

**Project Report on**  
**Online Parking Portal**



Submitted in partial fulfilment for the award of  
**Post Graduate Diploma in Advance Computing**  
**PG-DAC**

**Guided By:**

**Ms. Prajakta Patil.**

**Presented By:**

PRN	NAME
210930920043	Rathod Venkateshkumar Poonya
210930920042	Rathod Shital Subhash
210930920020	Kanse Akshay Balu
210930920019	Kajal Balasaheb Phapale
210930920002	Tiwari Akash

**Centre of Development of Advanced Computing (C-DAC), Pune**

**CERTIFICATE**

This is to certify that the project work under the title 'Online Parking Portal' is done by Rathod Venkateshkumar, Rathod Shital, Kanse Akshay, Kajal Phapale, AkashTiwari in partial fulfilment of the requirement for award of Diploma in Advanced Computing Course

**Mrs. Prajakta Patil**  
**Project Guide**

**Mr. Bhanu Pratap Singh**  
**Head of Department**

**Date:16<sup>th</sup> April 2022**

## ACKNOWLEDGEMENT

This project “**ONLINE PARKING PORTAL**” was truly a great learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC ACTS).

We are very glad to mention Ms. Prajakta Patil for her valuable guidance to work on this project. Her guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

We are highly grateful to Mr.Bhanu sir, Head of Department of Netcom training Centre, C-DAC, for her guidance and support whenever necessary during the course of our journey to acquire Post Graduate Diploma in *Advanced Computing (PG-DAC)* through C-DAC ACTS, Pune.

Our heartfelt thanks goes to Mr.Nishant sir, who gave all the required support and kind coordination to provide all the necessities.

## TABLE OF CONTENTS

Introduction.....	05
Project Overview.....	06
Purpose.....	06
Scope.....	06
Feasibility.....	06
Project Description.....	07
Technology Stack.....	07
Backend.....	07
Frontend.....	07
User classes.....	07
Admin.....	07
User.....	08
Architecture Diagram.....	08
Software Requirements Specification.....	09
Functional Requirements.....	09
Complete System.....	09
Use Case Diagram.....	09
Scenario 1: Mainline Sequence.....	10
Scenario 2: Mainline Sequence.....	10
Sequence in detail.....	11
ADMIN.....	11
CUSTOMER.....	12
Non-Functional Requirement.....	12
Performance Requirement.....	12
Security Requirement.....	13
Database Table.....	13
Entity Relationship Diagram.....	14
UML Diagram.....	15
DFD Diagram.....	15
Class Diagram.....	16
Activity Diagram.....	17
Admin Activity Diagram.....	17
User Activity Diagram.....	19
User Interface.....	20
Common Functionality.....	20
Home Page.....	20
Login Page.....	20
User Registration page.....	21

## Online Parking Portal

Registration Successful Page.....	21
Add Parking Facility.....	23
Booking List.....	23
Payment Details.....	23
Parking Facility List.....	23
Slot Booking.....	23
Payment Details.....	23
Parking Facility List.....	23
SlotBooking.....	23
Booking Details Table.....	25
Payment.....	26
User Details Table.....	26
Vehicle Info Table.....	27
References.....	29

### **Introduction :**

This project is about mobile management of parking areas. A parking lot or car park is a dedicated cleared area that is intended for parking vehicles. In most countries where cars are a major mode of transportation, parking lots are a feature of every city and suburban area. Shopping malls, sports stadiums, megachurches, and similar venues often feature parking lots over large areas. Parking is a problem is now a days in every big city it consumes a lot of time to find the right place to park your vehicles. It overcomes the problem of finding a parking space in commercial areas that unnecessary consumes time. So this system assist the user to online find and book the space for parking it will help the management to reduce the parking issue and also increase their revenue. This system will save the user time in search of parking areas and reduce the need of human resources in order to manage parking space. In this Web application we can access the information of parking slots in the building where is free. By finding the empty space the user is able to book the slot. This application also provides information about the user like Car No, License No. and mobile number. After selecting the empty space in the parking slot the user is able to pay the amount and confirm his/her booking.

### **2. Existing System :**

In the existing Online Parking Reservation systems, domestic and local level parking areas are not covered also parking for various vehicles are not preferred. So this makes the current system trivial and incomplete.

### **3. Proposed System :**

The proposed Online Parking Reservation system provides a smarter way for customer to select parking location, parking duration, extend parking duration. It also provides parking reservation for various types of vehicles under one parking portal.

### **4. Advantages :**

- 1) Users can get learn about parking areas for particular locations
- 2) It saves user time in search of parking space available in such a long parking area.
- 3) Cost-effective.

### **5. Assumptions :**

- 1) Vendor is available to provide service 24x7.
- 2) Services within one region/city.

### **6. Future Scope :**

- 1) Implementation of Live Location
- 2) App Development.

## Project Overview

### ▣ Purpose :

- √ The purpose of this project is to ease the process of parking through the web application. In this we mainly concentrate on parking slots in the building and the user is able to book the slot before entering into the building.

### ▣ Scope :

- √ A detailed study of the existing system is necessary. The functions of the system, requirements of the user, structure of the current system is made through the system study. The problems faced in the current system are found and solution pertaining to it is done in the system study.

### ▣ Feasibility :

- √ A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, and scheduling considerations—to ascertain the likelihood of completing the project successfully.
- √ A feasibility study is simply an assessment of the practicality of a proposed plan or project.
- √ The following feasibility studies were conducted to make sure that our software is feasible.

#### a) Technical feasibility

Technical feasibility includes the software's and hardware that are needed to develop the system. Software's and hardware have to be chosen according to the client requirements. We have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organisation currently processes the required technologies. Is the required technology available with the organization?

#### b) Operational feasibility

As per this study, we came to the conclusion that system is user friendly and easy to maintain.

This test of feasibility asks if the system will work when it is developed and installed.

The project offers a great deal of user experience and convenience to the target group.

#### c) Economical feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system. To develop the project from top to bottom, the estimated cost will come under this feasibility. It defines whether the client is able to pay the estimated cost or not. If the client is unable to spend then the software's have to be changed.

## Project Description

### ▣ Technology Stack

#### ➤ Backend

Category	Technology Name
Framework	Spring Boot
ORM Tool	Hibernate
Database	MySQL
Build Tool	Maven
Language	Java

#### ➤ Frontend

Category	Technology Name
Framework	ReactJs
Language	HTML, CSS, Javascript

### ▣ User Classes :

#### ➤ Admin

The super user, admin class represents complete authority over the system. An admin can,

- a) Add Parking details.
- b) Delete Parking details.
- c) Edit/Update Parking details.

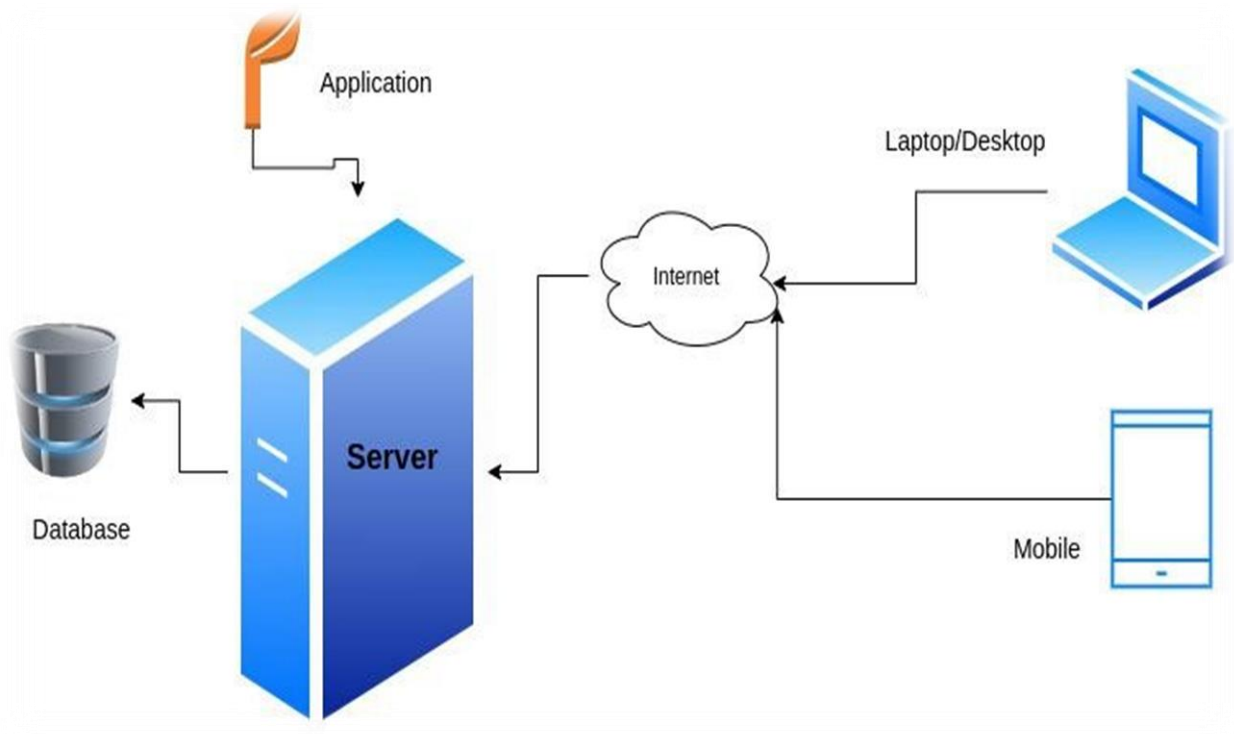
#### ➤ Customer

Once the customer register, Customer can,

- a) Add Vehicle details whether it is car or bike.
- b) View Parking Slots available in particular given area.
- c) Book a Slot for the specific time.
- d) Once the slot clicked then further confirm the booking.



