



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BENGALURU, KARNATAKA, INDIA.**

B. TECH. (CSE) - V SEMESTER

AUG. – DEC. 2024

**UE22CS351A – DATABASE MANAGEMENT SYSTEM PROJECT
REPORT
ON
ONLINE VEHICLE BOOKING MANAGEMENT SYSTEM**

**Submitted by:
PES1UG23CS846 - Sanket Muttur
PES1UG23CS847 - Veeresh Amaragatti**

Abstract:

The **Online Vehicle Booking Management System** (OVBMS) is a web-based platform designed to facilitate vehicle rentals, offering users a streamlined process to search, book, and manage vehicle reservations. The system provides a user-friendly interface for customers to register, browse available vehicles, make bookings, and receive confirmations, while administrators are empowered with tools to manage vehicle listings, user accounts, and bookings. The OVBMS leverages a MySQL database for data management, with RESTful APIs ensuring smooth data communication. Secure data transmission is maintained via HTTPS. This platform addresses the need for a reliable, secure, and scalable vehicle rental solution that enhances user experience and operational efficiency for both customers and administrators.

Table of Contents

1. User Requirement Specification	1
1.1. Purpose of the Project	1
1.2. Scope of the Project	1
1.3. Detailed Description of the Project.....	2
2. List of Softwares/Tools/Programming languages used.....	3
3. ER Diagram	4
4. Relational Schema	4
5. DDL Commands for creating Tables in OVBMS	5
6. CRUD operation Screenshots	7
6.1. CREATE: Adding Vehicle.....	7
6.2. READ: Vehicle Listing.....	10
6.3. UPDATE: Updating Vehicle Information	11
6.4. DELETE: Deleting the Vehicle	15
7. List of functionalities/features of the application.....	17
8. Triggers, Procedures/Functions, Nested query, Join, Aggregate queries.....	26
8.1. Procedure 1	26
8.2. Procedure 2	27
8.3. Trigger.....	28
8.4. Event	29
8.5. Nested Query & Aggregate Query.....	30
8.6. Join Query.....	31
9. Code snippets for invoking the Procedures/Functions/ Trigger.....	32
9.1. Invoking “BookVehicle” Procedure:	32
9.2. Invoking “HandleBookingRequest” Procedure:	32
10. SQL queries (Create, Insert, Triggers, Procedures/ Functions, Nested query, Join, Aggregate queries) used in the project in the form of .sql file	33
11. GitHub Repo Link.....	42

1. User Requirement Specification

1.1. Purpose of the Project

The purpose of this project is to design and implement a Vehicle Rental Management System aimed at providing an integrated platform for customers to reserve and rent vehicles efficiently. The system will allow customers to browse available vehicles, make reservations, while enabling managers to maintain the vehicle inventory, approve or reject booking requests.

This system will automate the vehicle rental process, making it more accessible for customers and easier to manage for rental service providers. By replacing manual booking, the system enhances both operational efficiency and customer satisfaction.

1.2. Scope of the Project

The scope of the **Vehicle Rental Management System** includes the development and deployment of a web-based application that facilitates the end-to-end vehicle rental process. It covers the functionalities required by both customers and managers, ensuring that all aspects of the rental service are effectively managed.

Key Features:

- **Customer Functions:**

- **Account Registration:** Customers can create accounts, enter personal details such as name, password, email, and license ID.
- **Vehicle Reservation:** Customers can view available vehicles, reserve vehicles for specific dates, and track their bookings.

- **Manager Functions:**

- **Vehicle Management:** Managers can register new vehicles into the system, including specifications like seating capacity, fuel type, model year, and rental price.
- **Booking Management:** Managers are responsible for approving or rejecting customer booking requests based on vehicle availability and service conditions.

The system is built to accommodate a wide range of rental operations, from small fleets to large-scale car rental companies. It is designed to be scalable and extendable, with potential future additions such as online payment integration, vehicle maintenance tracking, and automated notifications.

1.3. Detailed Description of the Project

The **Vehicle Rental Management System** consists of several interrelated components, designed to ensure smooth and efficient operation of a vehicle rental service. Below is a detailed breakdown of the system's entities and processes:

The Entity-Relationship (ER) Diagram illustrates the core entities in the system and their interactions. These include:

- **Manager:**

The manager is responsible for overseeing vehicle registrations, approving or rejecting customer bookings. Managers interact with the system through functionalities such as:

- **Vehicle Registration:** Add and manage vehicles in the rental fleet.
- **Booking Request Approval:** Approve or reject booking requests from customers.

- **Customer:**

Customers can register accounts, view vehicle availability, make bookings. The customer interacts with the system via:

- **Registration:** Provide personal details, including name, email, password, and license ID.
- **Vehicle Reservation:** Browse available vehicles and place reservations by specifying booking dates.

- **Vehicle:**

The system keeps detailed records of each vehicle in the fleet, including its name, model year, seating capacity, fuel type, price per day, and other relevant attributes.

- **Booking**

- Request:**

Customers place booking requests for specific vehicles, specifying a date range. Each request is reviewed and either approved or rejected by a manager. The booking process includes:

- **Request Date:** The date when the booking request was submitted.
- **Rental Period:** The start and end dates of the booking.
- **Approval Status:** Indicates whether the request has been approved or rejected by the manager.

Functional Requirements for the Online Vehicle Booking Management System:

1. Customer Registration and Login

- Customers can create accounts and log in using credentials, including personal details like name, email, password, and license ID.

2. Vehicle Browsing and Availability Check

- Customers can browse the available vehicles and check their availability for specific dates.

3. Vehicle Reservation

- Customers can reserve vehicles for specific dates, selecting from the available fleet.

4. Booking Management

- Managers can view booking requests, approve or reject them based on vehicle availability and service conditions.

5. Vehicle Inventory Management

- Managers can add, update, or remove vehicle details, including attributes like seating capacity, fuel type, model year, and rental price.

2. List of Softwares/Tools/Programming languages used

1. Frontend: ReactJS

- For building the user interface of the web application.

2. Backend: Node.js, Express.js

- For handling server-side logic and APIs.

3. Database: MySQL

- To store and manage data such as user information, vehicle details, and bookings.

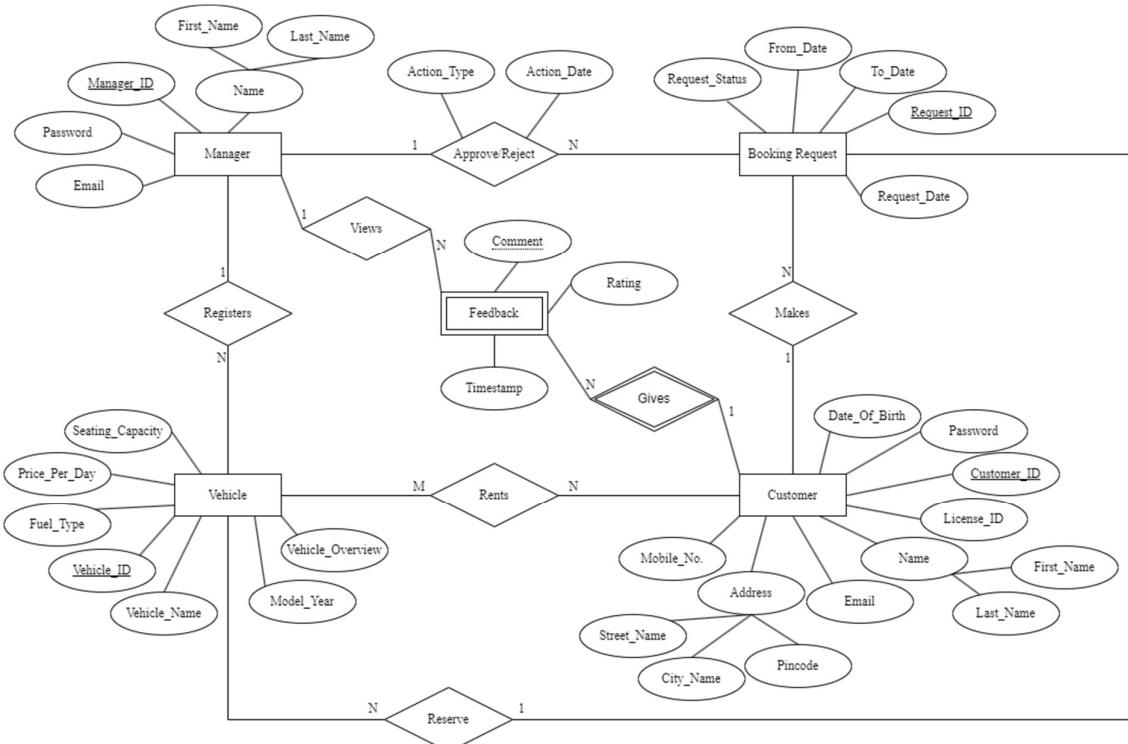
4. IDE: Visual Studio Code (Visual Studio Code)

