A

PROJECT REPORT ON

**Online Parking Management System**

**By**

**KEVAL TALAVIYA (CE-132) (17CEUOS077)**

**SHUBHAM VEKARIYA (CE-144) (17CEUOS111)**

**NEMISH ZALAVADIYA (CE-146) (17CEUOG074)**

**B.Tech CE Semester-VI**

**Subject: System Design Practice**

**Guided by:**

Prof. Pandav K. Patel

Assistant Professor

Dept. of Comp. Engg.



**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**



**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**

**CERTIFICATE**

This is to certify that the practical / term work carried out in the subject of

**System Design Practice** and recorded in this journal is the

bonafide work of

**KEVAL TALAVIYA (CE-132) (17CEUOS077)**

**SHUBHAM VEKARIYA (CE-144) (17CEUOS111)**

**NEMISH ZALAVADIYA (CE-146) (17CEUOG074)**

of B.Tech semester **VI** in the branch of **Computer Engineering**

during the academic year **2019-2020**.

|  |  |
| --- | --- |
| Prof. Pandav k. Patel  Assistant Professor,  Dept. of Computer Engg.,  Faculty of Technology  Dharmsinh Desai University, Nadiad | Dr. C. K. Bhensdadia  Head,  Dept. of Computer Engg.,  Faculty of Technology  Dharmsinh Desai University, Nadiad |

**Table of Contents**

1. Abstract………………………………………………………………….…….4
2. Introduction…………………………………………………………….……..5
   1. 2.1 Project Details: Brief Introduction…………………………...…...5
   2. 2.2 Technology and Tools Used……………………………….……...5
3. Software Requirement Specifications……………………………….…..6

3.1System Functional Requirements ………………………………….6

1. Design …………………………………………………………………..……..10
2. Implementation Details……………………………………………………..23

5.1 implementation Details…………………………………………..….23

5.2 Function Prototype………………………………………………..…24

1. Testing ……………………………………………………………………..….25

6.1 Testing Method…………………………………………………..…..25

6.2 Test Cases…………………………………………………………...25

1. Screen‐shots of the System ………………………………………….……27
2. Conclusion …………………………………………………………….……...34
3. Limitations of System…………………………………………………....…35
4. Future Extensions to the System …………………………………….…35

11 Bibliography………………………………………………………….. …....35

**Abstract**

In this 21st century, population growth and urbanization increased so it is a big problem to find place for parking in city area.Therefore, our web application will be useful to the people who have been face a daily problem in the park of their vehicles And if the user is at an unknown place, then the user will not know the parking place near him, then this application will also be useful to the user and user can park their vehicle.we provide this facility to user so user have not problem for parking vehicle.

# Brief Introduction

The "**Online Parking Management System**" project has been designed keep in mind Landlord, Administrator and end user who wants to take land on rent for parking purpose. Landlord can add their land with location and land details And Landlord can edit his land in future. Landlord can also delete the land which he has added and can also see all the land which he has added and also view the history of all lands and also get payout of land. The administrator can verify the land added by the landlord by checking all the details. If the land added by Landlord is not verified then the land user cannot see that land. The end user can view any nearby parking space and also reserved that place. End user can reserve parking space from any place but for that, he has to added that place location once. The end user can also view the history of all the parking locations he or she had previously used. End user can give feedback of any land and When a user opens nearby parking place page, they can see this feedback.

* 1. **Tools/Technologies**
* **Technologies**: python, html5, Bootstrap4, jQuery
* **Framework**: Django
* **Tool**: Visual Studio Code, PyCharm, GitHub Desktop, Selenium Testing Tool
* **Platform**: Window, Linux

**Software Requirement Specification**

R1. **Authentication**

R1.1 Login

Input: Email address and password

Output: Authentication successful

Description: If credentials are succeeded then user related pages display else error page display on failure.

R1.2 Register

Input: New user credentials

Output: Registration successful

Description: User can register themselves and get successful message.

R1.3 Forgot Password

Input: Email Address

Output: Reset password link mail to user

Description: User reset password by link.

R1.4 Edit profile information

Input: User modify credentials

Output: Edit successful

Description: User can modify their details and get successful message.

R2. **Admin**

R2.1 Verify Land Details

Input: Select land

Output: Send successful mail to landlord

Description: Admin can verify land details by checking all details correctly.

R3. **User**

R3.1 Reserve parking spot based on current location

R3.1.1 Get nearby parking spots

Input: Date and enable location

Output: Show all nearby parking spots

Description: User can get list of all parking spots which can nearby user current location and list sorted by distance.

R3.1.2 Reserve parking spot

Input: Select parking spot

Output: Reserve parking spot successful

Description: User reserve a nearby parking spot for particular day.

R3.2 Reserve parking spot based on other location

R3.1.1 Add location

Input: Location name and enable location

Output: Add location successful

Description: User add location so he/she can be reserve parking spot in advance.

R3.1.1 Get parking spots based on added location

Input: Date and select location name

Output: Show all parking spots

Description: User can get list of all parking spots which can nearby user selected location and list sorted by distance.

R3.1.2 Reserve parking spot

Input: Select parking spot

Output: Reserve parking spot successful

Description: User reserve a nearby parking spot for particular day in advance.

R3.3 View usage history

Input: Select history page

Output: Show list of all reservation details

Description: User can view history of reservation details which reserved earlier.

R3.4 Provide feedback

Input: Select particular record and provide feedback

Output: Feedback provided successfully

Description: User can provide feedback to particular parking spot which reserved earlier.

R4. **Landlord**

R4.1 Add land details

Input: New land credential and enable location

Output: Get successful message

Description: Landlord can add land details successfully.

R4.2 Edit land details

Input: Land modify credential

Output: Get successful message

Description: Landlord can edit land details successfully.

R4.3 Remove land

Input: Select particular land

Output: Get successful message

Description: Landlord can delete land if and only if land should not reserve by any user.

R4.4 View all land details

Input: Select land details page

Output: Show list of all land among with their details.

Description: Landlord can view all land among with their details which add earlier.

R4.5 View history of particular land

Input: Select particular land

Output: Show list of all usage history among with their user details.

Description: Landlord can all usage history among with their user details for particular land.

R4.6 Get payout

Input: Select particular land

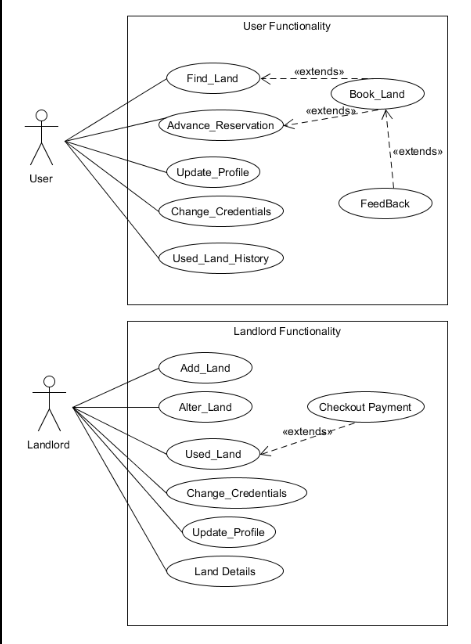
Output: Show the accumulated amount and get the payout successfully.

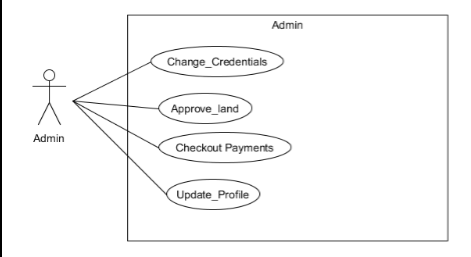
Description: Landlord view the accumulated amount and get the payout

for particular land.

