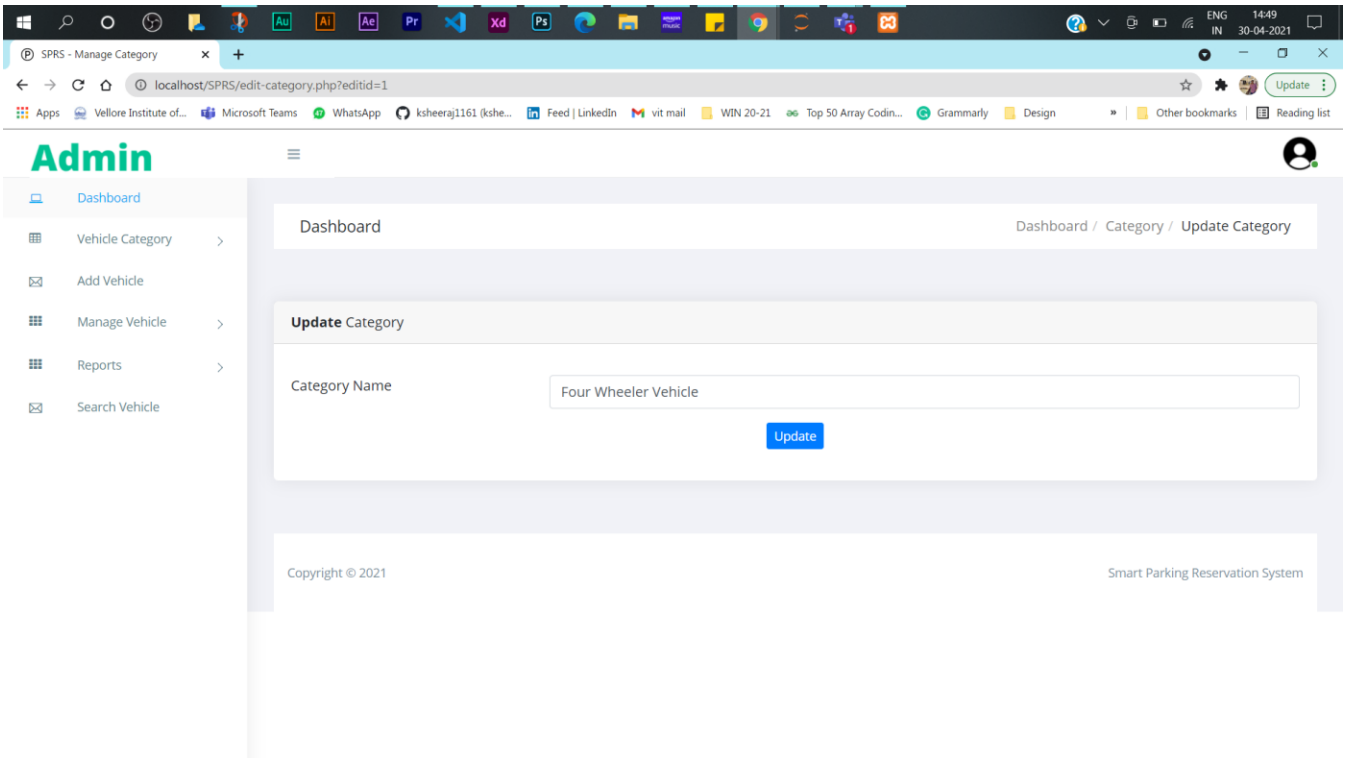

SPRS - Admin Dashboard
localhost/SPRS/dashboard.php
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Search Vehicle
0
Todays Vehicle Entries
1
Yesterdays Vehicle Entries
2
Last 7 days Vehicle Entries
8
Total Vehicle Entries
Copyright © 2021
Smart Parking Reservation System