- Used for coding and project development.

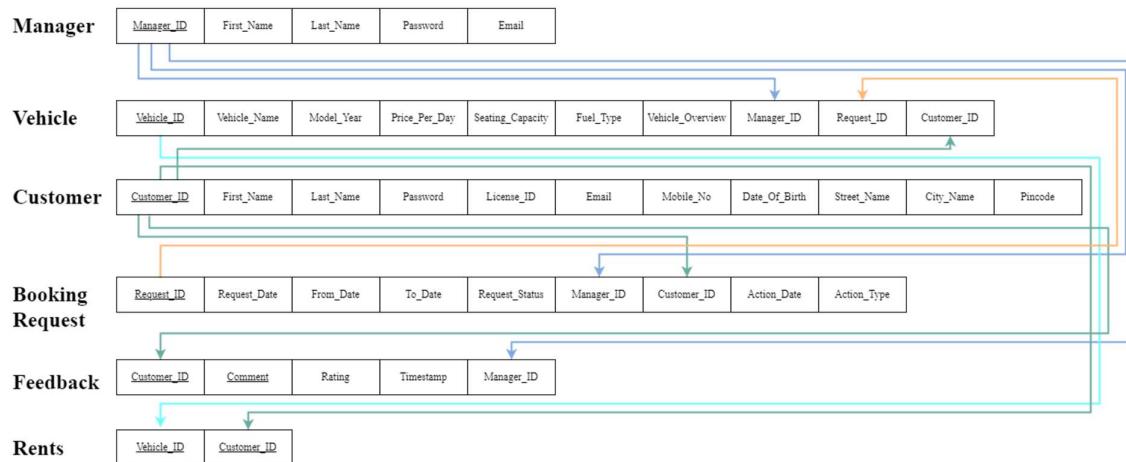
5. Version Control: GitHub

- For version control and collaborative development.

3. ER Diagram



4. Relational Schema



5. DDL Commands for creating Tables in OVBMS

Customer Table:

```
CREATE TABLE `customer` (
    `email` varchar(45) NOT NULL,
    `name` varchar(45) NOT NULL,
    `password` varchar(255) NOT NULL,
    `license_id` varchar(16) NOT NULL,
    `mobile_no` varchar(10) NOT NULL,
    `dob` date NOT NULL,
    PRIMARY KEY (`email`)
)
```

Booking Request Table:

```
CREATE TABLE `booking_requests` (
    `request_id` varchar(255) NOT NULL,
    `request_date` datetime NOT NULL,
    `from_date` date NOT NULL,
    `to_date` date NOT NULL,
    `request_status` varchar(15) NOT NULL DEFAULT 'Pending',
    `customer_id` varchar(45) NOT NULL,
    `vehicle_id` varchar(10) NOT NULL,
    `manager_id` varchar(45) DEFAULT NULL,
    `action_type` varchar(45) DEFAULT NULL,
    `action_date` datetime DEFAULT NULL,
    PRIMARY KEY (`request_id`),
    KEY `fk_manager_id_idx` (`manager_id`),
    KEY `fk_br_manager_id_idx` (`manager_id`),
    KEY `fk_customer_id_idx` (`customer_id`),
    KEY `fk_vehicle_id_idx` (`vehicle_id`),
    CONSTRAINT `fk_br_manager_id` FOREIGN KEY (`manager_id`) REFERENCES
    `manager` (`email`) ON UPDATE CASCADE,
    CONSTRAINT `fk_customer_id` FOREIGN KEY (`customer_id`) REFERENCES
    `customer` (`email`) ON UPDATE CASCADE,
    CONSTRAINT `fk_vehicle_id` FOREIGN KEY (`vehicle_id`) REFERENCES `vehicles`(
    `license_no`) ON UPDATE CASCADE
)
```

Vehicles Table:

```
CREATE TABLE `vehicles` (
  `license_no` varchar(10) NOT NULL,
  `vehicle_name` varchar(45) NOT NULL,
  `model_year` year NOT NULL,
  `price_per_day` decimal(10,0) NOT NULL,
  `seating_capacity` int NOT NULL,
  `fuel_type` varchar(10) NOT NULL,
  `vehicle_image` longtext,
  `vehicle_overview` varchar(225) DEFAULT NULL,
  `manager_id` varchar(45) NOT NULL,
  `availability` varchar(45) NOT NULL DEFAULT 'Available',
  PRIMARY KEY (`license_no`),
  KEY `fk_manager_id_idx` (`manager_id`),
  CONSTRAINT `fk_manager_id` FOREIGN KEY (`manager_id`) REFERENCES `manager`(`email`) ON UPDATE CASCADE
)
```

Manager Table:

```
CREATE TABLE `manager` (
  `email` varchar(45) NOT NULL,
  `name` varchar(45) NOT NULL,
  `password` varchar(255) NOT NULL,
  PRIMARY KEY (`email`)
)
```