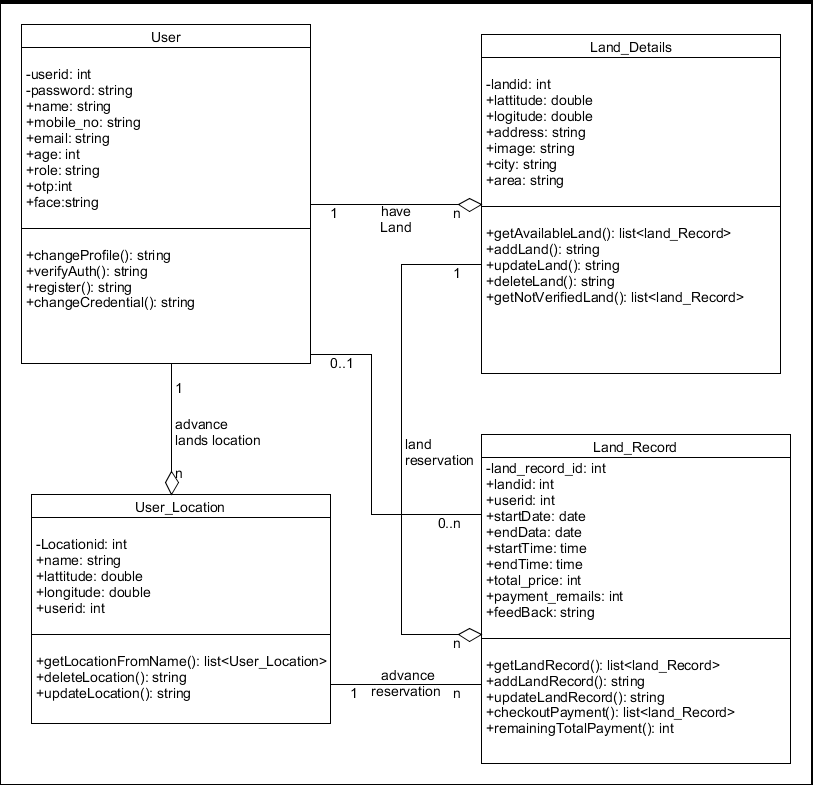
# Design

**Use Case Diagram**

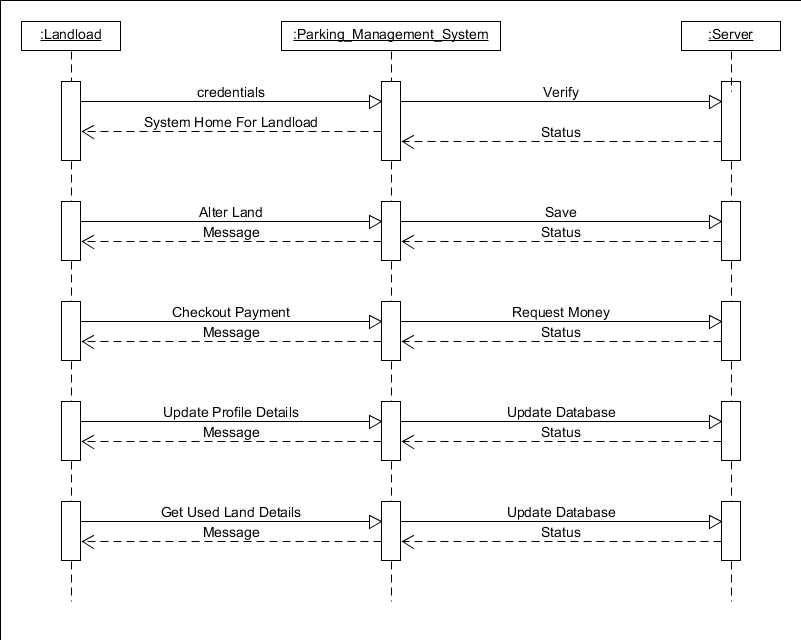
****

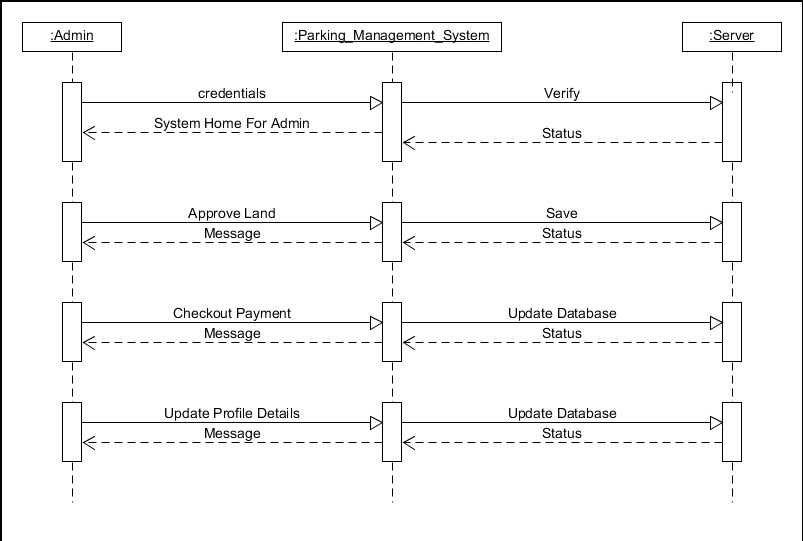
****

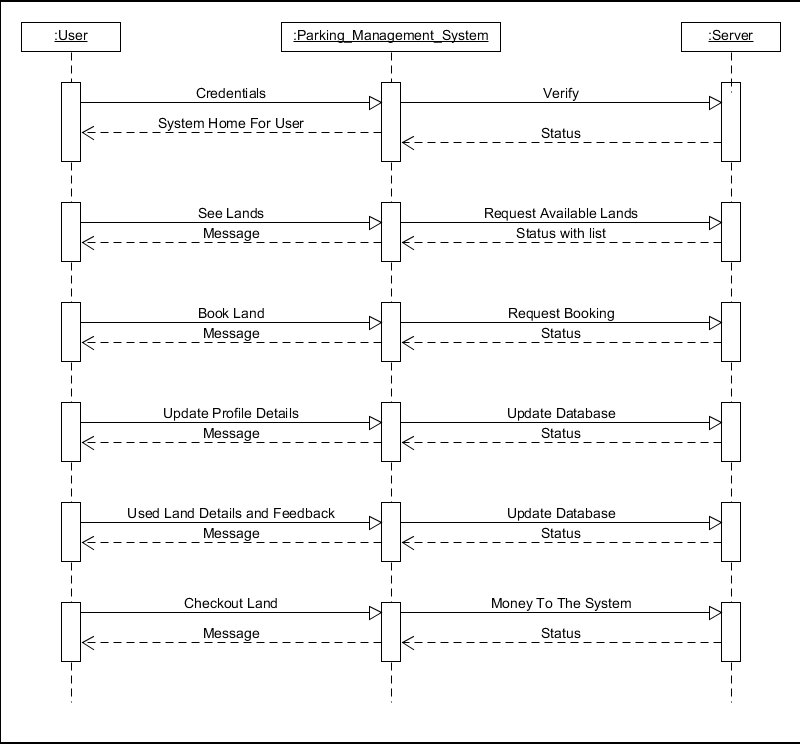
**Class Diagram**

****

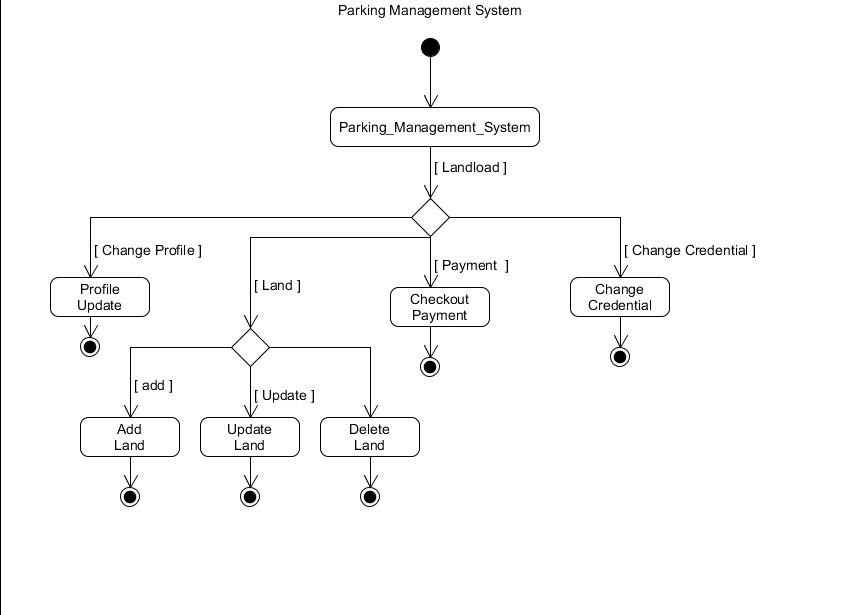
**Sequence Diagram**

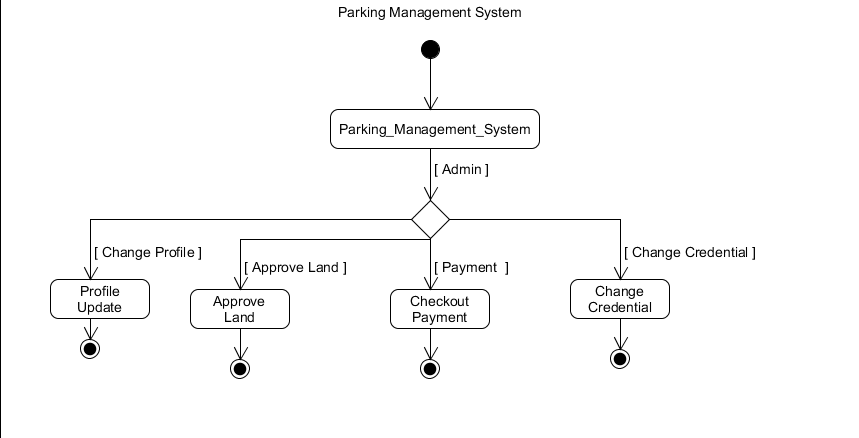
****

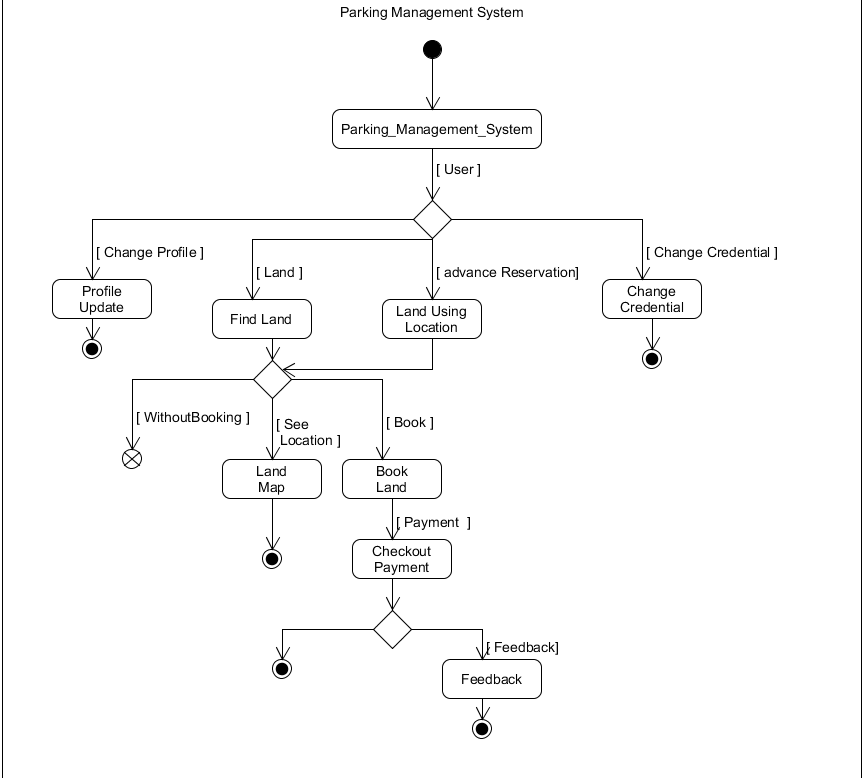
****

****

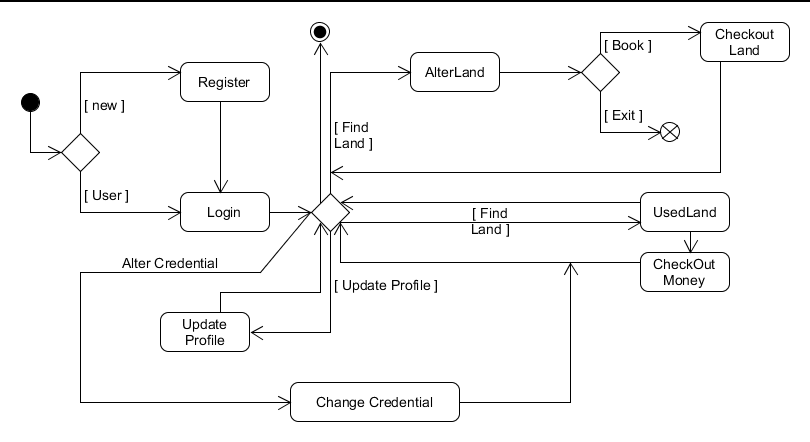
**Activity Diagram**

****

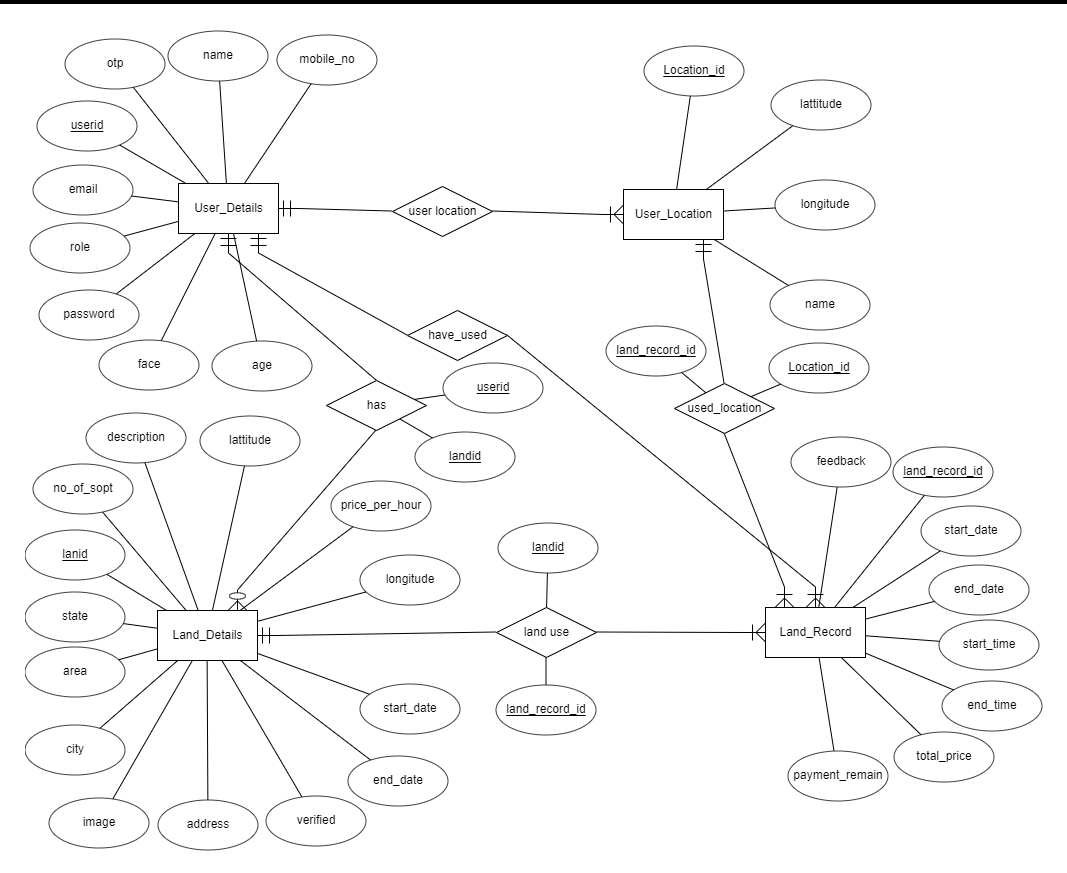
****

****

**State Diagram**

****

1. **R Diagram**

****

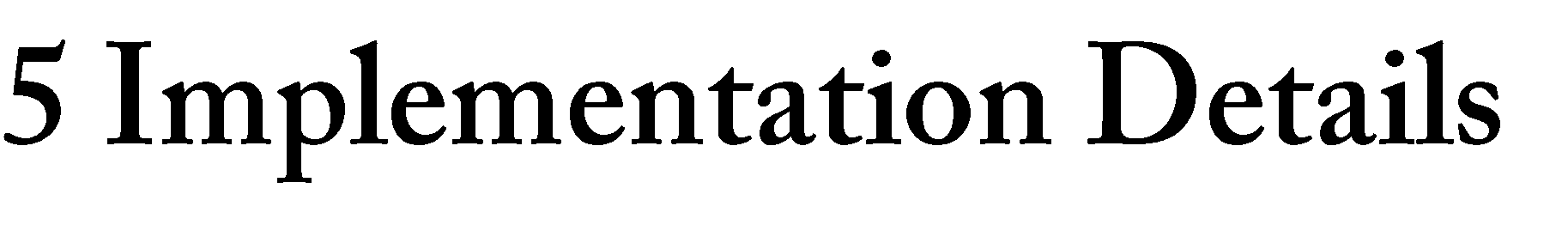
**Data Dictionary**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User\_detail** | | | | | | | | |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | userid | Number | 11 | Yes | Yes | PK |  |  |
| 2. | name | Varchar | 100 | Yes | No |  |  |  |
| 3. | password | Varchar | 14 | Yes | No |  |  |  |
| 4. | mobile\_no | Number | 10 | Yes | No |  |  |  |
| 5. | email | Varchar | 100 | Yes | Yes |  |  |  |
| 6. | age | Number | 11 | Yes | No |  |  |  |
| 7. | role | Varchar | 10 | Yes | No |  |  |  |
| 8. | otp | Number | 11 | No | No |  |  |  |
| 9. | face | Longtext |  | No | No |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User\_Location** | | | | | | | | |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | Locationid | Number | 10 | Yes | Yes | PK |  |  |
| 2. | name | Varchar | 10 | Yes | No |  |  |  |
| 3. | lattitude | Double | 30 | Yes | No |  |  |  |
| 4. | langitude | Double | 30 | Yes | No |  |  |  |
| 5. | userid | Number | 10 | Yes | No | FK | User\_detail |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Land\_detail** | | | | | | | | |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | landid | Number | 11 | Yes | Yes | PK |  |  |
| 2. | lattitude | Double | 30 | Yes | No |  |  |  |
| 3. | langitude | Double | 30 | Yes | No |  |  |  |
| 4. | address | Varchar | 300 | Yes | No |  |  |  |
| 5. | image | Varchar | 100 | No | No |  |  |  |
| 6. | city | Varchar | 255 | Yes | No |  |  |  |
| 7. | area | Varchar | 255 | Yes | No |  |  |  |
| 8. | state | Varchar | 255 | Yes | No |  |  |  |
| 9. | no\_of\_spot | Number | 11 | Yes | No |  |  |  |
| 10. | description | Longtext |  | No | No |  |  |  |
| 11. | price\_per\_hour | Double | 20 | Yes | No |  |  |  |
| 12. | start\_date | Date |  | Yes | No |  |  |  |
| 13. | end\_date | Date |  | Yes | No |  |  |  |
| 14. | verified | Boolean | 1 | Yes | No |  |  |  |
| 15. | userid | Number | 1 | Yes | No | FK | User\_detail |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Land\_record** | | | | | | | | |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table | Description |
| 1. | land\_record\_id | Number | 11 | Yes | Yes | PK |  |  |
| 2. | landid | Number | 11 | Yes | No | FK | Land\_detail |  |
| 3. | userid | Number | 11 | Yes | No | FK | User\_detail |  |