- **Add Vehicle Category :**

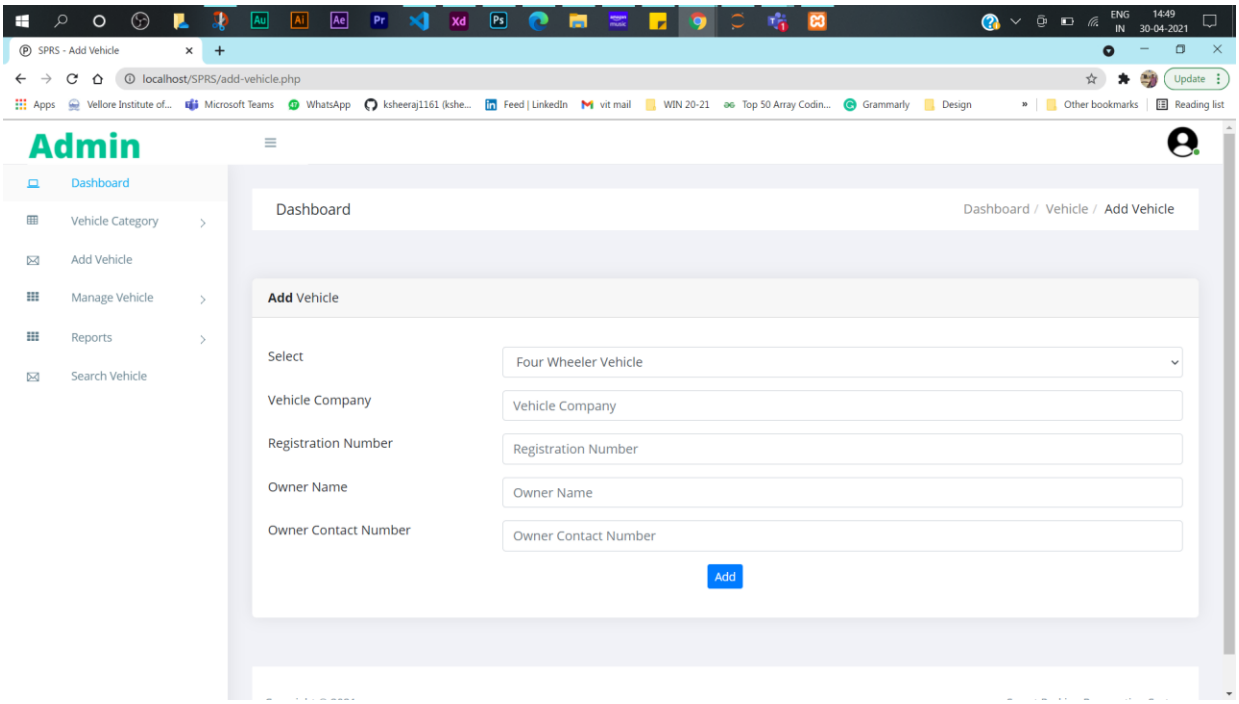


- **Manage Vehicle Category :**

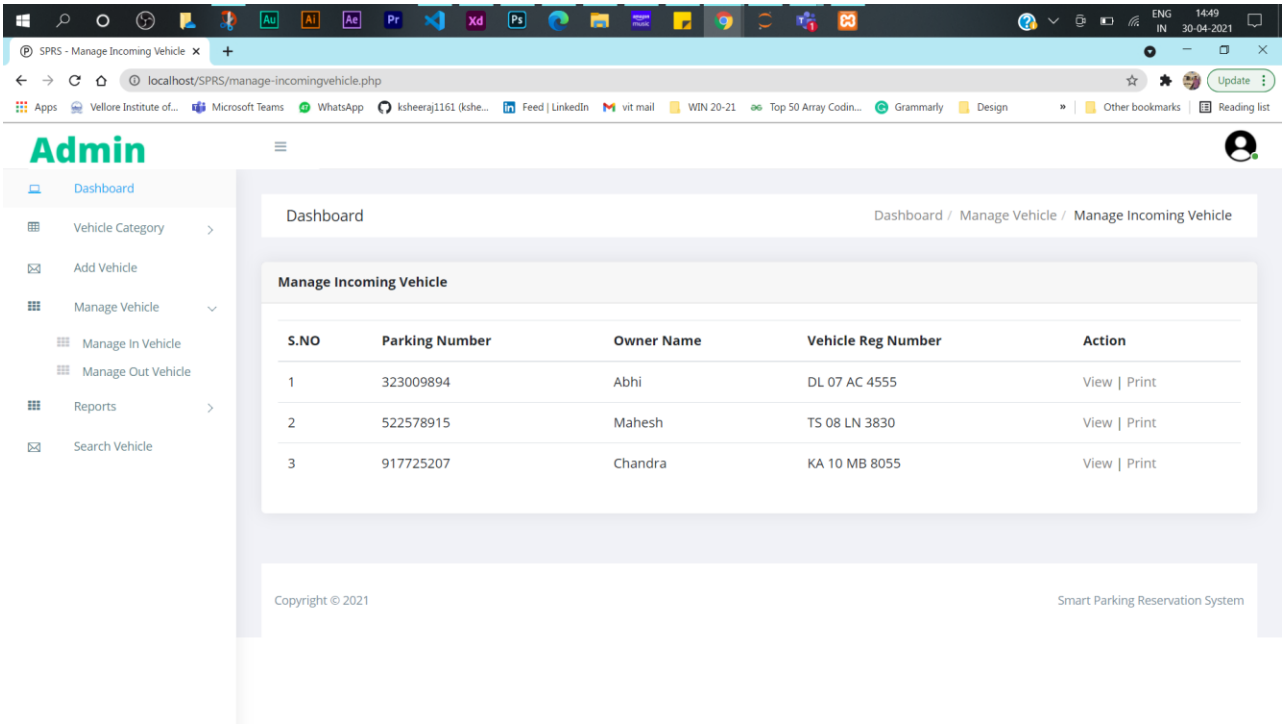



SPRS - Manage Category
localhost/SPRS/edit-category.php?editid=1
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Search Vehicle
Dashboard
Dashboard / Category / Update Category
Update Category
Category Name
Four Wheeler Vehicle
Update
Copyright © 2021
Smart Parking Reservation System