6. CRUD operation Screenshots

6.1. CREATE: Adding Vehicle

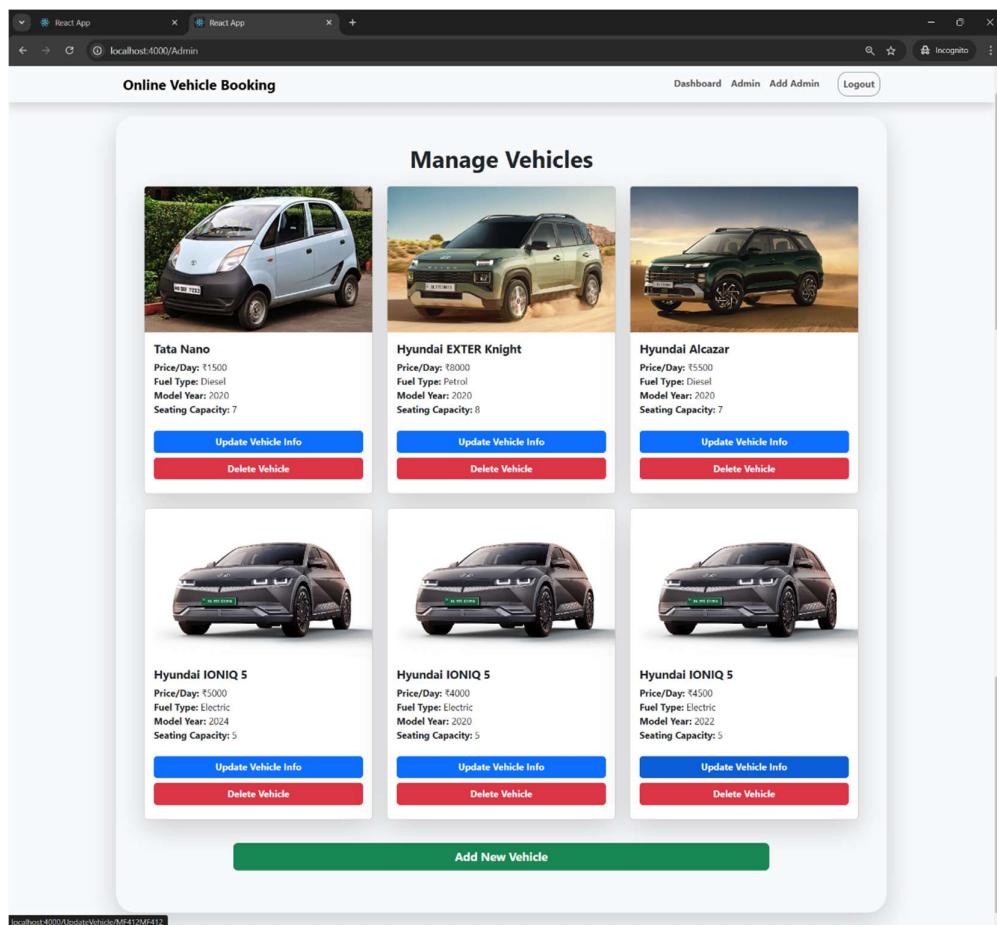
6.1.1. Adding Vehicle code

```

228 app.post('/AddVehicle', adminAuthenticateToken, (req, res)=>{
229   const q = `insert into vehicles
230   (License_No, Vehicle_Name, Model_Year, Price_Per_Day, Seating_Capacity, Fuel_Type, Vehicle_Image, Vehicle_Overview, manager_id)
231   values (?)`;
232   const values = [
233     req.body.License_No,
234     req.body.Vehicle_Name,
235     req.body.Model_Year,
236     req.body.Price_Per_Day,
237     req.body.Seating_Capacity,
238     req.body.Fuel_Type,
239     req.body.Vehicle_Image,
240     req.body.Vehicle_Overview,
241     req.user.id
242   ]
243 
244   AdminDB.query(q, [values], (err, data)=>{
245     if(err) return res.json(err)
246     return res.json(data)
247   })
248 })
  
```

You, 3 weeks ago • Users Creation/Varied Privileges & All Operatio...

6.1.2. Before Adding



DBMS Project Report | Online Vehicle Booking Management System

The screenshot shows the MySQL Workbench interface with a query results window titled 'vehicles'. The SQL query is:

```
SELECT * FROM ovbms.vehicles;
```

The results grid displays the following data:

license_no	vehicle_name	model_year	price_per_day	seating_capacity	fuel_type	vehicle_image	vehicle_overview	manager_id	availability
DL1TC01873	Hyundai Nexon	2020	6000	7	Petrol	https://www.hyundai.com/content/dam/hyundai/nexon/interior/exterior/interior/exterior.html	Good	admin@gmail.com	Unavailable
DL9TCC1369	Tata Nano	2020	1500	7	Diesel	https://upload.wikimedia.org/wikipedia/commons/2/2d/Tata_Nano_2015_01.jpg	Very Good Car	admin@gmail.com	Available
KA22EP7348	Hyundai EXTER Knight	2020	8000	8	Petrol	https://www.hyundai.com/content/dam/hyundai/exter-knight/interior/exterior/interior/exterior.html	Good Car	admin@gmail.com	Available
MF122WF34	Hyundai Alcazar	2020	5500	7	Diesel	https://www.hyundai.com/content/dam/hyundai/alcazar/interior/exterior/interior/exterior.html	Good Car	manager@gmail.com	Available
MF122WF40	Hyundai IONIQ 5	2024	5000	5	Electric	https://ionq5.hyundai.co.in/uploads/attachment/	Good Car	manager@gmail.com	Available
MF122WF412	Hyundai IONIQ 5	2022	4500	5	Electric	https://ionq5.hyundai.co.in/uploads/attachment/	Good Car	admin@gmail.com	Available

6.1.3. Adding the Vehicle

The screenshot shows a modal dialog box from the 'Online Vehicle Booking' application. The message says 'localhost:4000 says New Vehicle added successfully!' with an 'OK' button. The background shows the vehicle addition form with the following data:

MH47TC1242
Hyundai CRETA
2020
6000
7
Diesel
https://www.hyundai.com/content/dam/hyundai/n/creta/interior/exterior/interior/exterior.html
Good

A blue 'Add Vehicle' button is at the bottom of the form.

6.1.4. After Adding the Vehicle

The screenshot shows a vehicle detail page for a Hyundai CRETA. The vehicle image is a black SUV driving on a road with mountains in the background. The vehicle details listed are: Price/Day: ₹6000, Fuel Type: Diesel, Model Year: 2020, and Seating Capacity: 7. Below the details are two buttons: 'Update Vehicle Info' (blue) and 'Delete Vehicle' (red). At the bottom of the page is a green button labeled 'Add New Vehicle'.

The screenshot shows the MySQL Workbench interface with the 'vehicles' table selected. The table has the following columns: license_no, vehicle_name, model_year, price_per_day, seating_capacity, fuel_type, vehicle_image, vehicle_overview, manager_id, and availability. The data includes various vehicle models like Tata Nano, Hyundai EXTER Knight, and Hyundai CRETA, with details such as price per day (e.g., ₹6000), seating capacity (e.g., 7), and fuel type (e.g., Diesel or Electric).

license_no	vehicle_name	model_year	price_per_day	seating_capacity	fuel_type	vehicle_image	vehicle_overview	manager_id	availability
DL7TC01873	Hyundai NEXON	2020	6000	7	Petrol	https://www.hyundai.com/content/dam/hyundai/global/cars/nexon/interior/exterior/interior/exterior.html	Good	admin@gmail.com	Unavailable
DL7TCC1367	Tata Nano	2020	1500	7	Diesel	https://upload.wikimedia.org/wikipedia/commons/2/2d/Tata_Nano_2018_01.jpg	Very Good Car	admin@gmail.com	Available
DL7TCC1394	Hyundai EXTER Knight	2020	8000	8	Petrol	https://www.hyundai.com/content/dam/hyundai/global/cars/exter_knight/interior/exterior/interior/exterior.html	Good Car	admin@gmail.com	Available
MF12MF7946	Hyundai Tucson	2020	12000	7	Petrol	https://www.hyundai.com/content/dam/hyundai/global/cars/tucson/interior/exterior/interior/exterior.html	Good Car	manager@gmail.com	Available
MF12MF1004	Hyundai IONIQ 5	2021	5000	5	Electric	https://ionq5.hyundai.co.in/uploads/attachment...	Good Car	manager@gmail.com	Available
MF12MF412	Hyundai IONIQ 5	2022	4500	5	Electric	https://ionq5.hyundai.co.in/uploads/attachment...	Good Car	admin@gmail.com	Available
MH47TC1242	Hyundai CRETA	2020	6000	7	Diesel	https://www.hyundai.com/content/dam/hyundai/global/cars/creta/interior/exterior/interior/exterior.html	Good	admin@gmail.com	Available