| 4. | start\_date | Date | 300 | Yes | No |  |  |  |
| 5. | end\_date | Date | 100 | No | No |  |  |  |
| 6. | start\_time | Time | 6 | No | No |  |  |  |
| 7. | end\_time | Time | 6 | No | No |  |  |  |
| 8. | total\_price | Number | 11 | Yes | No |  |  |  |
| 9. | payment\_remaining | Boolean | 1 | Yes | No |  |  |  |
| 10. | feedback | Varchar | 255 | No | No |  |  |  |

**5.1 Module Description**

* **Login Module**

This is first page after home page if user want to perform any operation in our application. According to credentials given by any user they can login into the system.

* **Registration Module**

In our system, any user is registered by giving his details.in this system, there are two type of registration page one for landlord and another one for user. Any users of the system can sign up for landlord and end user. users can sign up for both if desired.

* **Landlord Module**

Landlord can add their land with location and land details And Landlord is also edit its land in the future. Landlord can also delete the land which he has added. He can also see all the land which he has added. He also get money from his land. Landlord also see the land wise history.

* **Administrator module**

The administrator is a person who only has the authority to verify the land that has been uploaded by the landlord and if the administrator does not verify that land, the end user cannot see that land for parking.

* **User module**

The end user can view any nearby parking space and also reserved that place. End user can reserve parking space from any place but for that, he has to added that place location once. The end user can also view the history of all the parking locations which he had previously used. End user can give feedback of any land and When a user opens nearby parking place page, they can see this feedback.

**5.2 Function Prototype**

**Operation** :

* **Create** : For registration, reserve parking spot and add land.
* **Read** : For view land list and usage history.
* **Update** : For modify land details and user profile details.
* **Delete** : For Delete particular land.
* **Search** : For view nearby parking spot details

**Testing**

We have done Black box testing and also some validation testing.

* **CSRF (Cross-Site Request Forgery) Token**

Attacker can send URL like for login, logout, payment etc. and if victim click on that link victim perform the task. It called cross site request forgery. For preventing this attack CSRF token is come into picture.so if victim click on the attacker's link than it gives invalid CSRF token because it is different for every request and not create cookies and also cannot access the account. So, we use CSRF Token.

* **Selenium testing tool**