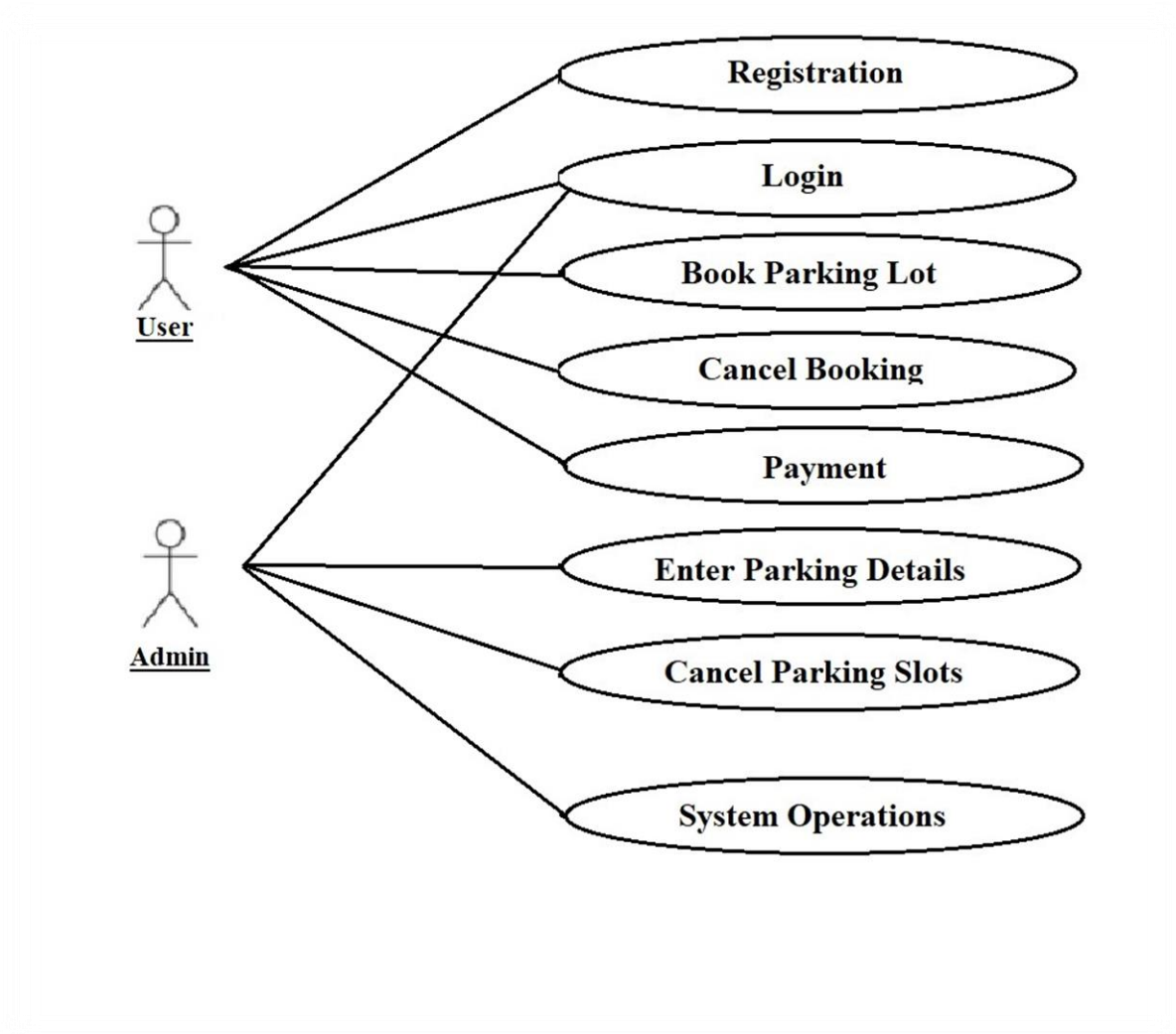
## Architecture Diagram



## Software Requirements Specification

### ▣ Functional Requirements

#### ➤ Complete System



*Figure 1: Use case diagram*

There is an entry interface that is intended to facilitate the actors [Admin|Vendor|Customer] to login to the system provided they have their own user account, i.e., already registered with the system . If not then register as Vendor or Customer. User has to enter the login credentials i.e. Username and Password information for Login.

### ◆ Scenario 1: Mainline Sequence

1. **Administrator:** Enter Admin Username and Password.
2. **System:**
  - a) Display the Admin dashboard where admin can approve request for vendors.
  - b) Can Fetch all the Pending facility List.

### ◆ Scenario 2: Mainline Sequence

1. **User:** Once registered, enter User Username and Password.
2. **System:**
  - a) Display the User dashboard where a User can add his/her Vehicle details.
  - b) User can Fetch Approved Parking Facility List.
  - c) And then after, Fetch User can Book a Parking Slot and confirmed booking.

### ➤ Sequence in detail

#### ◆ ADMIN

##### ❖ Add Parking Facility

Mainline Sequence:

1. **Admin:** Admin login.
2. **System:** Opens page which shows Add Parking Facility button.
3. **Admin:** Admin clicks on Add Parking Facility button.
4. **System:** Opens page which shows form to add Parking Facility details.
5. **Admin:** Fill the Information and click on the Add button.
6. **System:** Parking Facility get add to the Admin.

##### ❖ Delete Parking Facility

Mainline Sequence:

1. **Admin:** Admin login.
2. **System:** Opens page which shows Delete Parking Facility button.
3. **Admin:** Admin clicks on Delete Parking Facility button.
4. **System:** Opens page which shows message that Parking Facility details deleted successfully.

##### ❖ Edit/Update Parking Facility

Mainline Sequence:

7. **Admin:** Admin login.
8. **System:** Opens page which shows Edit Parking Facility button.
9. **Admin:** Admin clicks on Edit Parking Facility button.
10. **System:** Opens page which shows form to Edit Parking Facility details.
11. **Admin:** Fill the Information and click on the Edit/Update button.
12. **System:** Parking Facility get updated to the Admin.

### ❖ View Various Lists

Mainline Sequence:

13. **Admin:** Admin login.
14. **System:** Opens page where in Sidebar showing User details, Booking details, Parking details, Admin Payment details.
15. **Admin:** Admin clicks on any one by one and See the Lists.

### ◆ User

### ❖ Register User

Mainline Sequence:

1. **User:** User Signup.
2. **System:** Opens User Registration page.
3. **User:** User Enters Data with vehicle info.
4. **System:** After Click on register redirect to login page.

### ❖ Parking Facility List

Mainline Sequence:

1. **User:** User can See the List of Parking Facility.
- 2.
3. **System:** Opens page which shows search task for searching the available parking in particular Area.
4. **User:** User Enter the details of area and click on Search button.
5. **System:** Shows the Parking Facility in given Area.

### ❖ Book Slot

Mainline Sequence:

1. **User:** User clicks on Book Slot in Parking Facility List.
2. **System:** Opens page which shows form in which customer book the slot with StartDate and EndDate.
3. **User:** Customer Enter the details of StartDate and EndDate and click on Book button.
4. **System:** Shows the message that Booking is Confirmed.
5. **User:** Click on the Make Payment Button And Fill The Card details.
6. **System:** Shows the Payment Success in Admin Payment List.