6.2. READ: Vehicle Listing

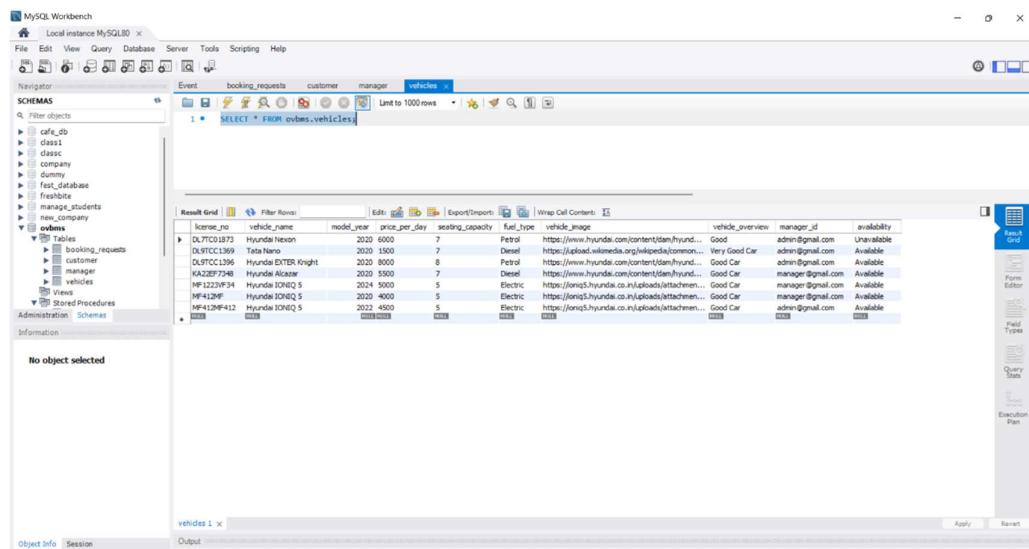
6.2.1. Vehicle Listing Code

```

122  app.get("/VehicleListings", (req, res) => {
123    const q = `SELECT * FROM ovbms.vehicles where availability in ('Available','Waiting')`;
124    clientDB.query(q, (err, data) => {
125      if (err) return res.json(err);
126      return res.json(data);
127    })
128  })

```

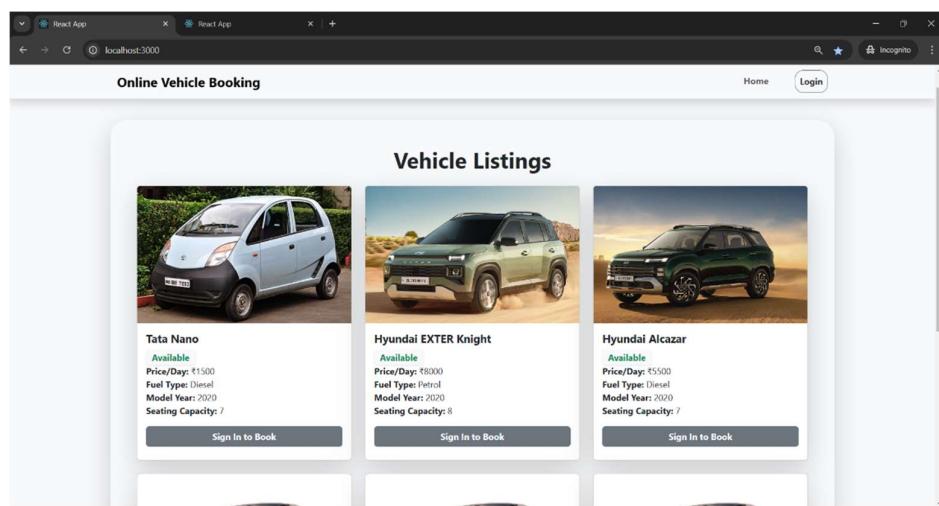
6.2.2. Vehicle Table



The screenshot shows the MySQL Workbench interface with the 'vehicles' table selected in the query editor. The results grid displays the following data:

license_no	vehicle_name	model_year	price_per_day	seating_capacity	fuel_type	vehicle_image	vehicle_overview	manager_id	availability
DL77CO1873	Hyundai Nios	2020	6000	7	Petrol	https://www.hyundai.com/content/dam/hyundai/global/cars/nios/interior/exterior/interior/exterior.html	Good	admin@gmail.com	Unavailable
DL9TCO1369	Tata Nano	2020	1500	7	Diesel	https://upload.wikimedia.org/wikipedia/commons/0/0d/Tata_Nano_2015_01.jpg	Very Good Car	admin@gmail.com	Available
DL9TCO1370	Hyundai EXTER Knight	2020	8000	8	Petrol	https://www.hyundai.com/content/dam/hyundai/global/cars/exter_knight/interior/exterior/interior/exterior.html	Good	admin@gmail.com	Available
AJX320FT7348	Hyundai Alcazar	2020	5500	7	Diesel	https://www.hyundai.com/content/dam/hyundai/global/cars/alcazar/interior/exterior/interior/exterior.html	Good Car	manager@gmail.com	Available
MP1223VF34	Hyundai IONIQ 5	2024	5000	5	Electric	https://ioniq5.hyundai.co.in/uploads/attachments/	Good Car	manager@gmail.com	Available
MP412MF412	Hyundai IONIQ 5	2020	4000	5	Electric	https://ioniq5.hyundai.co.in/uploads/attachments/	Good Car	manager@gmail.com	Available
MP412MF412	Hyundai IONIQ 5	2022	4500	5	Electric	https://ioniq5.hyundai.co.in/uploads/attachments/	Good Car	admin@gmail.com	Available

6.2.3. Vehicle Listing in UI



6.3. UPDATE: Updating Vehicle Information

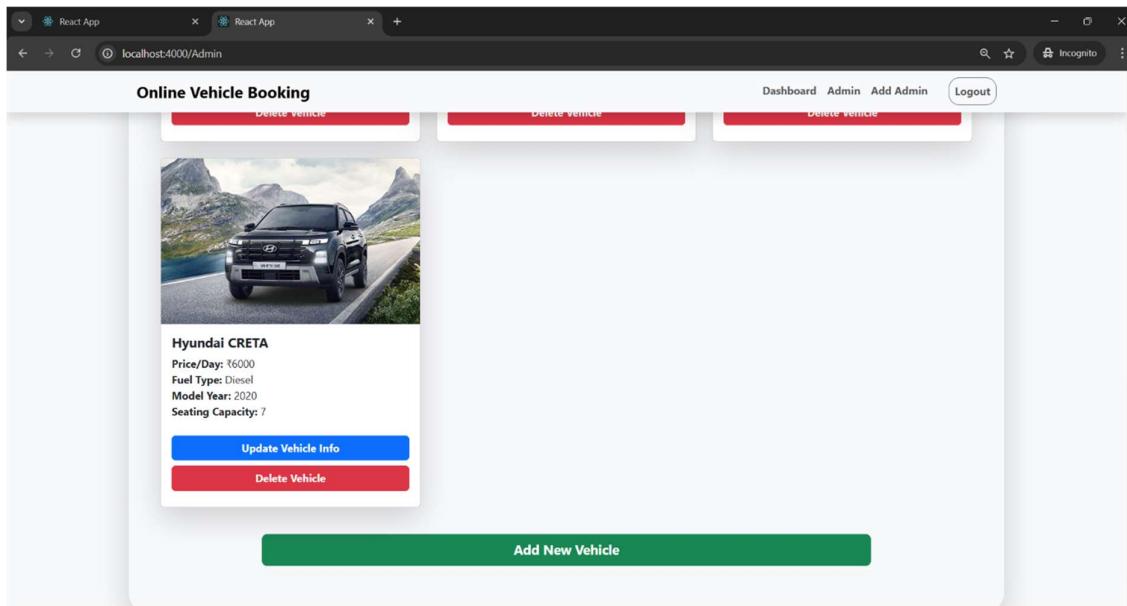
6.3.1. Vehicle Update Code

```

270   app.put("/UpdateVehicle/:id", adminAuthenticateToken, (req, res) => [
271     const vehicle_license_no = req.params.id
272     const q = `
273       UPDATE vehicles
274       SET
275         `License_No` = ?,
276         `Vehicle_Name` = ?,
277         `Model_Year` = ?,
278         `Price_Per_Day` = ?,
279         `Seating_Capacity` = ?,
280         `Fuel_Type` = ?,
281         `Vehicle_Image` = ?,
282         `Vehicle_Overview` = ?,
283         `manager_id` = ?
284       WHERE License_No = ?
285   ]
286
287   const values = [
288     req.body.License_No,
289     req.body.Vehicle_Name,
290     req.body.Model_Year,
291     req.body.Price_Per_Day,
292     req.body.Seating_Capacity,
293     req.body.Fuel_Type,
294     req.body.Vehicle_Image,
295     req.body.Vehicle_Overview,
296     req.user.id
297   ]
298
299   AdminDB.query(q, [...values, vehicle_license_no], (err, data) => {
300     if (err) return res.json(err)
301     return res.json("Vehicle Info has been Updated Successfully")
302   })
303 ]
304