we use selenium testing tool for test our website because It create robust, browser-based regression automation suites and tests, scale and distribute scripts across many environments. for that we have writing test script in both famous browser Google chrome and Mozilla Firefox and we have also test our website in different platform like Linux and window. In this script, we did different test cases in sign in page and sign up page like, in sign in page we check for blank username or password, incorrect username or incorrect password. And in sign up page we check for any blank data, age below 18, both passwords must same. And we also check test case for add parking land page like location enable or not and required field.

* **Test Cases**

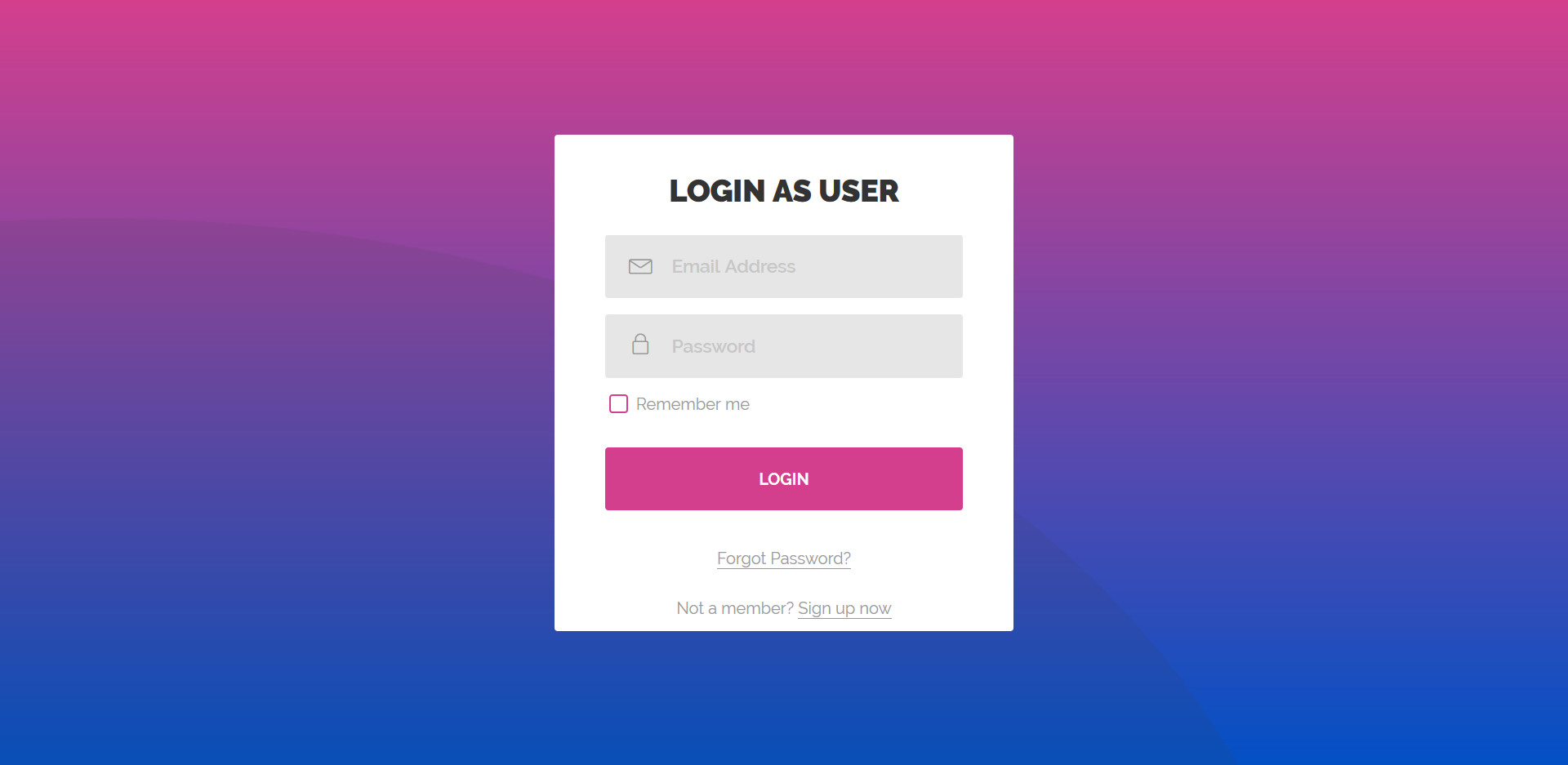
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Scenario** | **Expected Result** | **Actual Result** | **PASS / FAIL** |
| 1. | Password must be same | if both are different then give error otherwise successful. | if both are different then give error otherwise successful. | PASS |
| 2. | Blank field | Give required field error | Give required field error | PASS |
| 3. | Age Restriction | If age less than 18 then give error otherwise successful | f age less than 18 then give error otherwise successful | PASS |
| 4. | Email notification have no significant delay | User is notified in less than 30 seconds after forgot password email entered. | User is notified in less than 20 seconds after forgot password email entered. | PASS |
| 5. | Email address must be valid | It will give error if email address is wrong otherwise successful | It will give error if email address is wrong otherwise successful. | PASS |
| 6. | While Searching  For Lands  If Location is not  Turned on | Please Check This Box If You Want to Proceed | Please Check This Box If You Want to Proceed | PASS |
| 7. | While Adding New Location for Advanced Reservation  Name Is Not Valid. | Please Fill Name  As per Instruction. | Please Fill Name  As per Instruction. | PASS |
| 8. | While Adding New Land Location  Successfully. | Title: Location Done | Title: Location Done | PASS |
| 9. | While Editing Profile Age  If Added age less than 18 | Age must be greater than or equal to 18. | Age must be greater than or equal to 18. | PASS |
| 10. | While Adding Value to the Input  Text boxes. | Show errors according to JavaScript listed | Showing errors according to JavaScript listed | PASS |
| 11. | While Searching for Space showing Land Data | Title: Reserve your favorite space | Title: Reserve your favorite space | PASS |
| 12. | While Searching  For Lands  If Land is not verified by Admin | Land not visible for USER otherwise visible | Land not visible for USER otherwise visible | PASS |
| 13. | While Adding  Land  If Location is not Turned on | Land not visible for USER otherwise visible | Land not visible for USER otherwise visible | PASS |
| 14. | While Deleting  Land  If Land is not reserved by user | Land details delete successful | Land details delete successful | PASS |