### ▣ Non-Functional Requirement

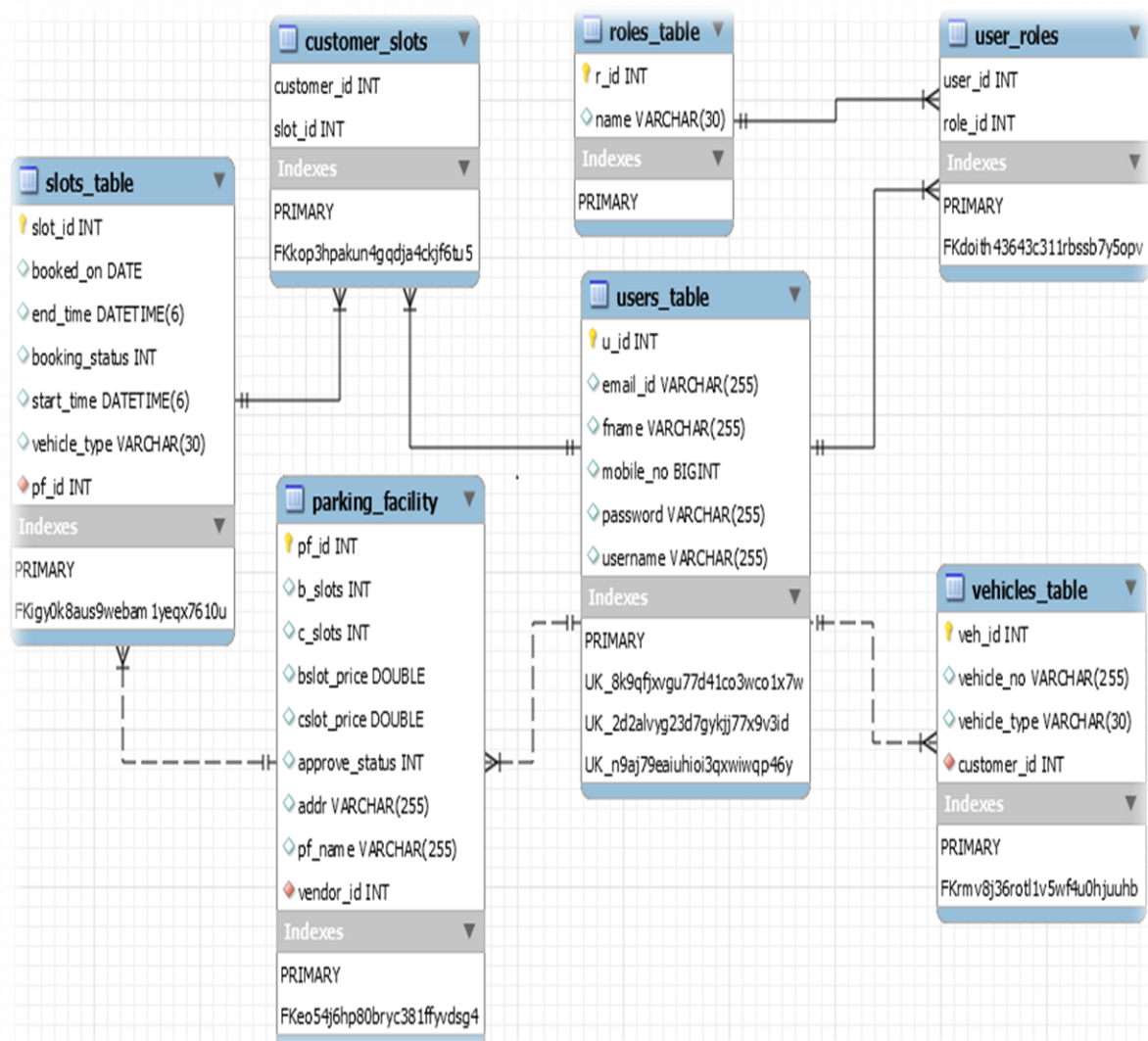
#### ➤ Performance Requirement

1. The time between request and response should be less.
2. Minimum time should be taken by the application to display the result.
3. In case of power failure, the data should be stored in the state that was last saved by the user.

## ➤ Security Requirement

1. One session per user
2. Passwords shall never be viewable at the point of entry or at any other time.
3. User are not allowed to update their own information.