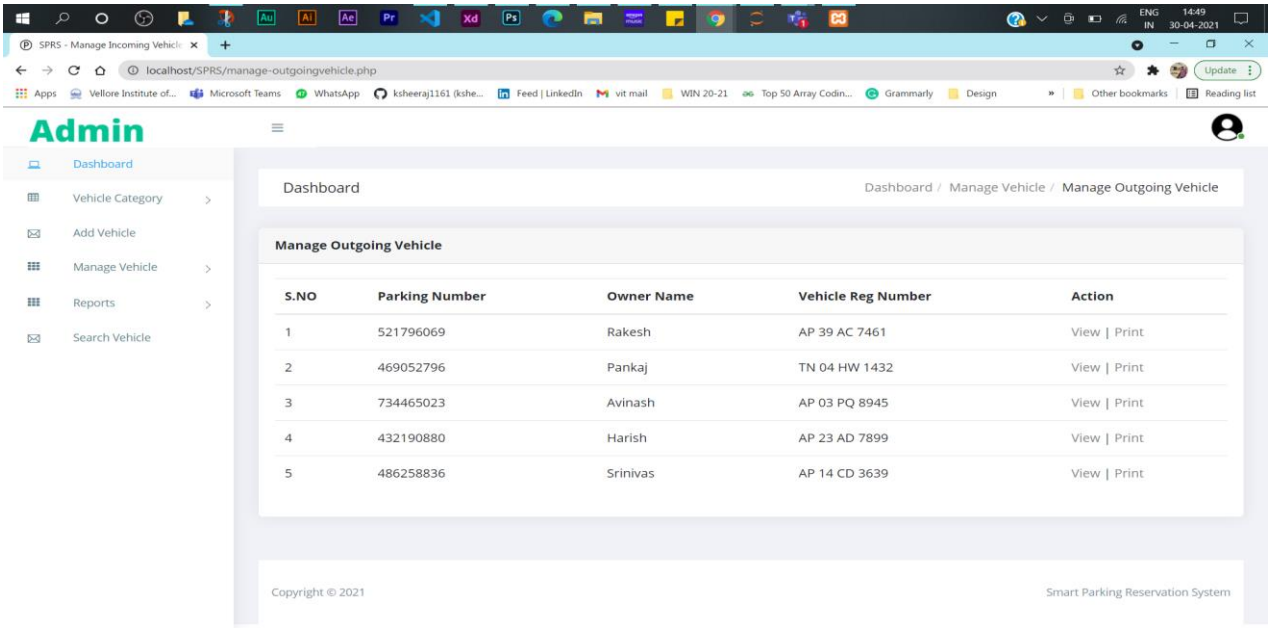
- **Add Vehicle :**

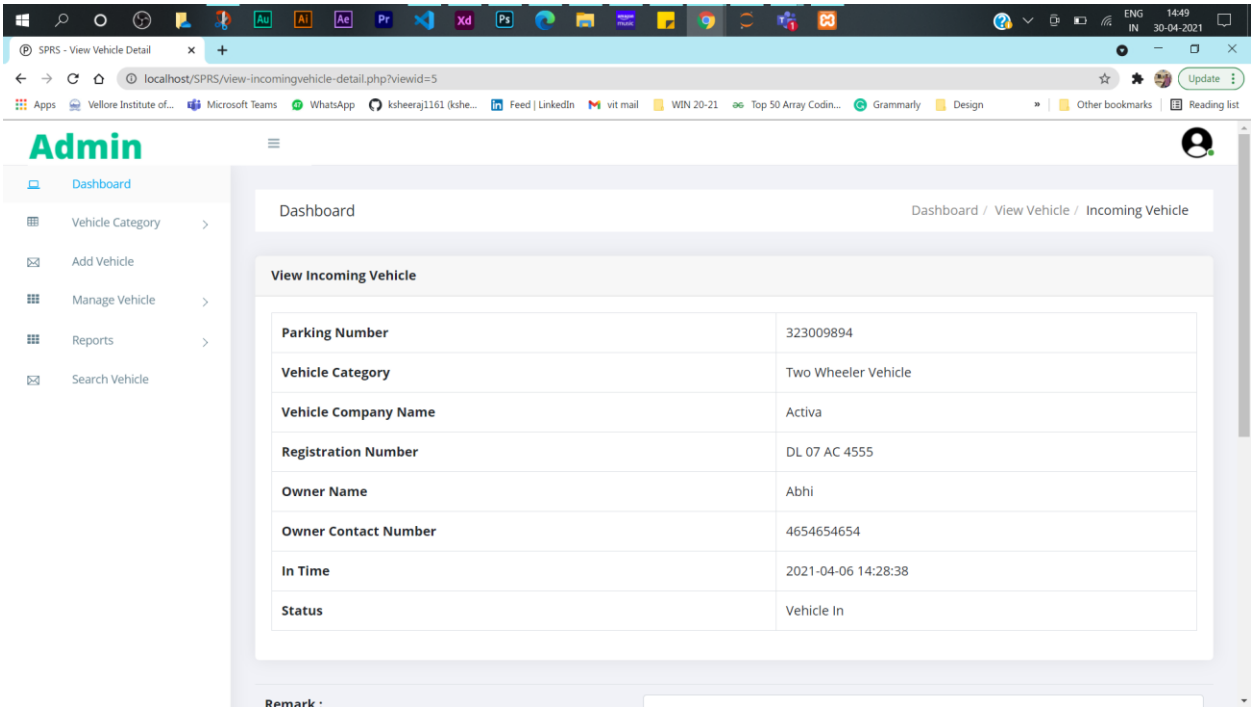

SPRS - Add Vehicle
localhost/SPRS/add-vehicle.php
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Search Vehicle
Dashboard
Dashboard / Vehicle / Add Vehicle
Add Vehicle
Select
Four Wheeler Vehicle
Vehicle Company
Vehicle Company
Registration Number
Registration Number
Owner Name
Owner Name
Owner Contact Number
Owner Contact Number
Add