**Screen Shot**

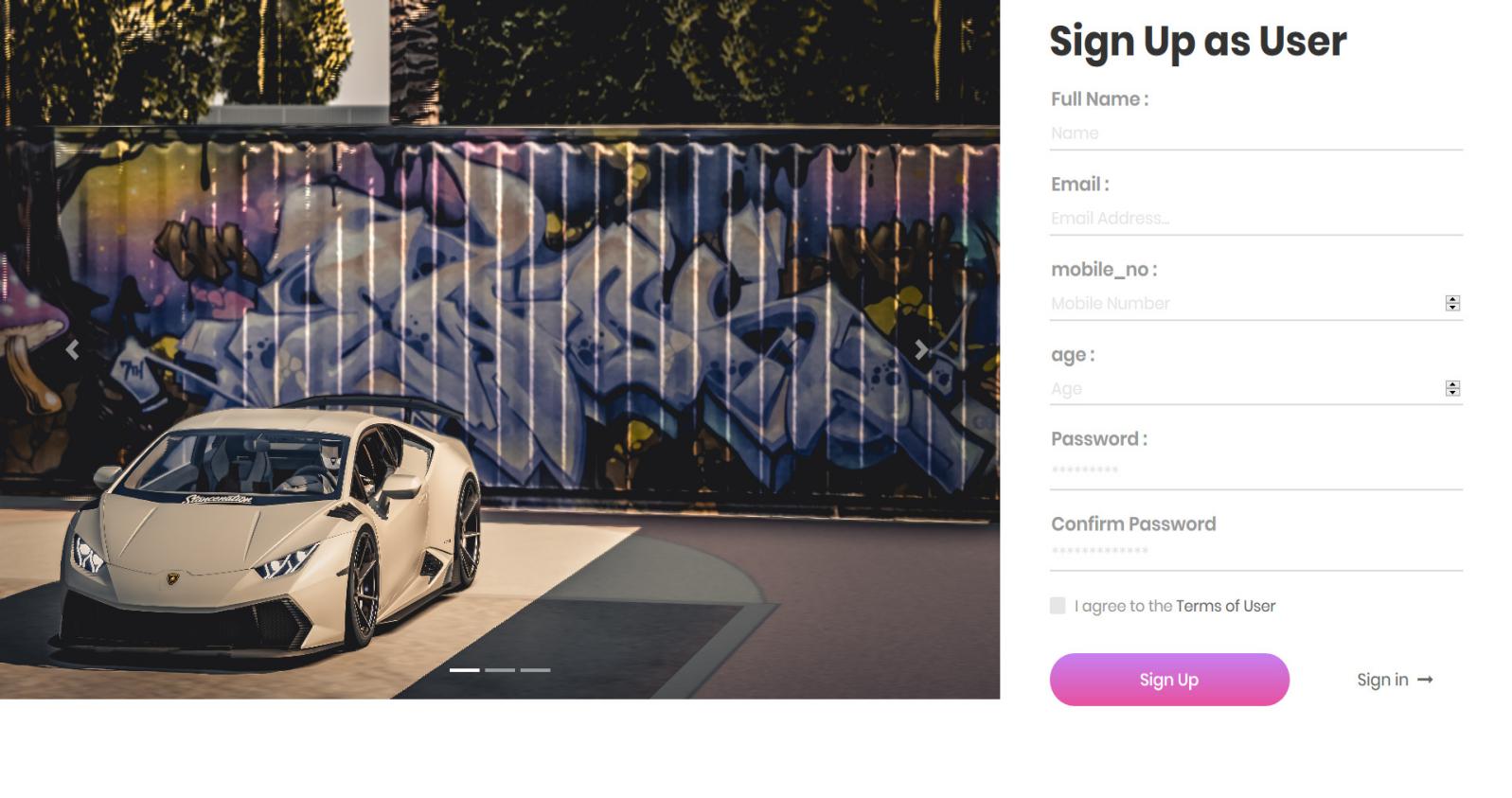
* Home Page

****

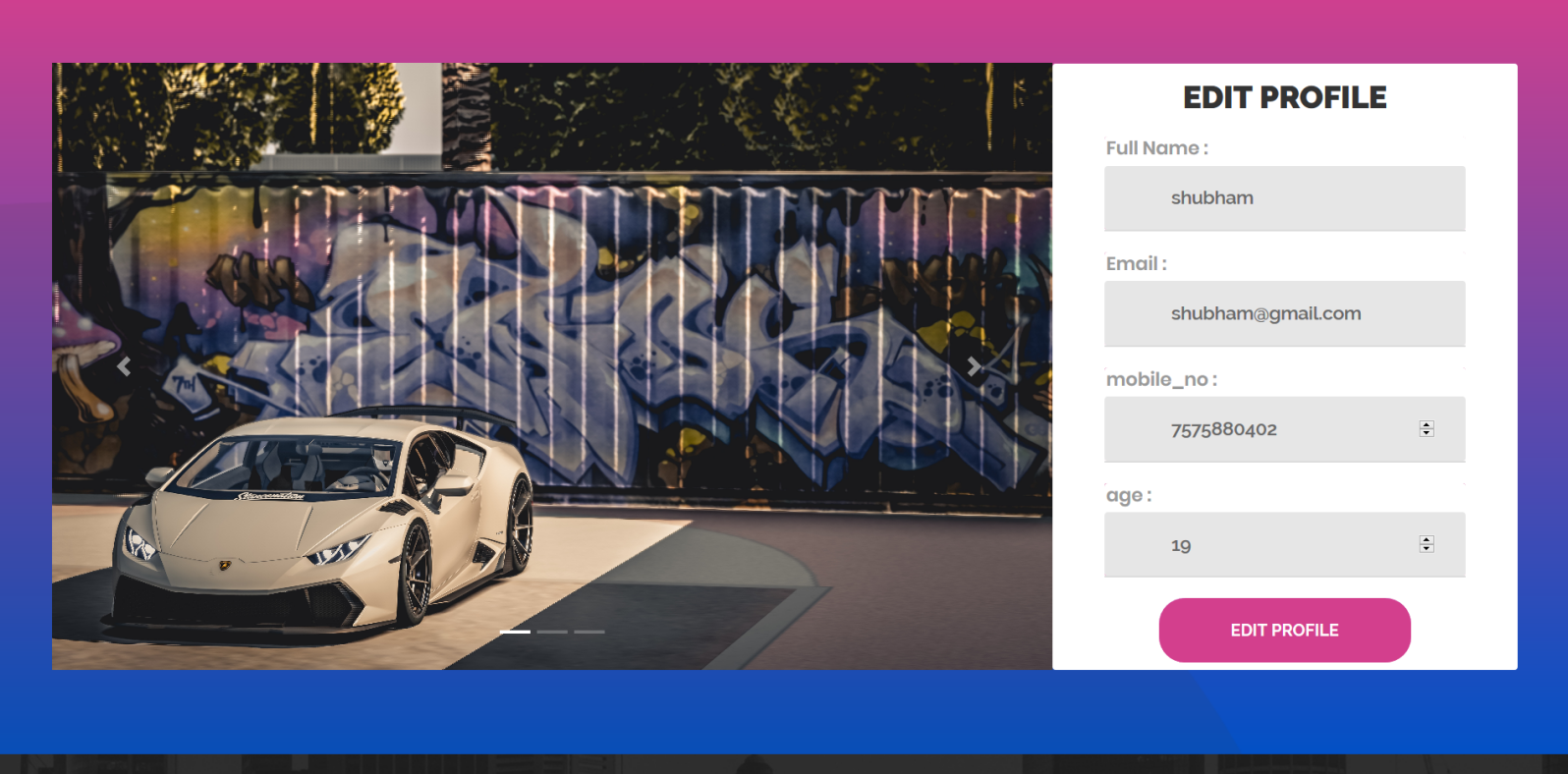
* Login Page