## Database Table



## Entity Relationship Diagram

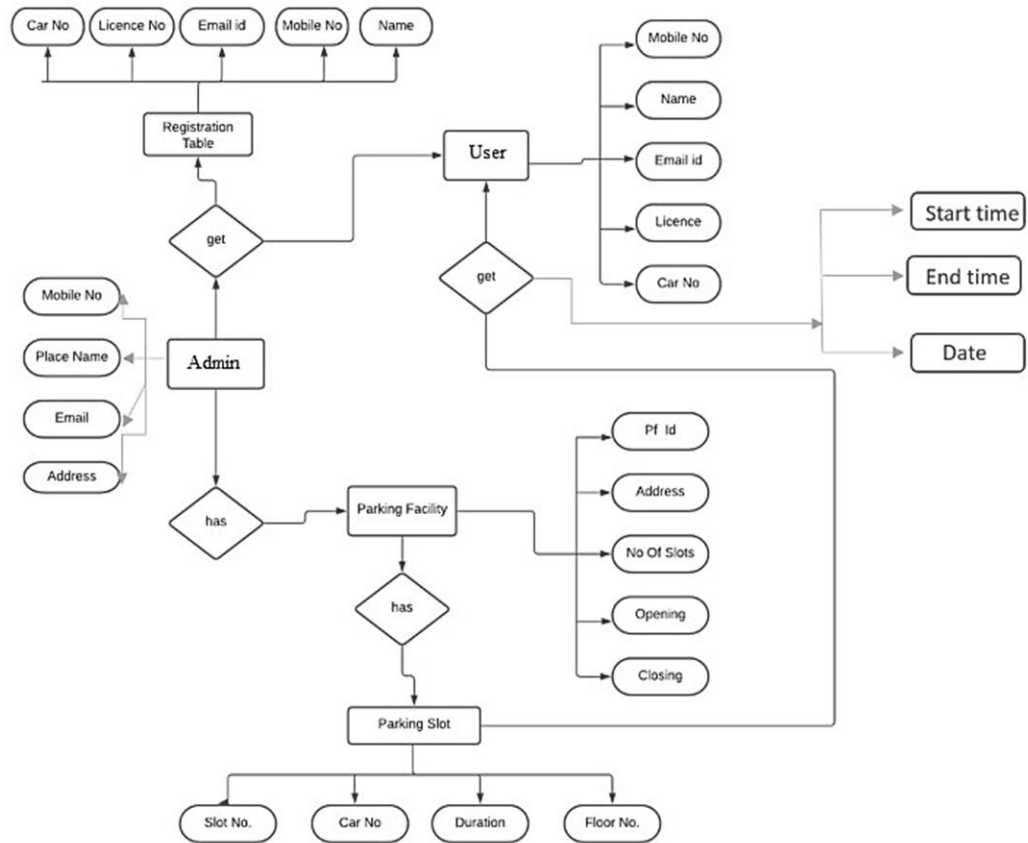


Figure - ER Diagram

## UML Diagram

### ▣ DFD Diagram

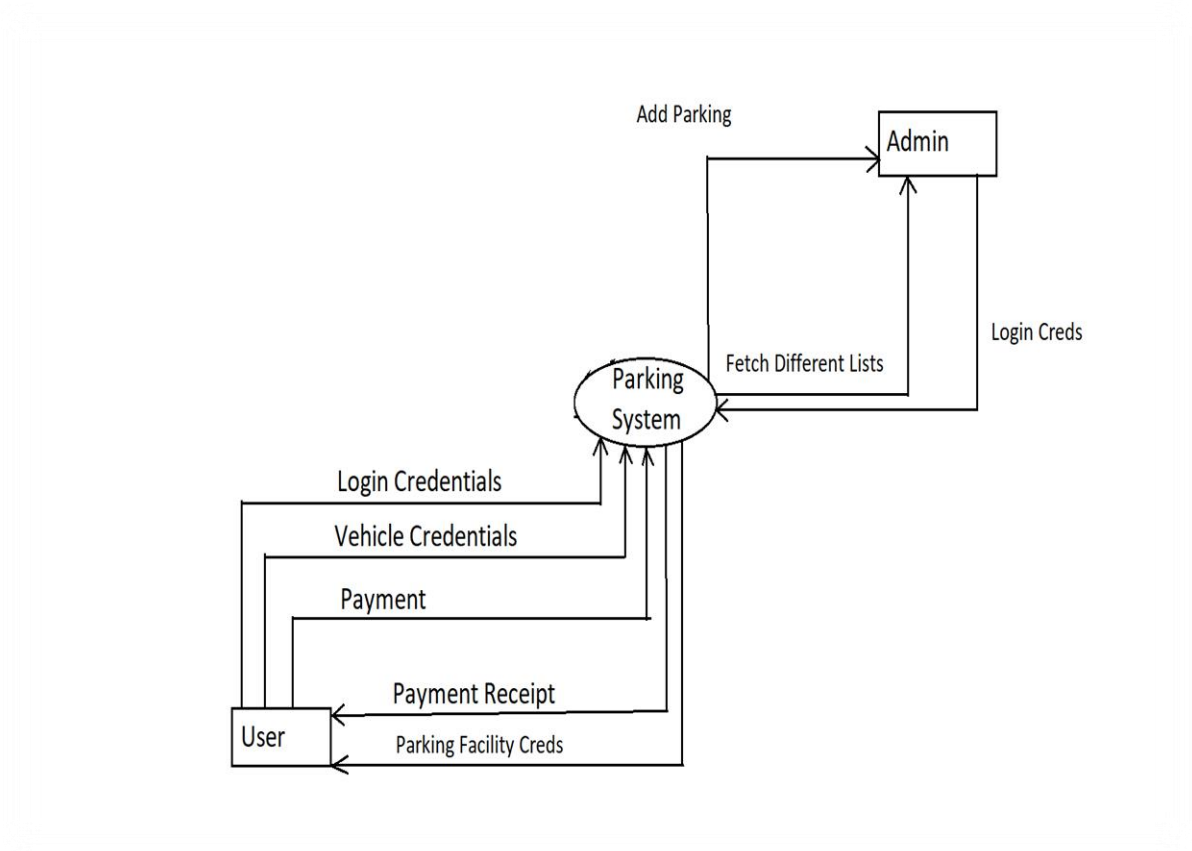


Figure - DFD Level Diagram

## Class Diagram

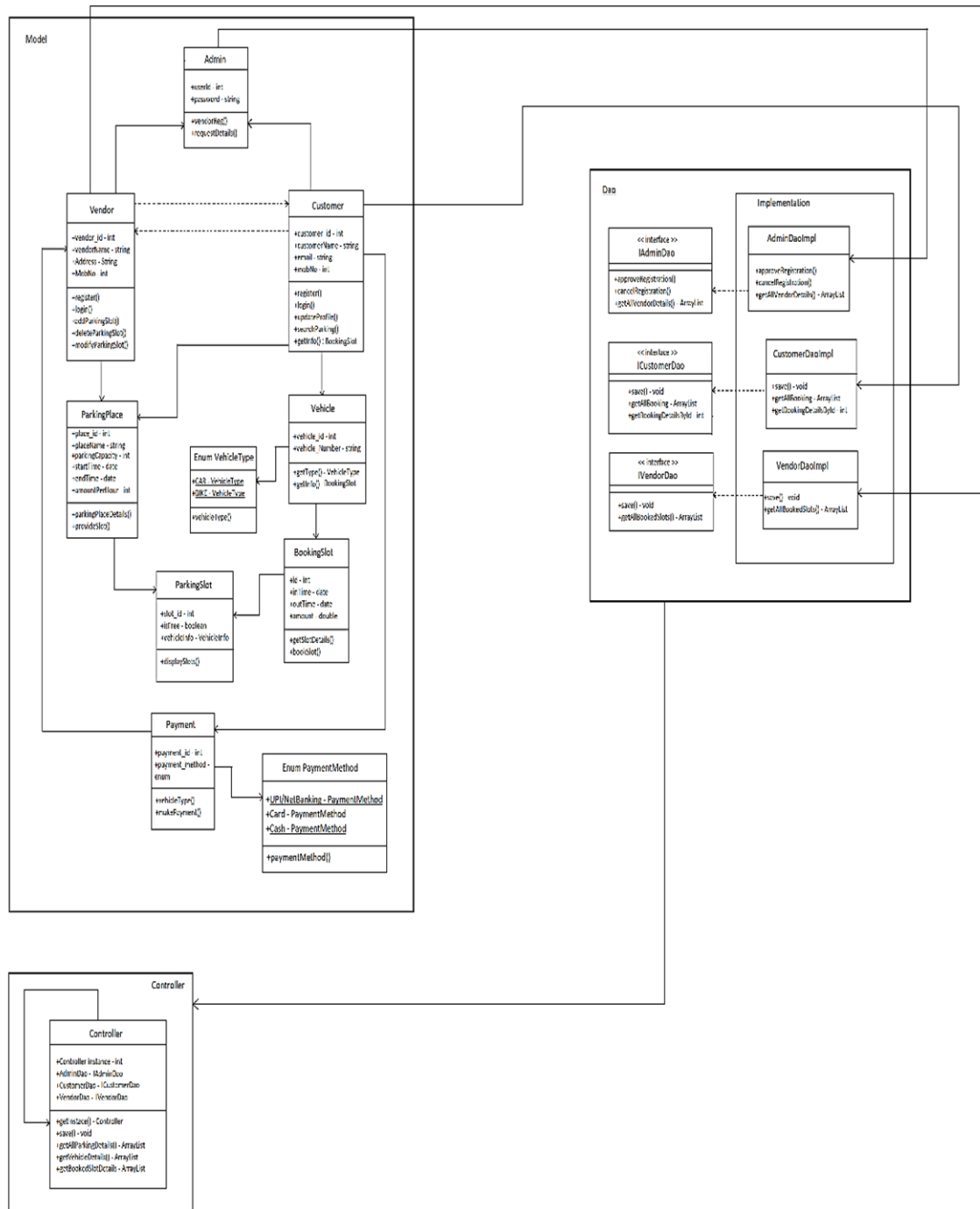
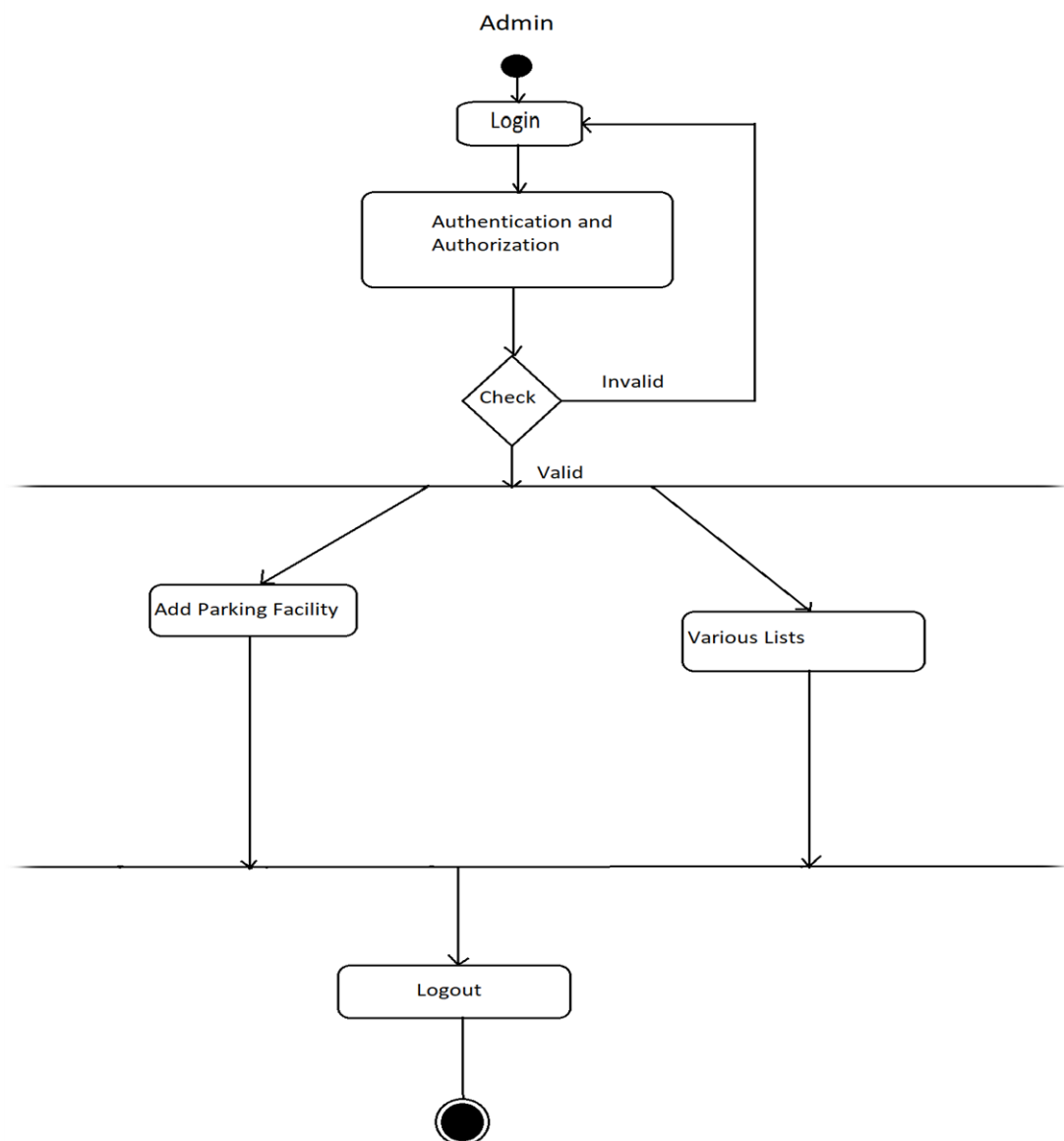