```

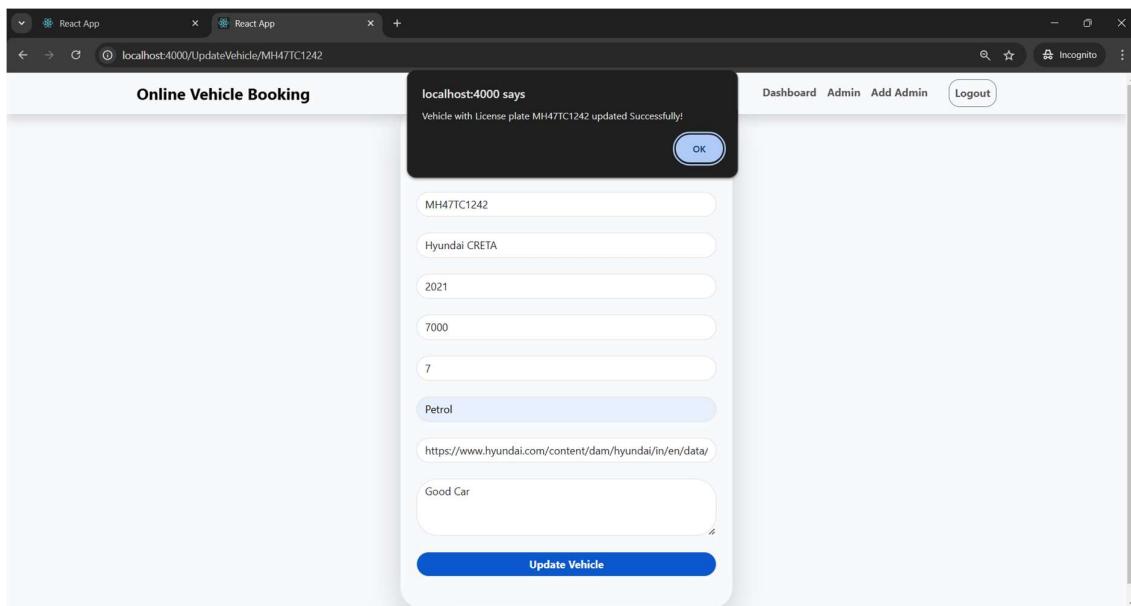
6.3.2. Before Update



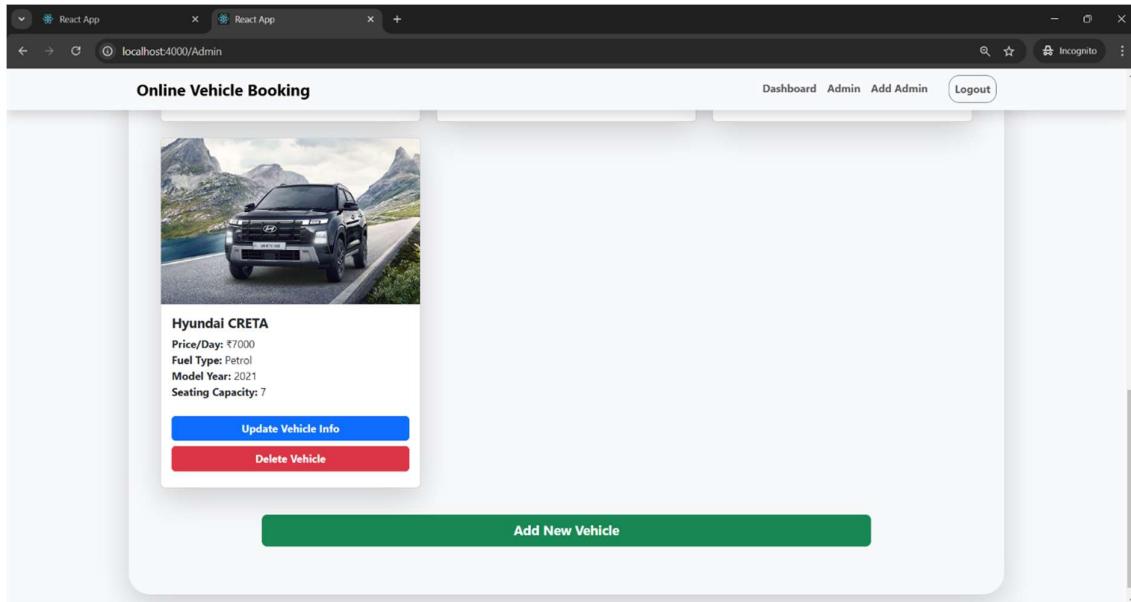
The screenshot shows the MySQL Workbench interface with the 'vehicles' table selected. The table contains the following data:

license_no	vehicle_name	model_year	price_per_day	seating_capacity	fuel_type	vehicle_image	vehicle_overview	manager_id	availability
DL7TC01873	Hyundai NEXON	2020	6000	5	Petrol	https://www.hyundai.com/content/dam/hyundai/motorcars/global/cars/nexon/interior/exterior/interior/exterior.html	Good	admin@gmail.com	Unavailable
DL9TCC13693	Tata Nano	2020	1500	7	Diesel	https://upload.wikimedia.org/wikipedia/commons/0/0c/Tata_Nano_2015_01.jpg	Very Good	admin@gmail.com	Available
DL9TCC13946	Hyundai EXTER Knight	2020	8000	8	Petrol	https://www.hyundai.com/content/dam/hyundai/motorcars/global/cars/exter_knight/interior/exterior/interior/exterior.html	Good Car	admin@gmail.com	Available
MF123MF7946	Hyundai Tucson	2020	10000	7	Petrol	https://www.hyundai.com/content/dam/hyundai/motorcars/global/cars/tucson/interior/exterior/interior/exterior.html	Good Car	manager@gmail.com	Available
MF123MF584	Hyundai IONIQ 5	2024	5000	5	Electric	https://ionq5.hyundai.co.in/uploads/attachment...	Good Car	manager@gmail.com	Available
MF123MF412	Hyundai IONIQ 5	2022	4500	5	Electric	https://ionq5.hyundai.co.in/uploads/attachment...	Good Car	admin@gmail.com	Available
MH47TC1242	Hyundai CRETA	2020	6000	7	Diesel	https://www.hyundai.com/content/dam/hyundai/motorcars/global/cars/creta/interior/exterior/interior/exterior.html	Good	admin@gmail.com	Available