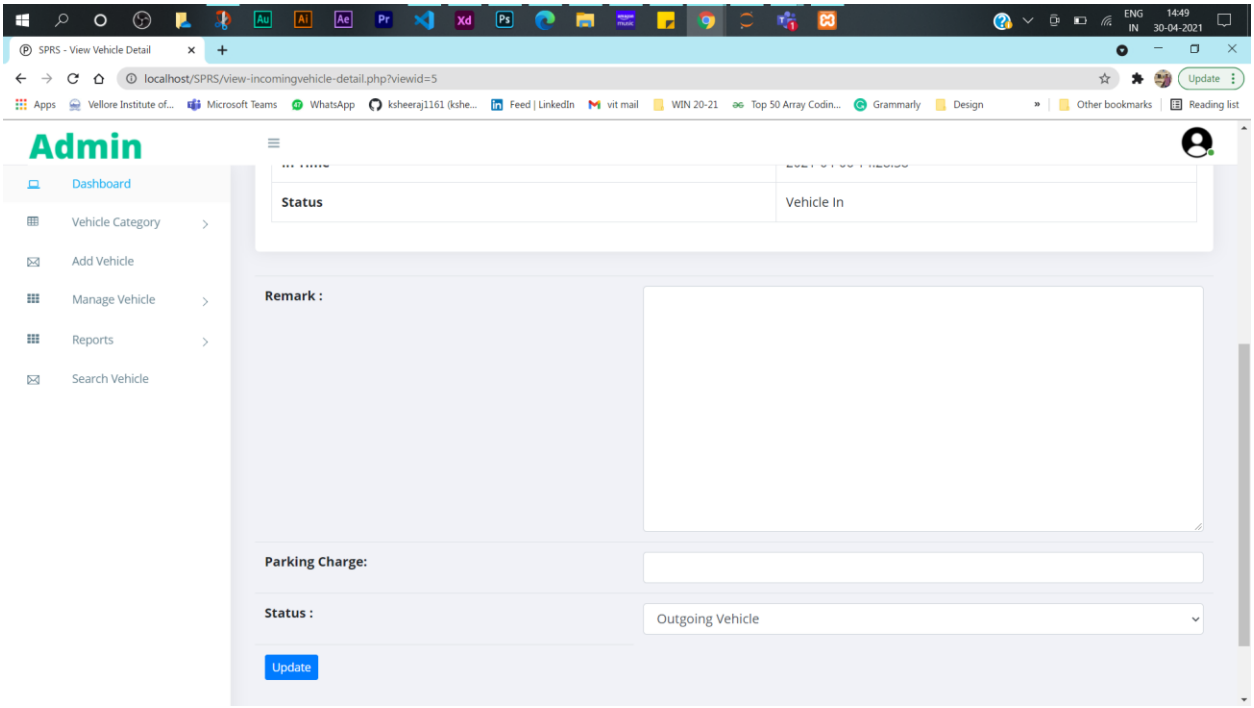
- **Manage Vehicle :**



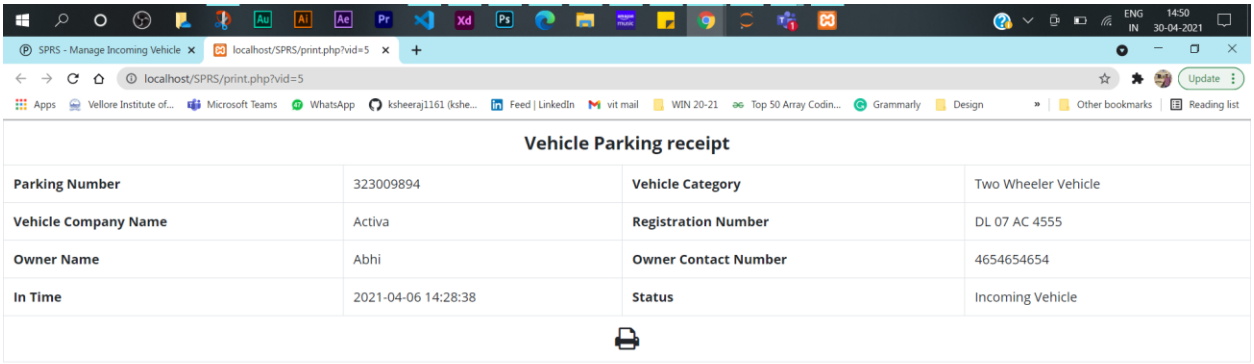
- **Incoming and Outgoing vehicle**




SPRS - View Vehicle Detail
localhost/SPRS/view-incomingvehicle-detail.php?viewid=5
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Search Vehicle
Dashboard
Dashboard / View Vehicle / Incoming Vehicle
View Incoming Vehicle
Parking Number
323009894
Vehicle Category
Two Wheeler Vehicle
Vehicle Company Name
Activa
Registration Number
DL 07 AC 4555
Owner Name
Abhi
Owner Contact Number
4654654654
In Time
2021-04-06 14:28:38
Status
Vehicle In
Remark :


SPRS - View Vehicle Detail
localhost/SPRS/view-incomingvehicle-detail.php?viewid=5
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Search Vehicle
Status
Vehicle In
Remark :
Parking Charge:
Status :
Outgoing Vehicle
Update