Figure - Class Diagram

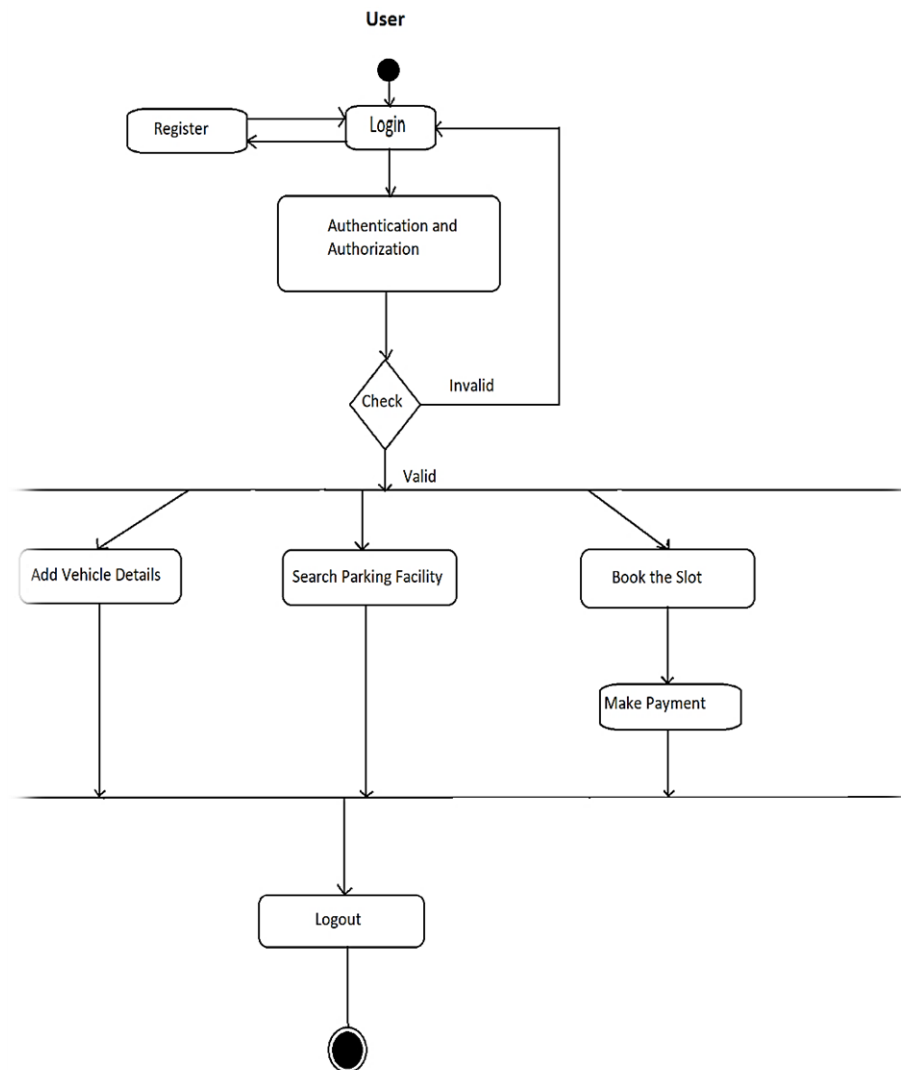


▣ Activity Diagram

➤ Admin Activity Diagram



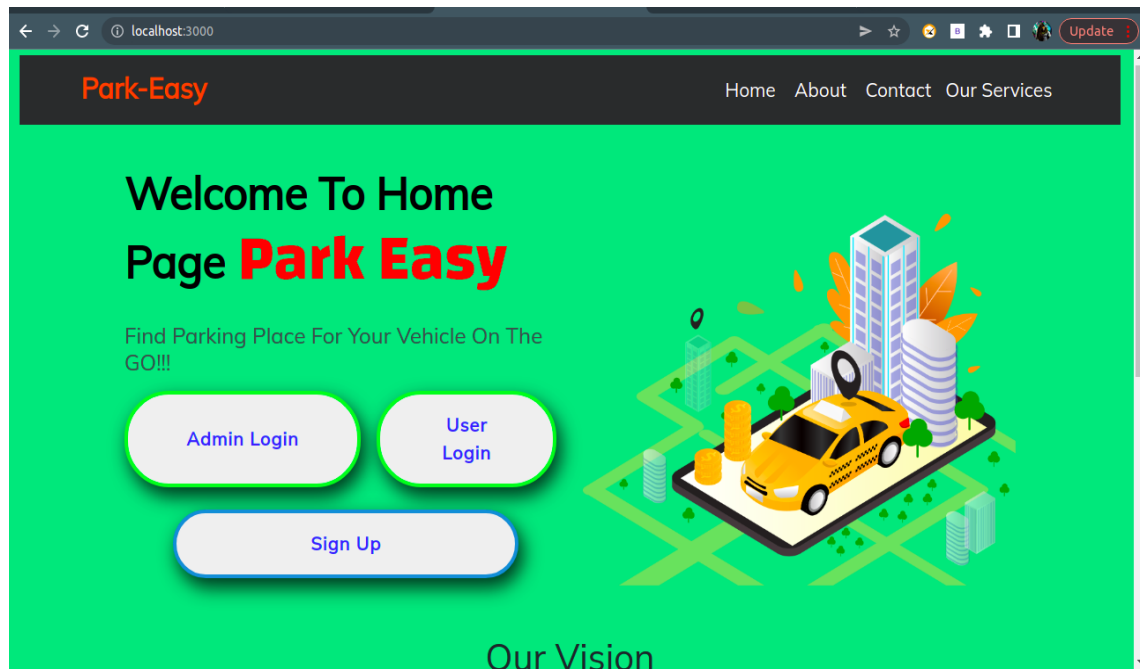
## ➤ User Activity Diagram



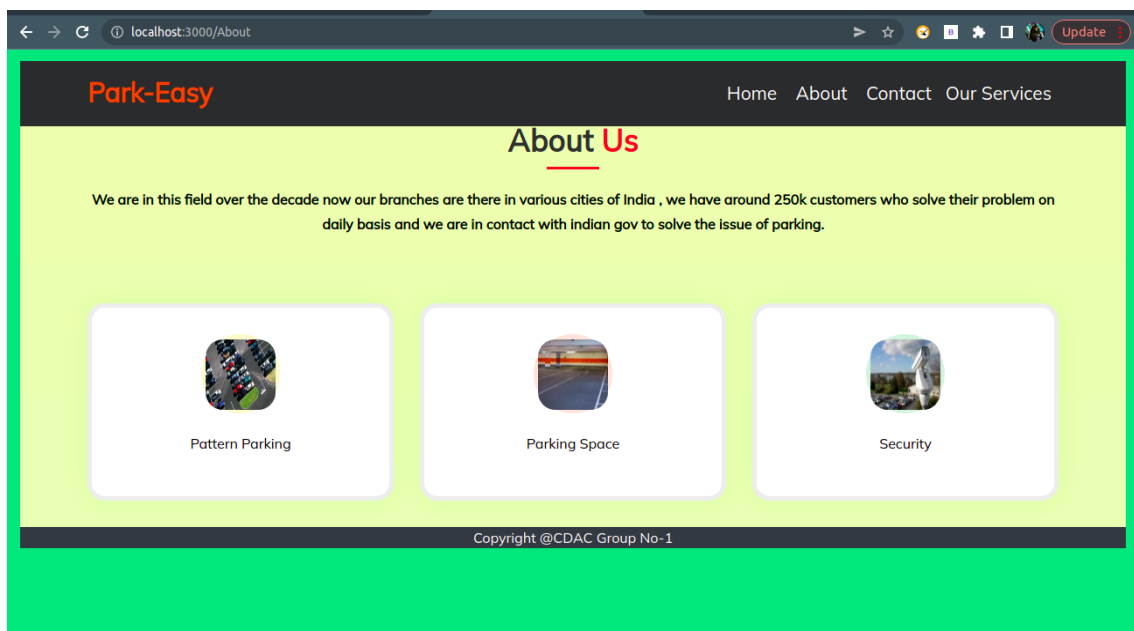
## User Interface

### Common Functionality

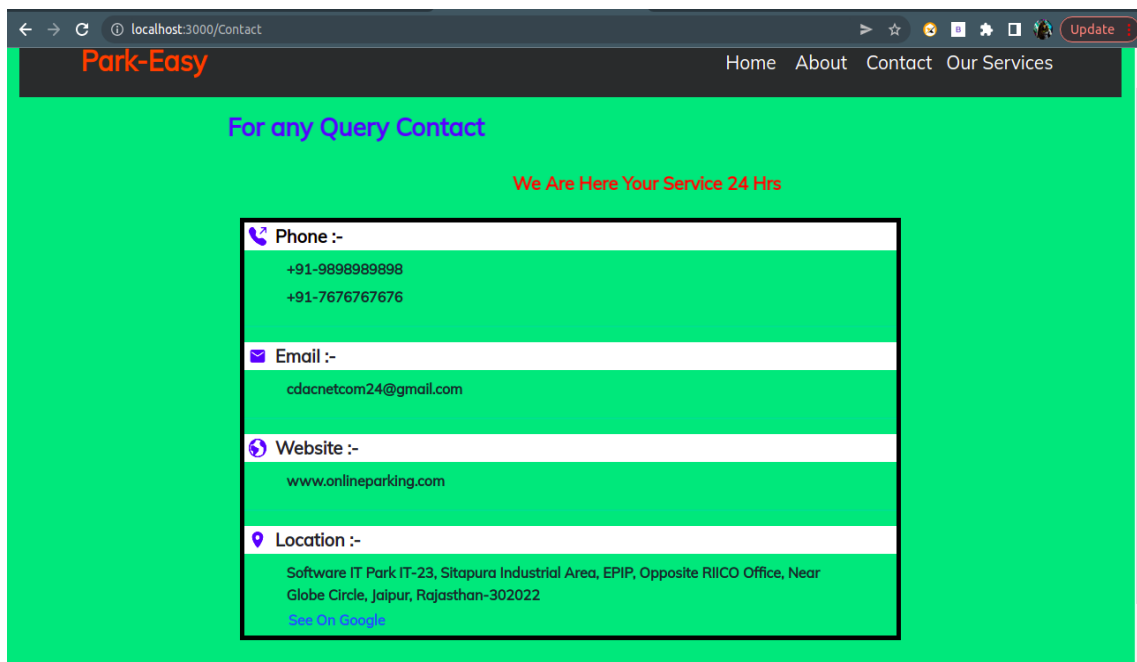
#### Home Page



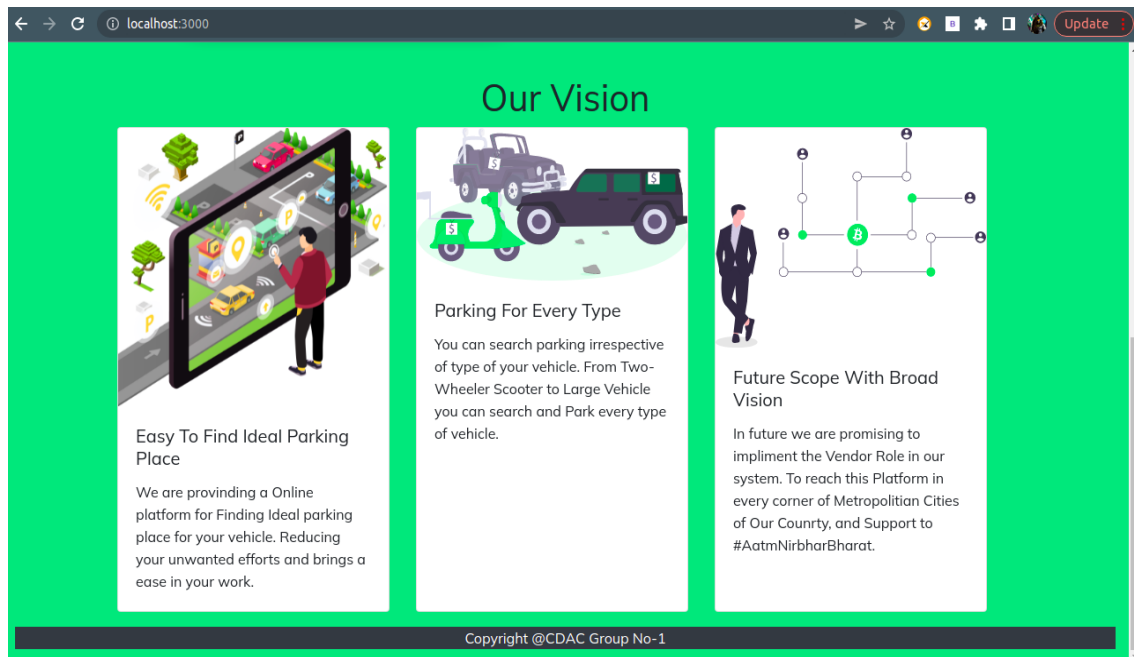
#### About Screen Page



## ➤ Contact Page

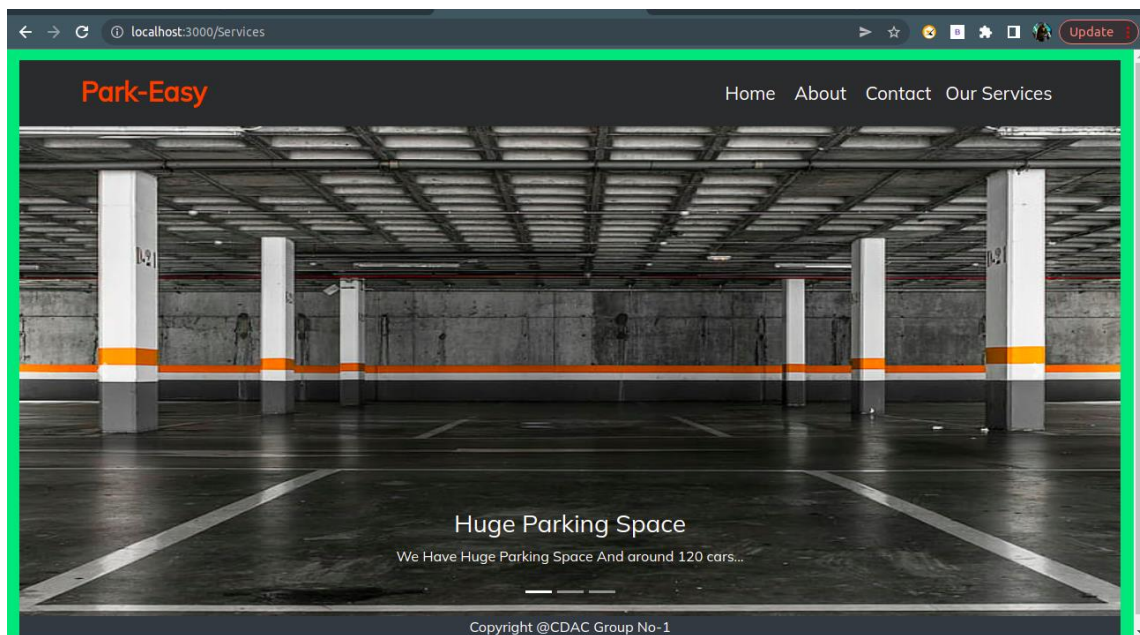


## ➤ Footer Page

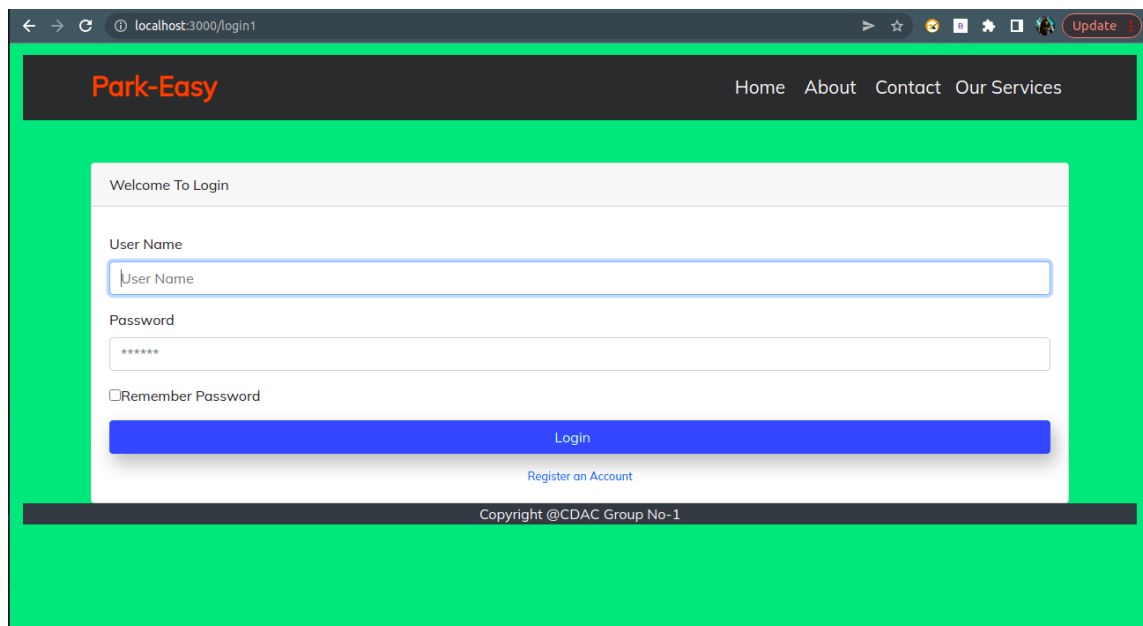


## Online Parking Portal

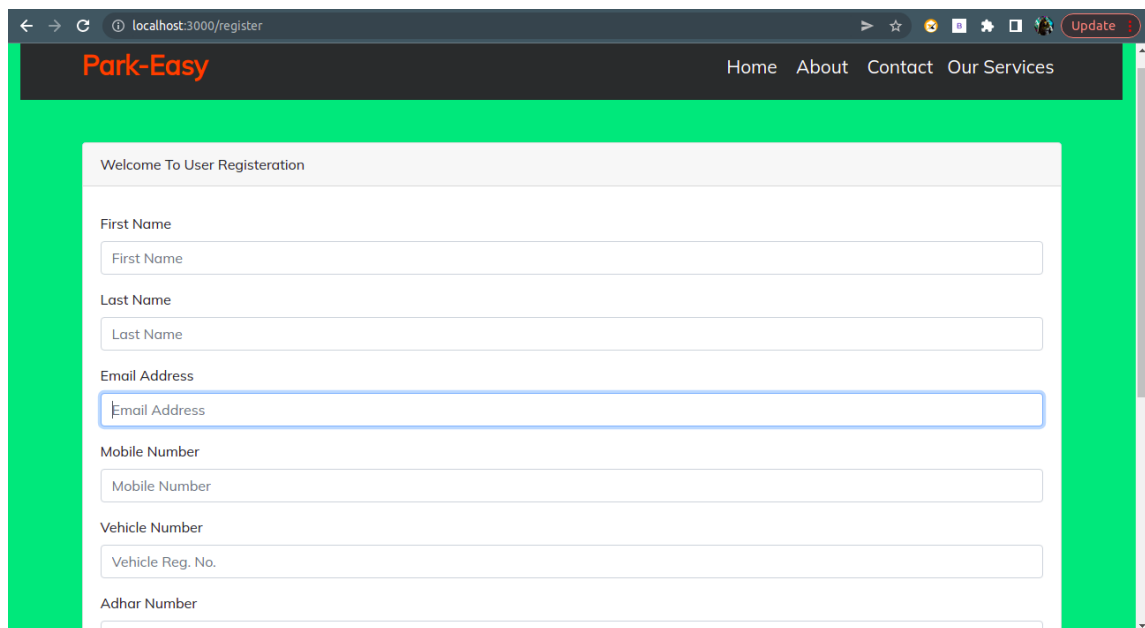
### ➤ Our Service Page



### ➤ Login Page

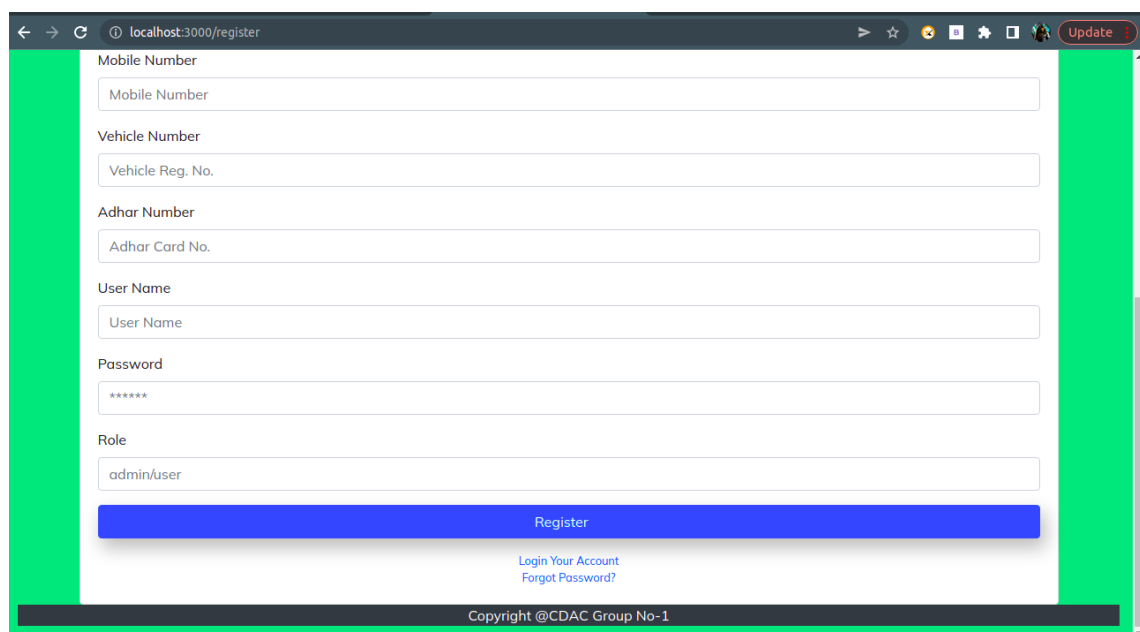


## ➤ User Registration Page 1



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/register'. The page has a dark header with the 'Park-Easy' logo in orange and navigation links: 'Home', 'About', 'Contact', and 'Our Services'. The main content area has a light green background and contains a registration form titled 'Welcome To User Registration'. The form includes the following fields: 'First Name', 'Last Name', 'Email Address' (which is highlighted with a blue border), 'Mobile Number', 'Vehicle Number', and 'Adhar Number'. Each field has a placeholder text indicating the type of information required.