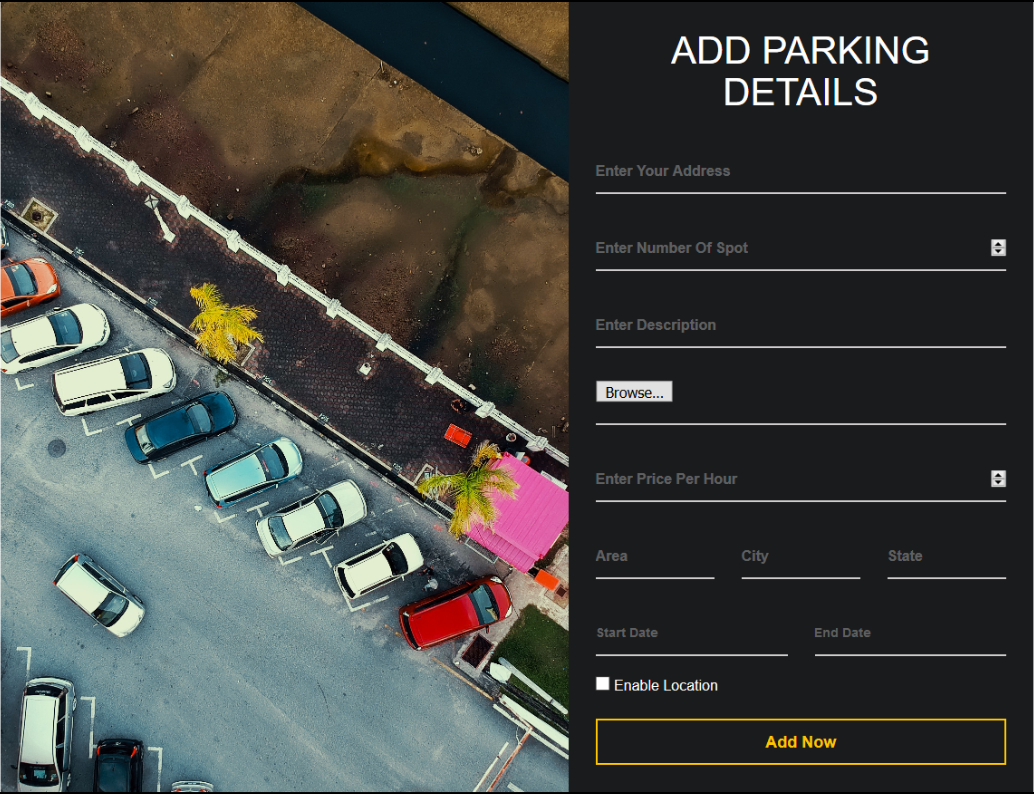
* Sign Up Page



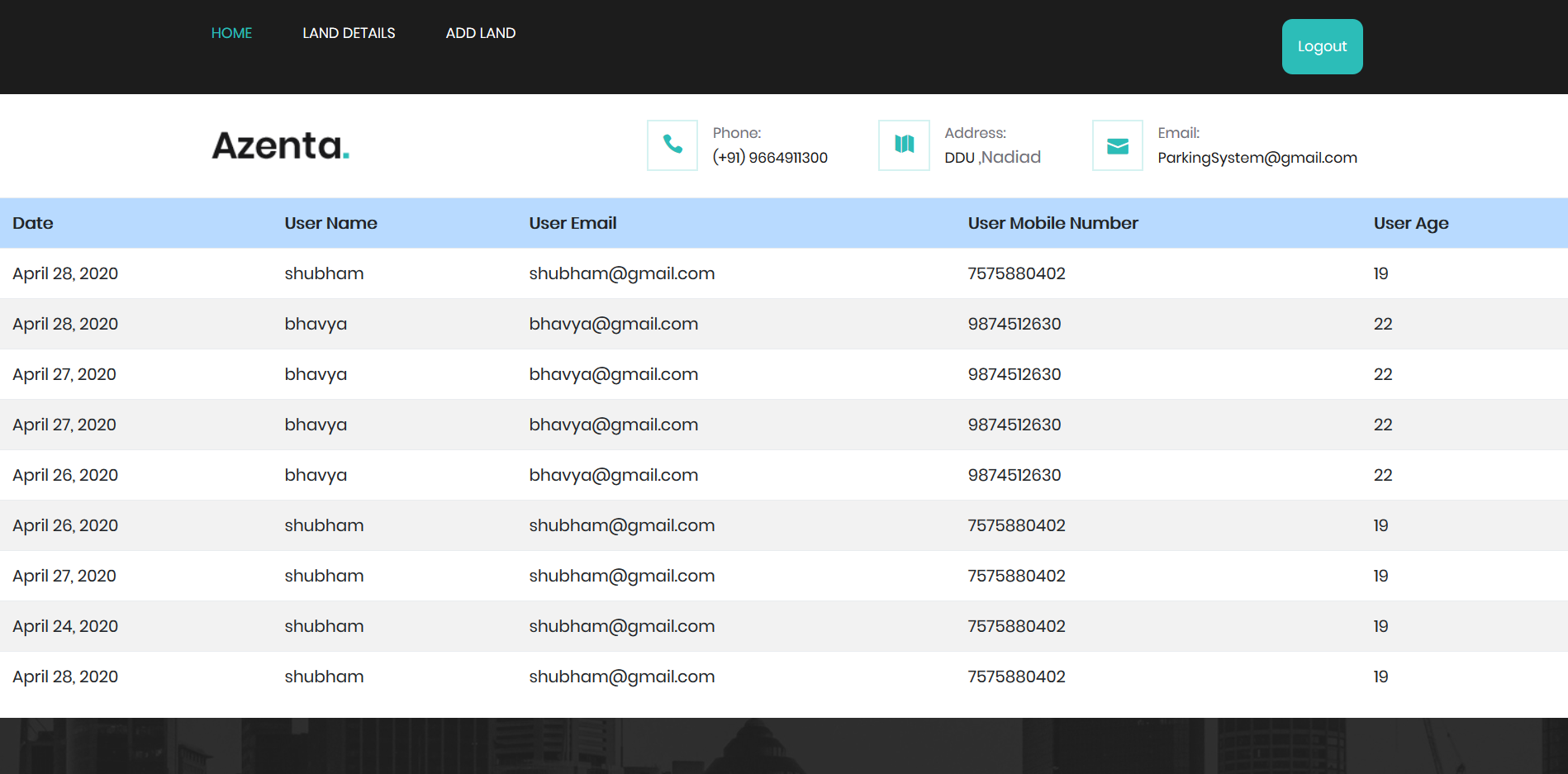
* Edit Profile



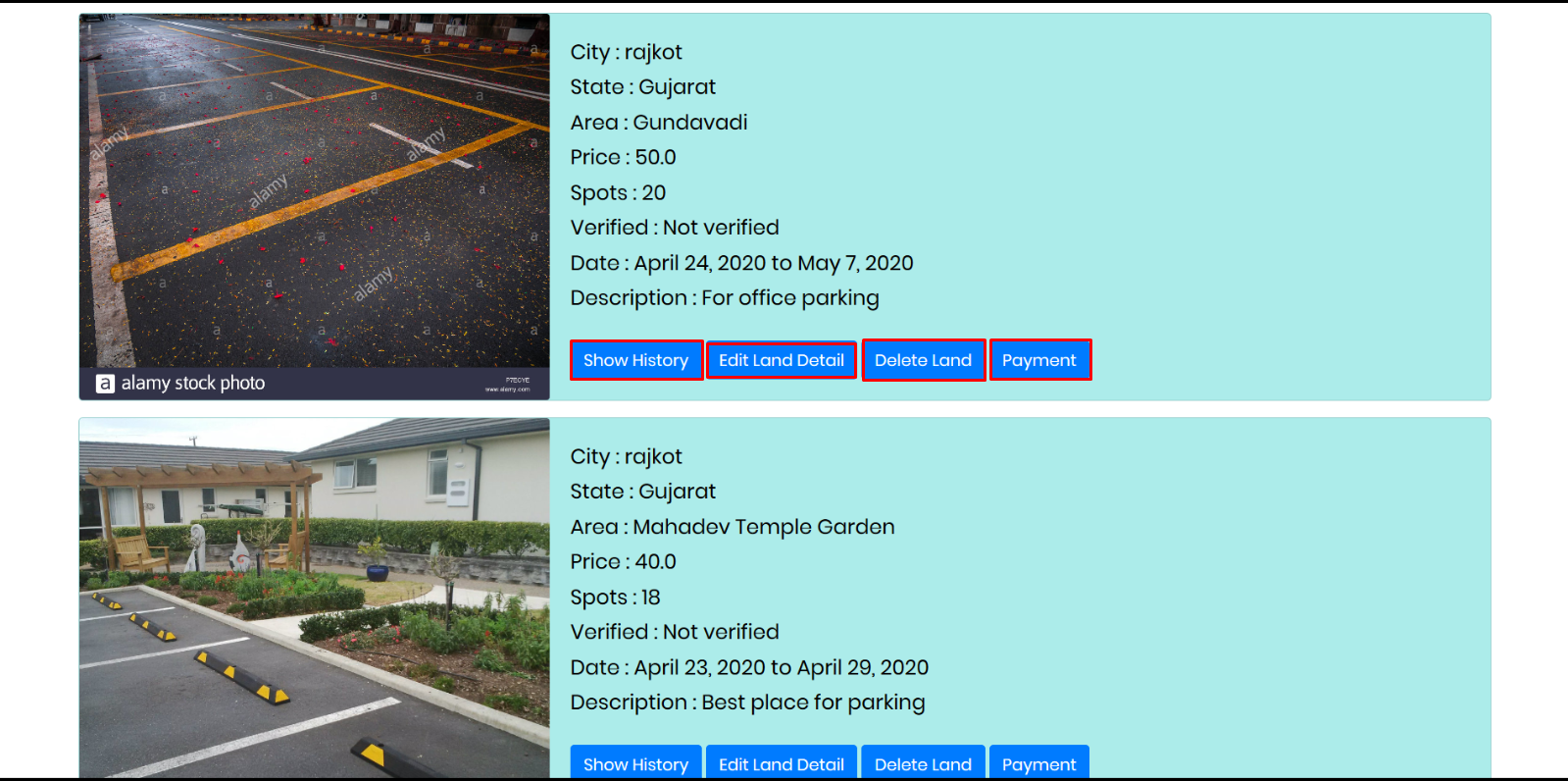