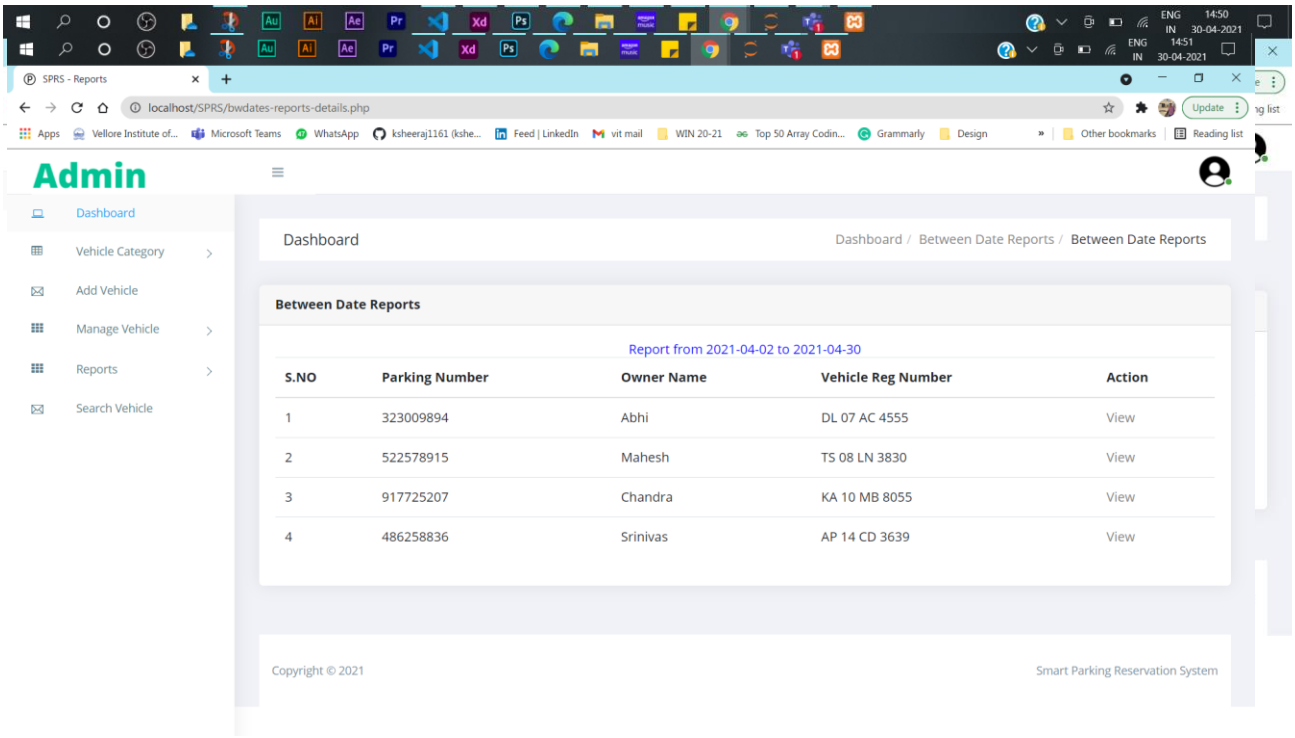
- **Print Receipt**



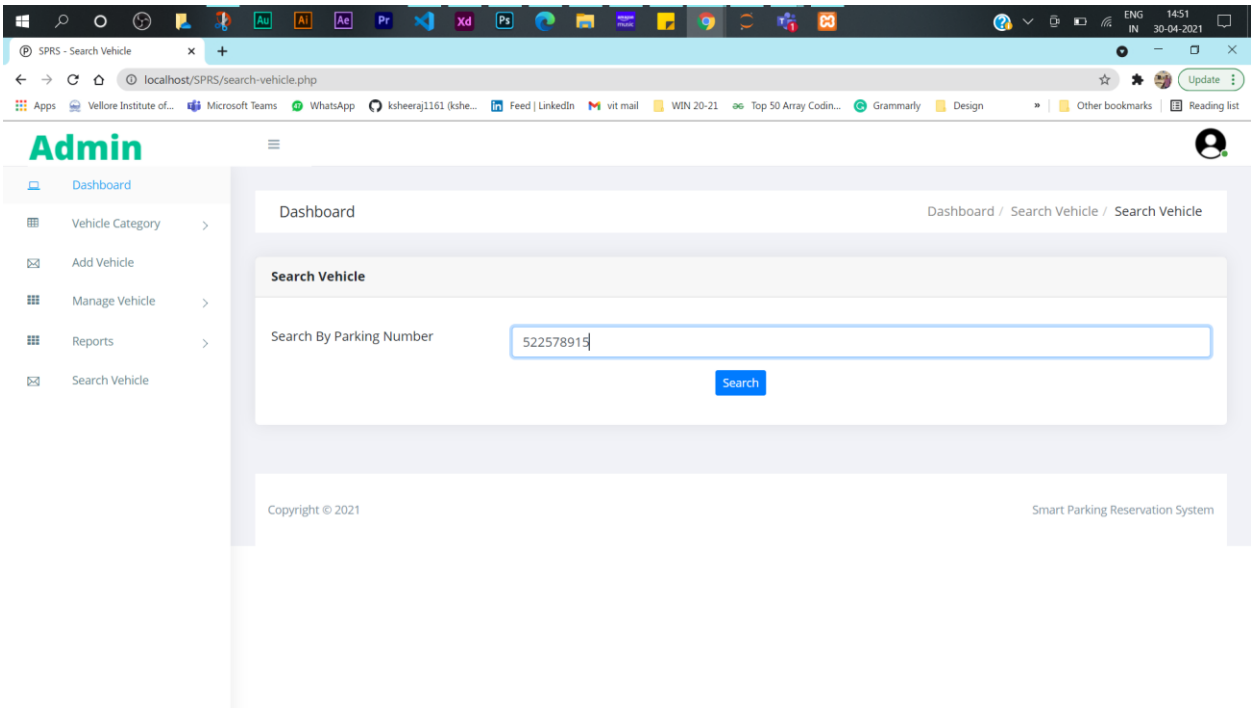
Vehicle Parking receipt

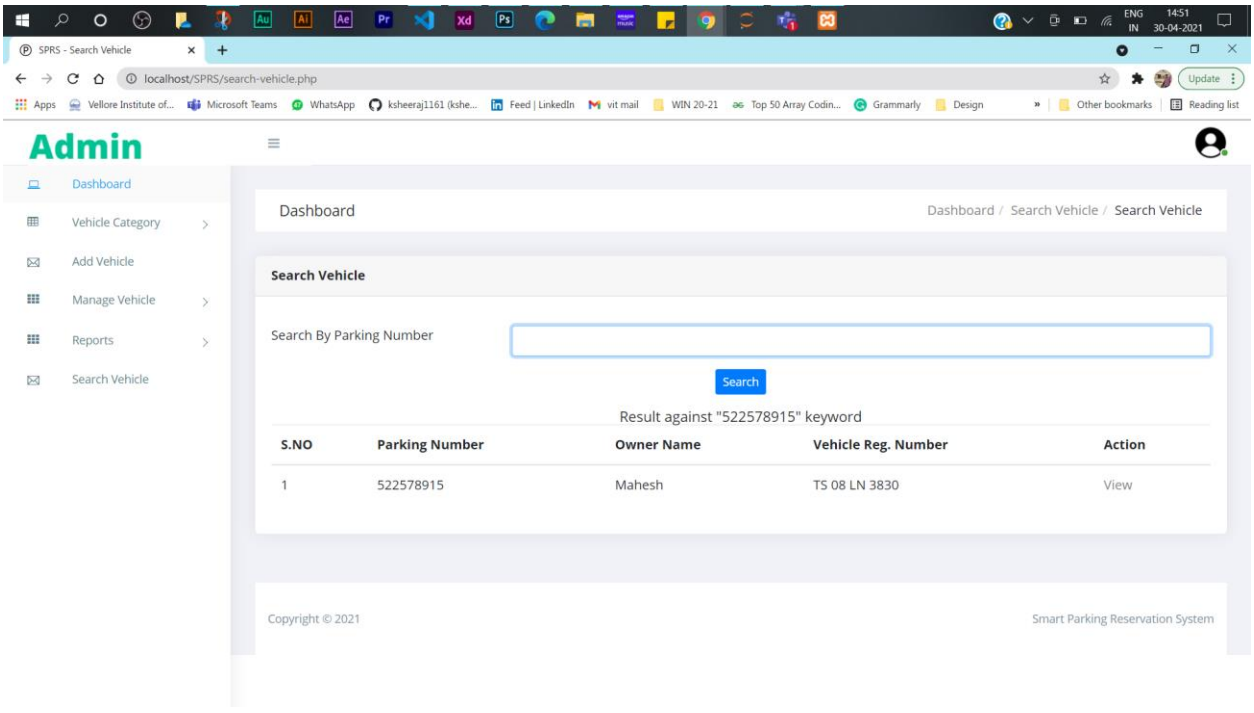
Parking Number	323009894	**Vehicle Category**	Two Wheeler Vehicle
Vehicle Company Name	Activa	**Registration Number**	DL 07 AC 4555
Owner Name	Abhi	**Owner Contact Number**	4654654654
In Time	2021-04-06 14:28:38	**Status**	Incoming Vehicle