## ➤ User Registration Page 2



The screenshot shows the second part of the user registration form. The browser address bar remains 'localhost:3000/register'. The form continues with the following fields: 'Mobile Number', 'Vehicle Number', 'Adhar Number', 'User Name', 'Password' (with a masked input showing '\*\*\*\*\*'), and 'Role' (with a dropdown menu showing 'admin/user'). Below these fields is a prominent blue 'Register' button. At the bottom of the form, there are two links: 'Login Your Account' and 'Forgot Password?'. The footer of the page displays 'Copyright @CDAC Group No-1'.

## Online Parking Portal

### ➤ Add Parking Facility Page

Parking Details / Add

Add Parking Slot

Parking Type

Enter Parking type

Enter No Of Slots Available

No Of Slots Available

Parking Address

Address

Contact Details

Enter Contact Details

Parking Name

Enter Name

Upload Parking Image

Choose File No file chosen

Enter Deposit Amount

Enter Deposite

Enter Rent

Enter Rent

Add Parking Slot

Logout

### ➤ Booking List

Park-Easy

Home About Contact Our Services

- User Details
- Booking Details
- Parking Details
- Admin Payment Details

Booking Details

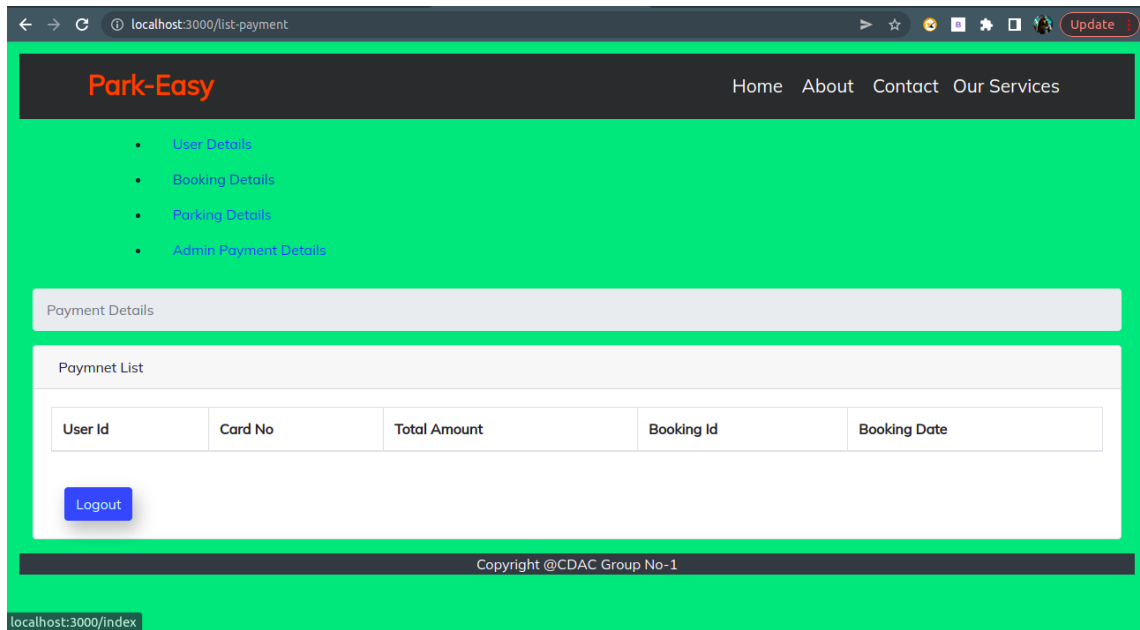
Booking List

Booking Id	Deposit Amount	Start Date	End Date	Charges	Total Amount	Parking Id	Status
------------	----------------	------------	----------	---------	--------------	------------	--------

Logout

Copyright @CDAC Group No-1

## ➤ Payment Details



The screenshot shows a web browser at localhost:3000/list-payment. The page has a green header with the 'Park-Easy' logo and navigation links: Home, About, Contact, and Our Services. A sidebar on the left contains links: User Details, Booking Details, Parking Details, and Admin Payment Details. The main content area is titled 'Payment Details' and contains a 'Paymnet List' table. The table has five columns: User Id, Card No, Total Amount, Booking Id, and Booking Date. Below the table is a 'Logout' button. The footer shows 'Copyright @CDAC Group No-1' and the browser address bar shows 'localhost:3000/index'.