6.3.3. Updating Vehicle Info



6.3.4. After Update



The screenshot shows the MySQL Workbench interface with a query editor window open. The query is:

```
SELECT * FROM ovbms.vehicles;
```

The results grid displays the following data:

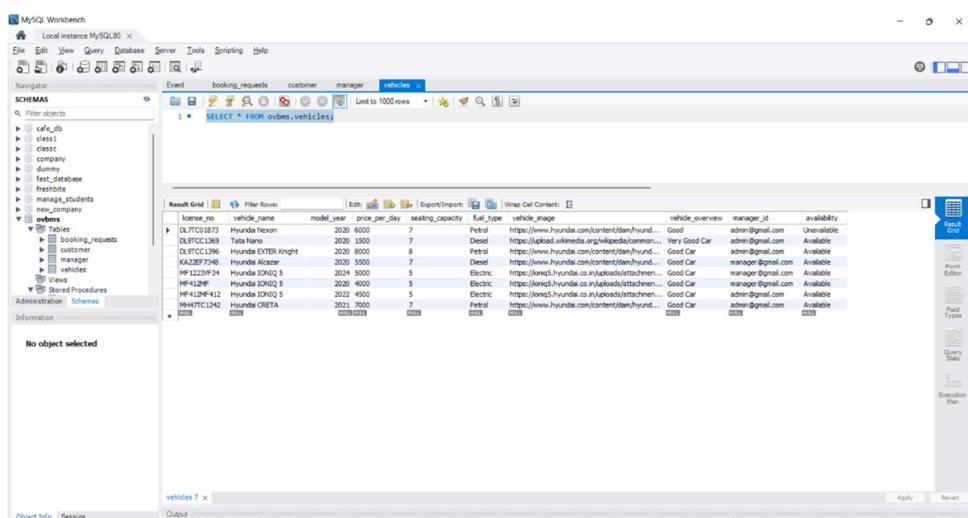
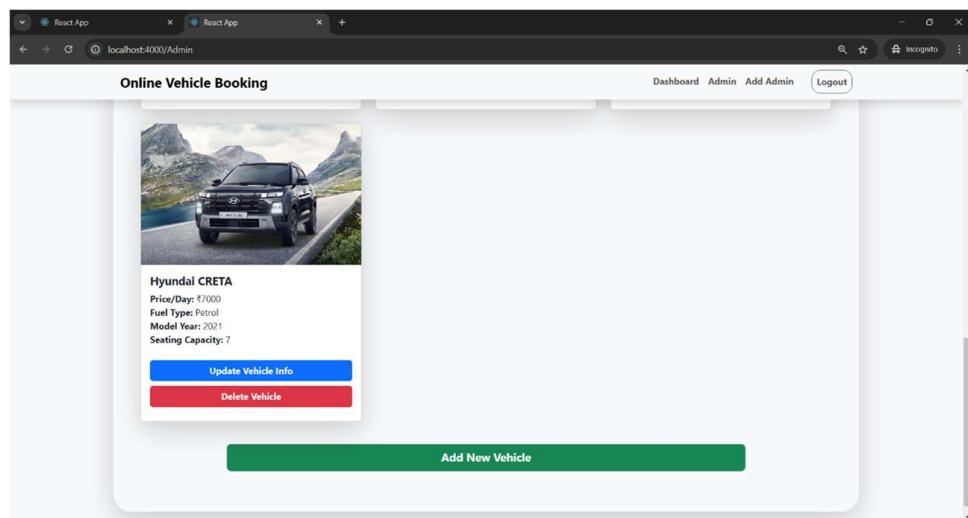
license_no	vehicle_name	model_year	price_per_day	seating_capacity	fuel_type	vehicle_image	vehicle_overview	manager_id	availability
DLT0C1873	Hyundai Hexon	2020	6000	7	Petrol	https://www.hyundai.com/content/dam/hyundai/global/cars/hexon/interior/exterior_hexon_01.jpg	Good	admin@gmail.com	Unreliable
DL9TC1369	Tata Nano	2020	3500	7	Diesel	https://upload.wikimedia.org/wikipedia/commons/2/2d/Tata_Nano_2015_01.jpg	Very Good Car	admin@gmail.com	Available
DL9TC1396	Hyundai EXTER Knight	2020	8000	8	Petrol	https://www.hyundai.com/content/dam/hyundai/global/cars/exter_knight/interior/exterior_exter_knight_01.jpg	Good Car	admin@gmail.com	Available
KA22EP7348	Hyundai Alcazar	2020	5500	7	Diesel	https://www.hyundai.com/content/dam/hyundai/global/cars/alcazar/interior/exterior_alcazar_01.jpg	Good Car	manager@gmail.com	Available
MF1223WF34	Hyundai IONIQ 5	2024	5000	5	Electric	https://ioniq.hyundai.co.in/uploads/attachment/IONIQ5/IONIQ5_01.jpg	Good Car	manager@gmail.com	Available
MF412MF	Hyundai IONIQ 5	2020	4000	5	Electric	https://ioniq.hyundai.co.in/uploads/attachment/IONIQ5/IONIQ5_01.jpg	Good Car	manager@gmail.com	Available
MF412MF412	Hyundai IONIQ 5	2022	4500	5	Electric	https://ioniq.hyundai.co.in/uploads/attachment/IONIQ5/IONIQ5_01.jpg	Good Car	admin@gmail.com	Available
MH47TC1242	Hyundai CRETA	2021	7000	7	Petrol	https://www.hyundai.com/content/dam/hyundai/global/cars/creta/interior/exterior_creta_01.jpg	Good Car	admin@gmail.com	Available

6.4. DELETE: Deleting the Vehicle

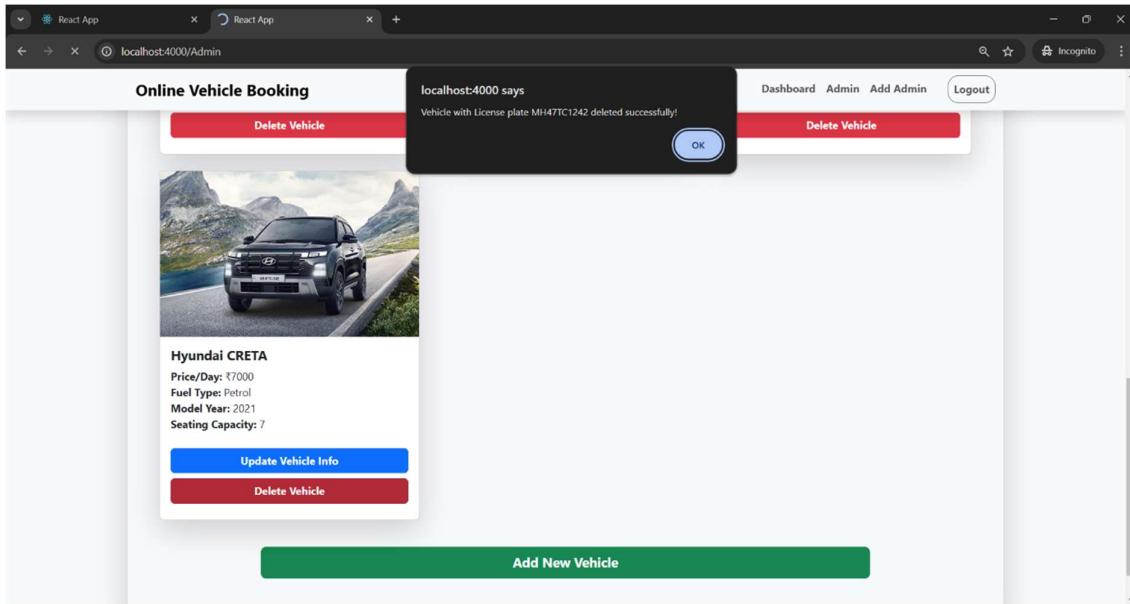
6.4.1. Vehicle Deletion Code

```
250 app.delete("/Admin/:id", adminAuthenticateToken, (req, res) => {
251   const License_No = req.params.id
252   const q = 'delete from vehicles where license_no = ?'
253
254   AdminDB.query(q, [License_No], (err, data) => {
255     if (err) return res.json(err)
256     return res.json("Vehicle Deleted")
257   })
258 })
```