- **Reports**

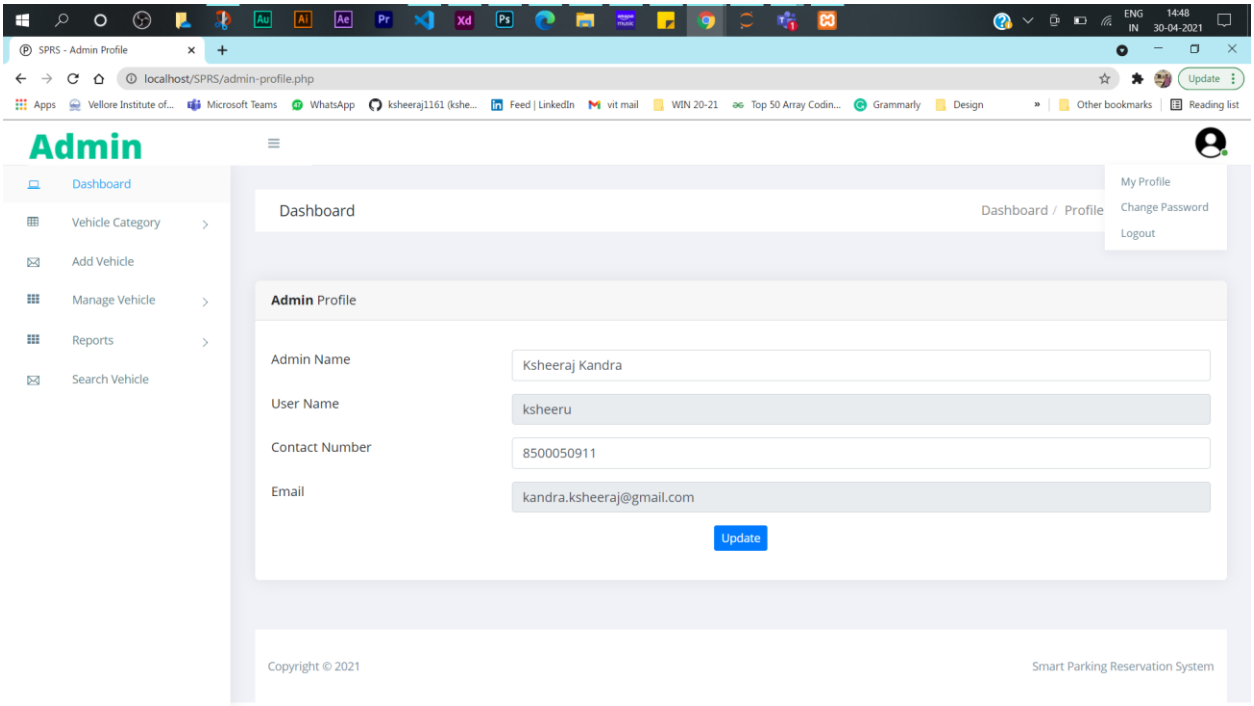


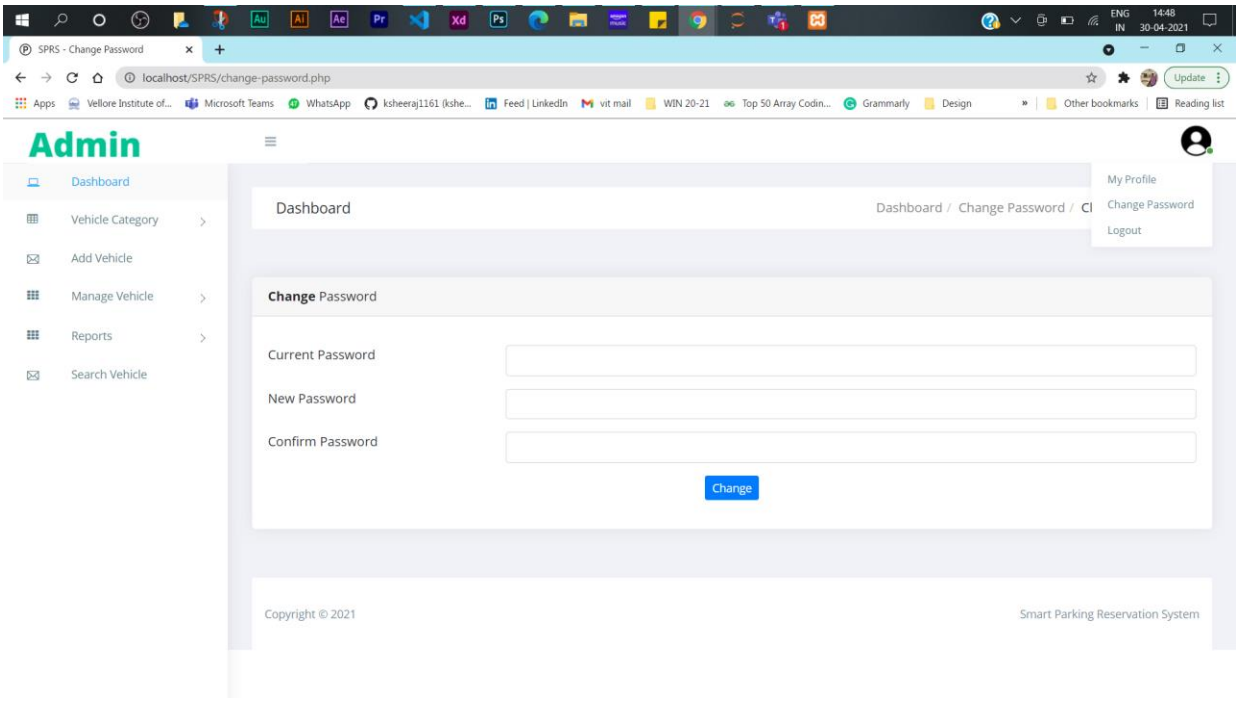
- **Search Vehicle :**





- **Profile Update**





iii) Tool / platform used

- VS Code - *code editor*
- Xampp - *the web server provider*
- Mysql - *database*
- Adobe XD - *user interface with prototyping*
- Thenounproject - *vector icons*
- Flaticon - *vector icons*
- Colours.neilorangepeel -*css codes*

VIII. Challenges overcome during the development of the project

1) **Gathering all the initial client(user) requirements.**
   Gathering all the module information first as we started implementing the project using the Waterfall model.
   Each module expected functionalities, assessing them is the first challenging problem that was to be by us.
   We have overcome it by understanding all requirements, workflow of clients and found ways in optimizing workflows.

2) **Time constraint(Limited time assigned for completion of project)**
   Due to lack of sufficient time, we could not implement all the ideas (like image processing of car number plates, location tracking).
   We could not validate the accuracy of login details of customers/users (by verification of aadhar number, mobile number, personal details).
   We didn't consider all the limitations of this system. But we tried to consider a few of them and we are in the process of rectifying them with new ideas/methods.

3) **Site usability(Failure of user interaction)**
   While creating the website we have failed to predict the user interaction with the site. This resulted in failure of many tasks and their functionalities.
   This was solved by visualizing every step of process layout by carefully creating and understanding UML diagrams and UI design.

4) **Implementation and completion of payment module.**
   Payment module is the most complex module. Implementation of this module took us a lot of time to discuss the procedure and types of payment methods.
   We may not implement all the functionalities as expected but we are working hard to emulate a perfect payment portal.We are working hard to implement the perfect working of this module for secure transactions of the users.

5) Navigation

Navigation is the most important aspect of website design. The navigational structure of the website is also a big challenge in the process of website creation. Websites should be easily navigable for the users.
Effective navigational structure enhances the usability of our website.

6) XAMPP service provider usage

XAMPP(cross-platform, Apache, MySQL, PHP and Perl) is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver.
This provides an interface between our backend (sql and php) database and front end of our website.This is hard to use and it took us time to understand the usage of XAMPP.This ensures the perfect working of our website.