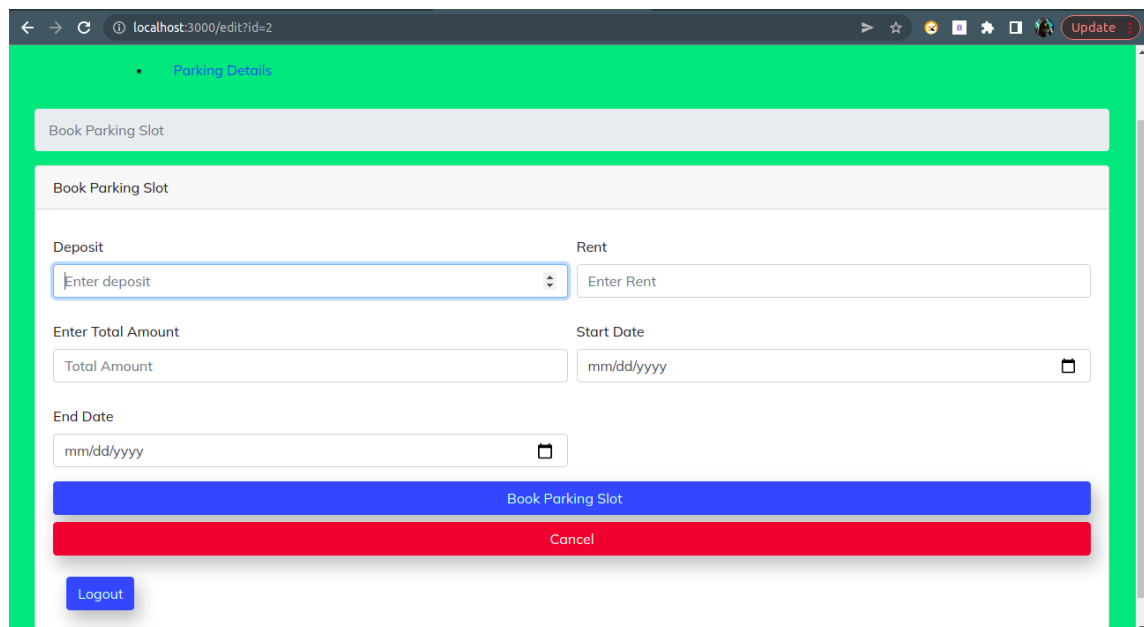
Payment Details

User Id	Card No	Total Amount	Booking Id	Booking Date
---------	---------	--------------	------------	--------------

Logout

Copyright @CDAC Group No-1

## ➤ Booking Parking Facility



The screenshot shows a web browser at localhost:3000/edit?id=2. The page has a green header with the 'Park-Easy' logo and navigation links: Home, About, Contact, and Our Services. A sidebar on the left contains links: User Details, Booking Details, Parking Details, and Admin Payment Details. The main content area is titled 'Book Parking Slot' and contains a form with the following fields: Deposit (text input), Rent (text input), Enter Total Amount (text input), Start Date (date picker), End Date (date picker), and a 'Logout' button. Below the form are two buttons: 'Book Parking Slot' (blue) and 'Cancel' (red).

Book Parking Slot

Deposit

Enter deposit

Rent

Enter Rent

Enter Total Amount

Total Amount

Start Date

mm/dd/yyyy

End Date

mm/dd/yyyy

Book Parking Slot

Cancel

Logout



## ➤ Booking Parking List

The screenshot shows a web browser at localhost:3000/booking-details. The page has a green header with the 'Park-Easy' logo and navigation links: Home, About, Contact, Our Services. Below the header is a green sidebar with a 'Parking Details' link. The main content area is titled 'Parking Facility List' and contains a table with the following data:

Parking Id	Parking Type	Parking Slots	Parking Address	Parking Name	Contact Number	Deposit	Rent	Action	Action	Action
1	car	8	block - b	compound	5895	50	1890	<button>Book</button>	<button>Update</button>	<button>Delete</button>
2	bike	2	pune	akshay	831852	2000	200	<button>Book</button>	<button>Update</button>	<button>Delete</button>

Below the table is a 'Logout' button. At the bottom of the page, there is a copyright notice: 'Copyright @CDAC Group No-1'.

## ➤ Booking\_dtls Table:-

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'carparkingdb' database with a tree view of its objects. The 'booking\_dtls' table is selected, and its schema is shown in the 'Table: booking\_dtls' section. The schema details are as follows:

Field	Type	Null	Key	Default	Extra
booking_id	int	NO	PRI	NULL	auto_increment
created_date	datetime	YES		NULL	
cust_id	int	YES		NULL	
deposit	double	YES		NULL	
from_date	date	YES		NULL	
rent	double	YES		NULL	
start_date	date	YES		NULL	
status	varchar(10)	NO		NULL	
total_amount	double	YES		NULL	
vehical_id	varchar(255)	YES		NULL	

The 'Query 1' window shows the following SQL script:

```

1 • create database carParkingDB;
2
3 • use carParkingDB;
4 • select * from user_dtls;
5
6 • desc booking_dtls;
7

```

The 'Result Grid' shows the output of the 'desc booking\_dtls' query, which is the table schema shown above. The 'Action Output' window shows the results of the 'select \* from user\_dtls' query, with 4 rows returned.

## Online Parking Portal

### ➤ Payment Table:-

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'carparkingdb' expanded, showing tables 'booking\_dtl', 'payment', and 'user\_dtl'. The 'payment' table is selected, and its columns are listed: payment\_id (int, AI, PK), amount (double), booking\_id (int), card\_no (varchar(33)), created\_date (datetime), and cvv (varchar(33)). The main query editor shows the following SQL script:

```
1 • create database carParkingDB;
2
3 • use carParkingDB;
4 • select * from user_dtl;
5
6 • desc payment ;
7
```

The 'Result Grid' shows the structure of the 'payment' table:

Field	Type	Null	Key	Default	Extra
payment_id	int	NO	PRI		auto_increment
amount	double	NO			
booking_id	int	YES			
card_no	varchar(33)	NO			
created_date	datetime	YES			
cvv	varchar(33)	NO			

The 'Action Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
6	09:57:53	select * from user_dtl LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
7	11:25:26	desc booking_dtl	10 row(s) returned	0.062 sec / 0.000 sec
8	11:26:34	desc payment	6 row(s) returned	0.109 sec / 0.000 sec

### ➤ user\_dtl Table:-

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'carparkingdb' expanded, showing tables 'booking\_dtl', 'payment', and 'user\_dtl'. The 'user\_dtl' table is selected, and its columns are listed: user\_id (int, AI, PK), adhar\_no (varchar(255)), created\_date (datetime), email (varchar(256)), first\_name (varchar(33)), last\_name (varchar(33)), license\_no (varchar(255)), mob\_no (varchar(13)), password (varchar(255)), status (varchar(10)), user\_name (varchar(255)), and user\_role (varchar(255)). The main query editor shows the following SQL script:

```
1 • create database carParkingDB;
2
3 • use carParkingDB;
4 • select * from user_dtl;
5
6 • desc user_dtl ;
7
```

The 'Result Grid' shows the structure of the 'user\_dtl' table:

Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI		auto_increment
adhar_no	varchar(255)	YES			
created_date	datetime	YES			
email	varchar(256)	NO			
first_name	varchar(33)	NO			
last_name	varchar(33)	NO			
license_no	varchar(255)	YES			
mob_no	varchar(13)	NO			
password	varchar(255)	NO			
status	varchar(10)	NO			
user_name	varchar(255)	NO			
user_role	varchar(255)	YES			

The 'Action Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
8	11:26:34	desc payment	6 row(s) returned	0.109 sec / 0.000 sec
9	11:27:24	desc user_dtl	12 row(s) returned	0.062 sec / 0.000 sec

## ➤ Vehicle\_info Table:-

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'user\_dtl' database selected, with the 'vechical\_info' table highlighted. Below this, the 'Table: vechical\_info' structure is shown with columns: vechical\_id (int, AI PK), color (varchar(255)), company (varchar(255)), created\_date (datetime), deposit (double), image (varchar(255)), mob\_no (varchar(255)), model\_no (varchar(255)), rent (double), running\_status (double), vechical\_status (varchar(10)), and vechical\_type (varchar(255)).

The central 'Query 1' pane contains the following SQL code:

```
3 • use carParkingDB;
4 • select * from user_dtl;
5
6
7 • desc vechical_info ;
```

The 'Result Grid' pane shows the table structure for 'vechical\_info' with columns: Field, Type, Null, Key, Default, and Extra. The table has 12 columns in total.

The 'Output' pane at the bottom shows an error message:

```
16 11:29:05 desc vechical_info Error Code: 1146. Table 'carparkingdb.vechical_info' doesn't exist
17 11:29:34 desc vechical_info 12 row(s) returned
```

An 'Activate Windows' watermark is visible in the bottom right corner of the screenshot.

**REFERENCES:**

<http://www.google.com>

<http://www.wikipedia.org>

<https://unsplash.com/s/photos/car-park>

<https://reactis.org/>