6.4.2. Before Deletion



6.4.3. Deleting the Vehicle

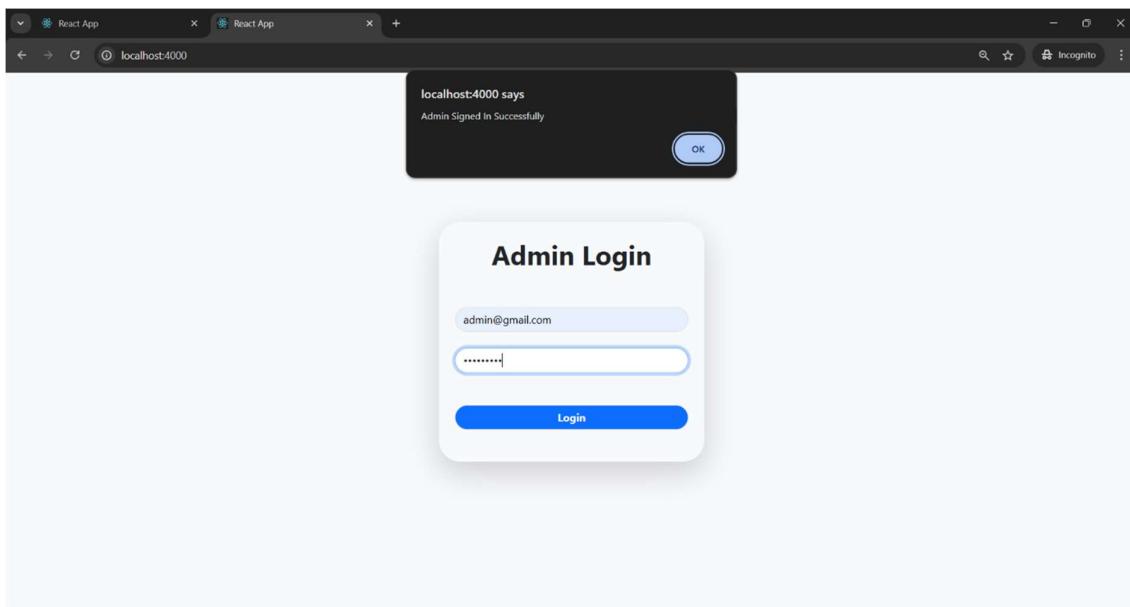


6.4.4. After Deletion

license_no	vehicle_name	model_year	price_per_day	seating_capacity	fuel_type	vehicle_image	vehicle_overview	manager_id	availability
DL5TC1249	Tata Nano	2020	1500	7	Diesel	https://www.tatamotors.com/tata-nano.html	Very Good Car	admin@gmail.com	Available
DL5TC1296	Tata Nano	2020	1500	7	Diesel	https://www.tatamotors.com/tata-nano.html	Very Good Car	admin@gmail.com	Available
KAZD2348	Hyundai IONIQ 5	2020	8000	5	Petrol	https://www.hyundai.com/content/us/hyundai/IONIQ-5.html	Good Car	manager@gmail.com	Available
MH422948	Hyundai IONIQ 5	2020	8000	5	Petrol	https://www.hyundai.com/content/us/hyundai/IONIQ-5.html	Good Car	manager@gmail.com	Available
MH422949	Hyundai IONIQ 5	2024	5000	5	Electric	https://ion5.hyundai.co.in/uploads/attachment...	Good Car	manager@gmail.com	Available
MH422949	Hyundai IONIQ 5	2020	4000	5	Electric	https://ion5.hyundai.co.in/uploads/attachment...	Good Car	admin@gmail.com	Available
MH422949	Hyundai IONIQ 5	2022	4500	5	Electric	https://ion5.hyundai.co.in/uploads/attachment...	Good Car	admin@gmail.com	Available
DL5TC1249	Tata Nano	2020	1500	7	Diesel	https://www.tatamotors.com/tata-nano.html	Very Good Car	admin@gmail.com	Available