* Add Parking Place



* User History for land



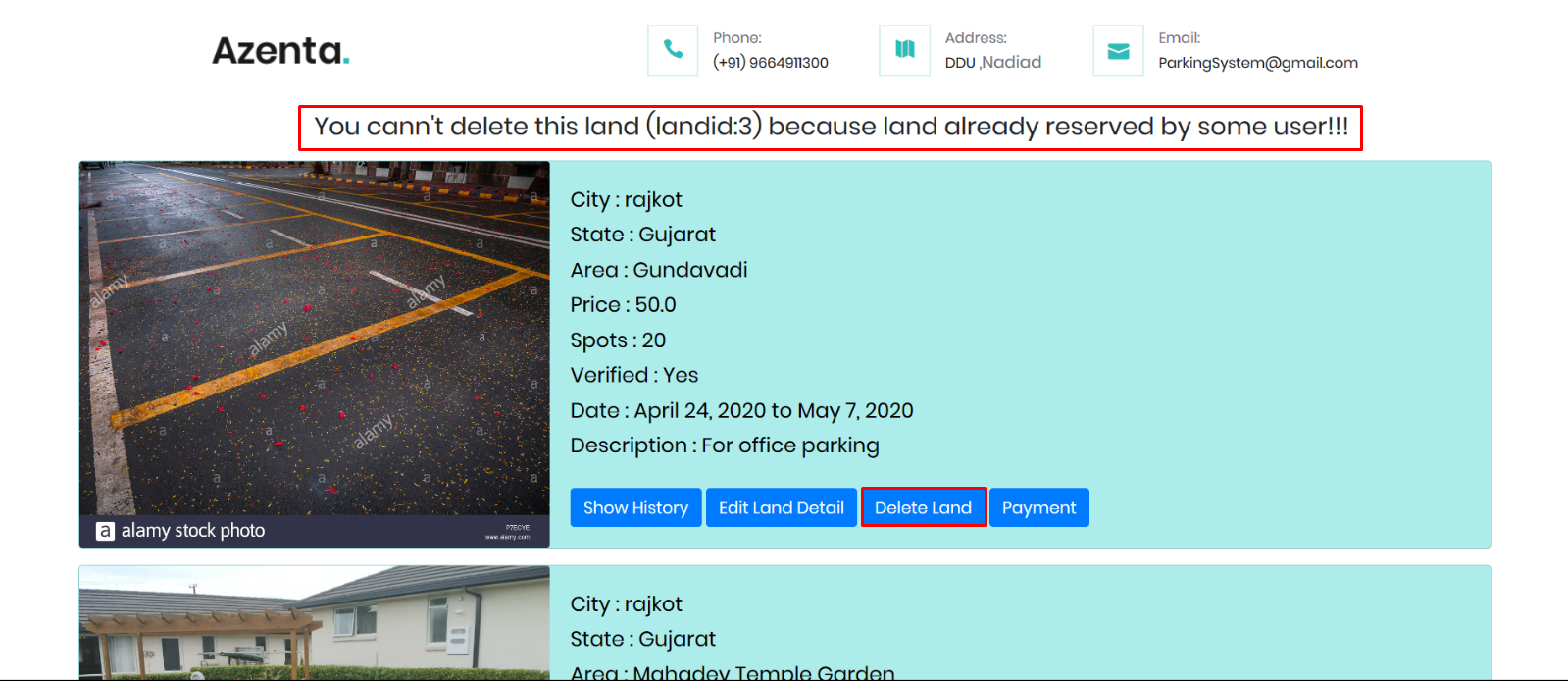
* Land Detail (Landlord)



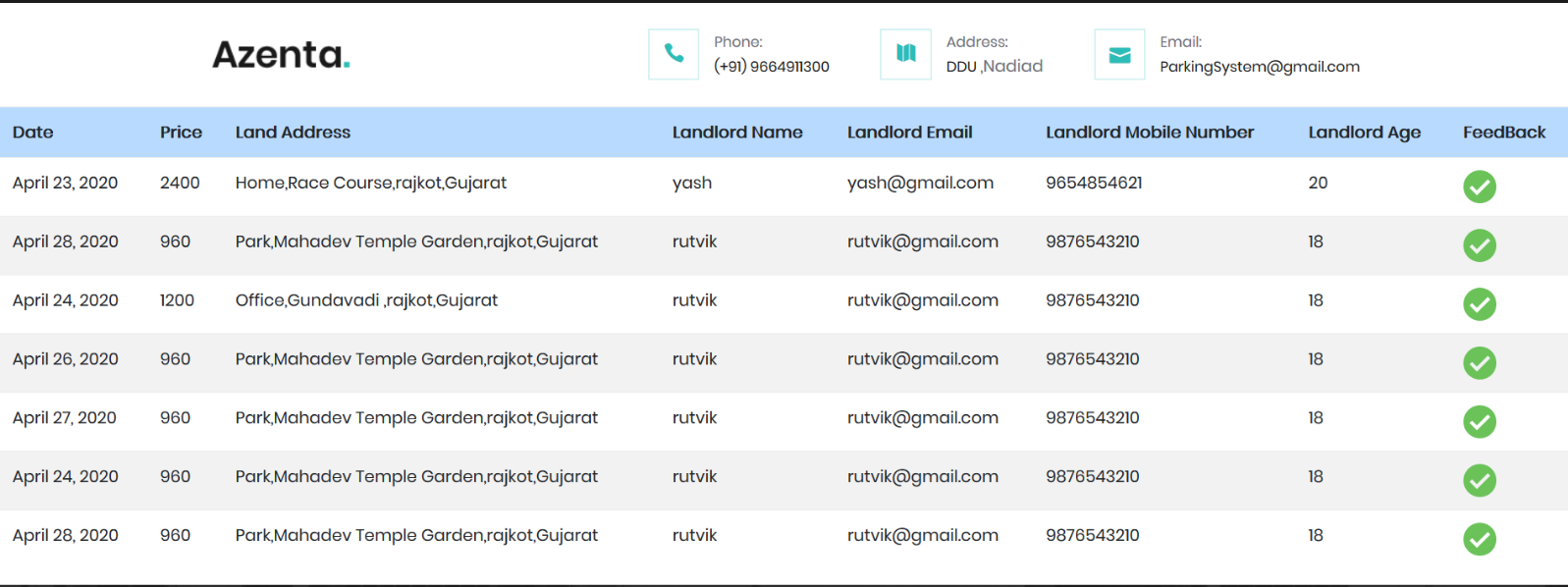
* Payment of land



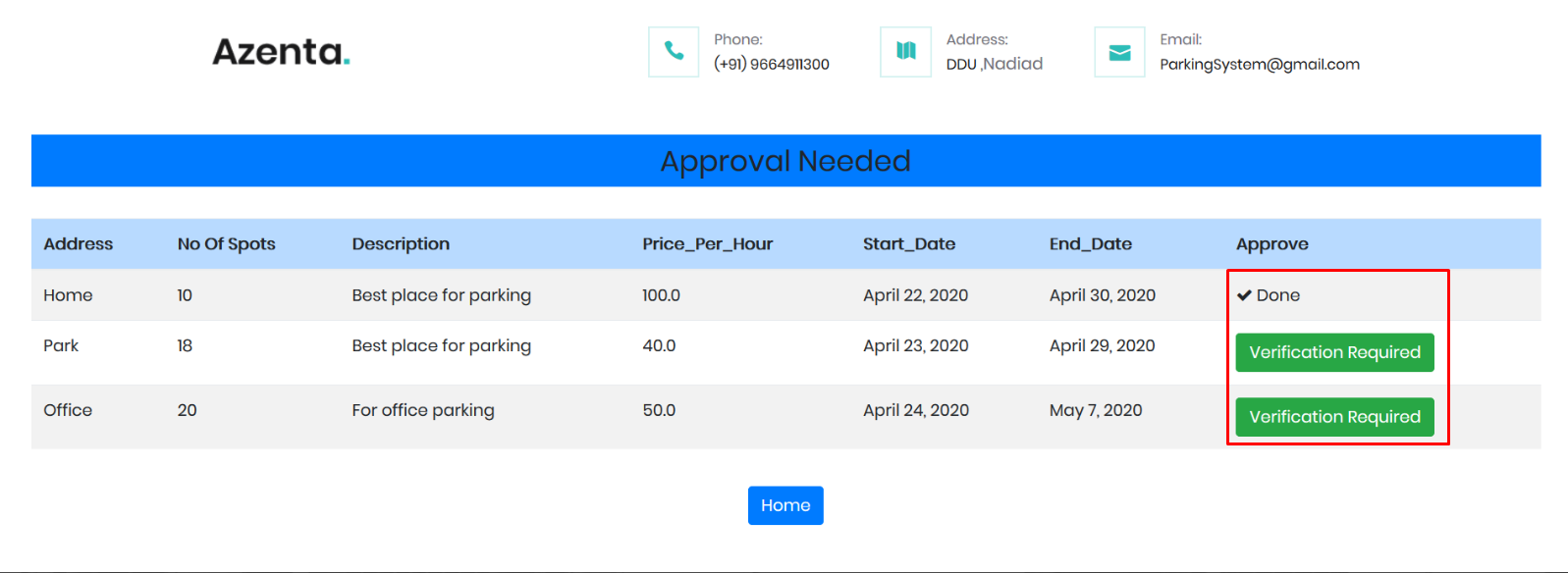
* Delete Land (Landlord)



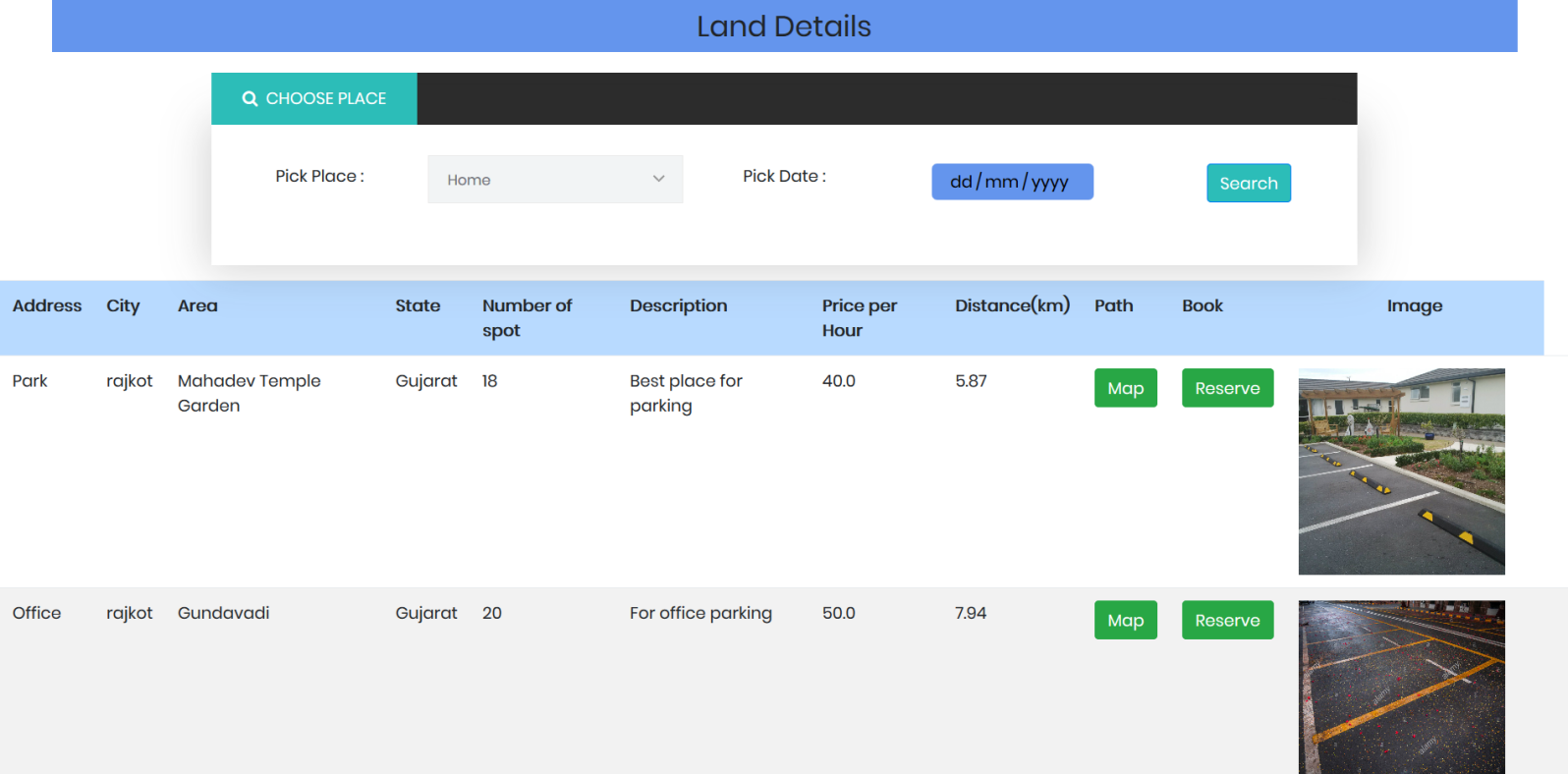
* History of Land(landlord)



* Land Verification (Admin)



* Advance Reservation (User)



* Reserve your favorite space (User)



**Conclusion**

The functionality implemented in system after understanding all the system modules according to the requirements.

**Functionality That are successfully implemented in the system are:**

* User registration with all the necessary validation on field
* Login
* Logout
* Forgot password with email verification
* Add land details
* Edit land details
* View usage history
* Delete land
* Get pay-out for land
* Verification of land by admin
* Reserve parking spot based on current location
* View reservation history
* Add location in advance
* Provide feedback

After the implementation and coding of system comprehensive testing was performed on the system to determine the errors and possible flaws in the system

**Limitation and Future Extension**

**Limitation**

* User can’t reserve parking spot for particular time interval.
* If user wants to add a location then he/she must be at that location. .

**Future Extension**

* SMS for reserved parking details
* Authentication using face detection
* Use of Google Map API.

**Bibliography**

* <https://www.python.org/doc/>
* <https://docs.djangoproject.com/en/3.0/>
* https://stackoverflow.com
* https://www.w3schools.com