7. List of functionalities/features of the application

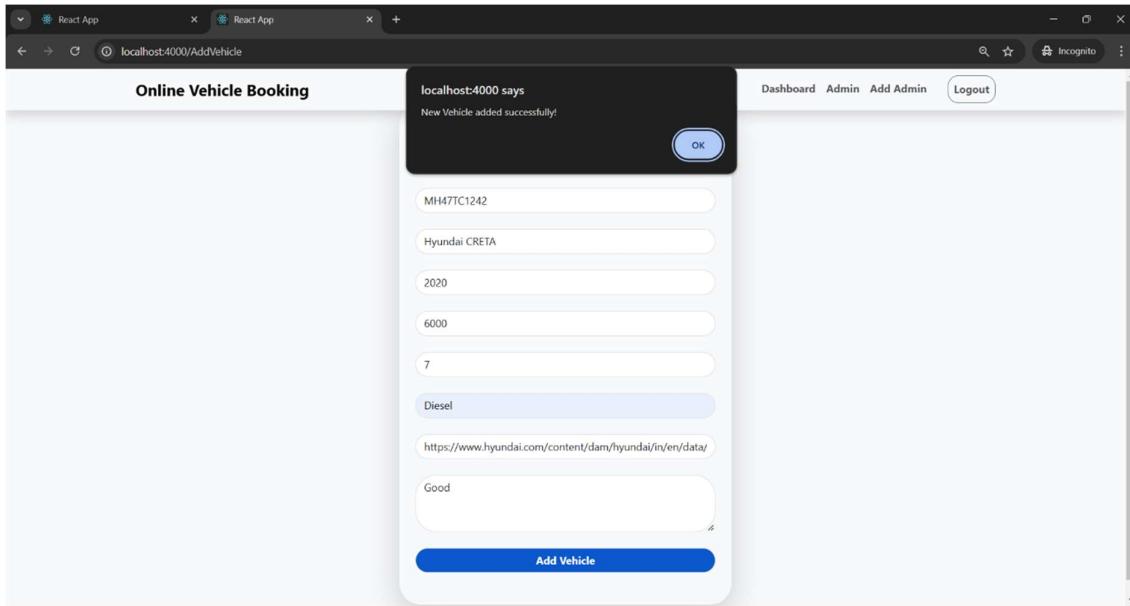
7.1. Admin Sign In



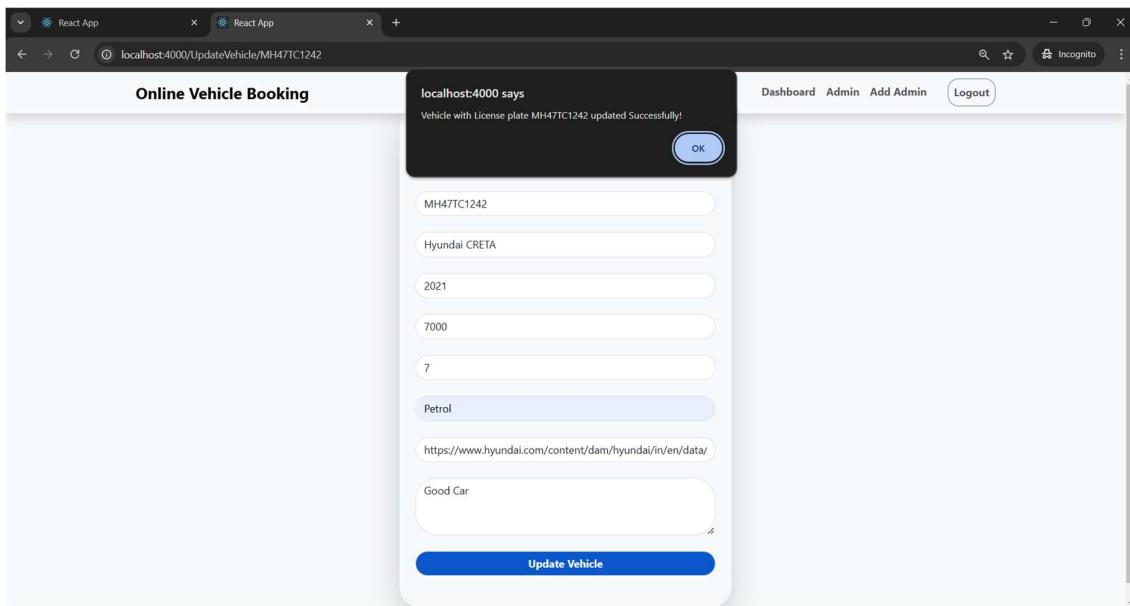
7.2. Vehicle Listing at Admin Side

A screenshot of the "Manage Vehicles" page from the "Online Vehicle Booking" application. The URL is "localhost:4000/Admin". The page header includes "Online Vehicle Booking", "Dashboard", "Admin", "Add Admin", and "Logout". The main content area is titled "Manage Vehicles" and displays three vehicle cards: 1. Tata Nano: Price/Day: ₹1500, Fuel Type: Diesel, Model Year: 2020, Seating Capacity: 2. Update Vehicle Info, Delete Vehicle. 2. Hyundai EXTER Knight: Price/Day: ₹8000, Fuel Type: Petrol, Model Year: 2020, Seating Capacity: 8. Update Vehicle Info, Delete Vehicle. 3. Hyundai Alcazar: Price/Day: ₹5500, Fuel Type: Diesel, Model Year: 2020, Seating Capacity: 7. Update Vehicle Info, Delete Vehicle.

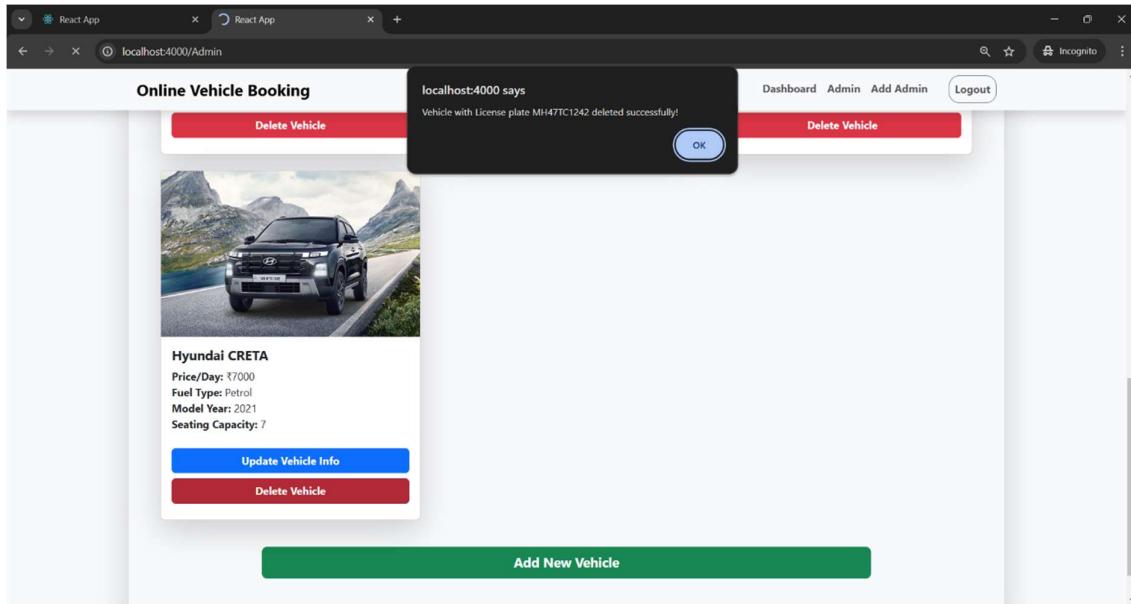
7.3. Adding Vehicle at Admin side



7.4. Updating Vehicle Info at Admin side



7.5. Deleting Vehicle at Admin side

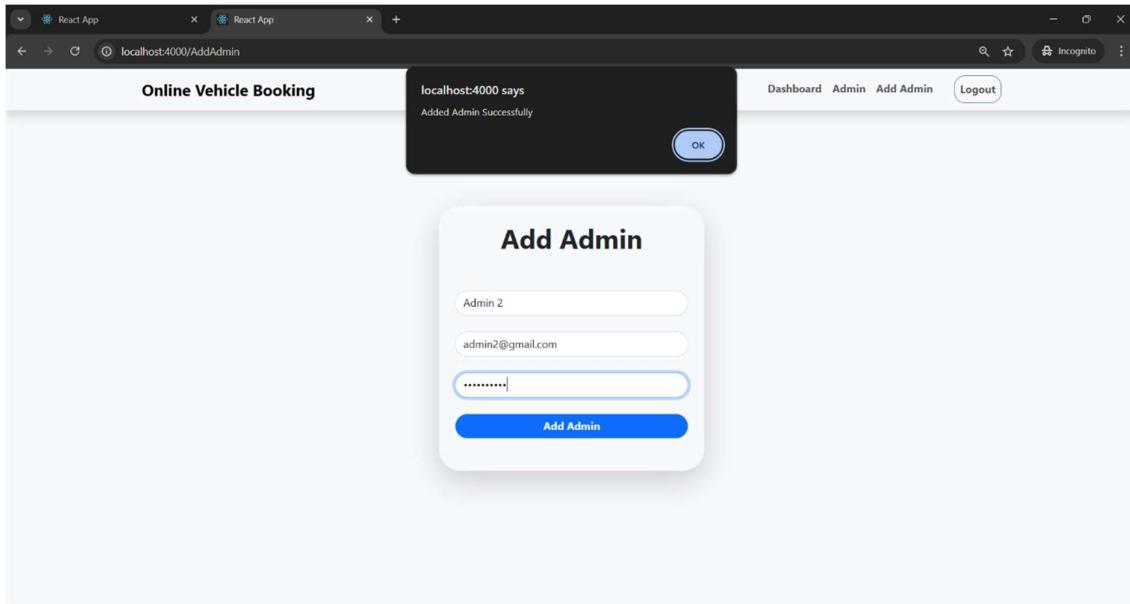


7.6. Admin Dashboard

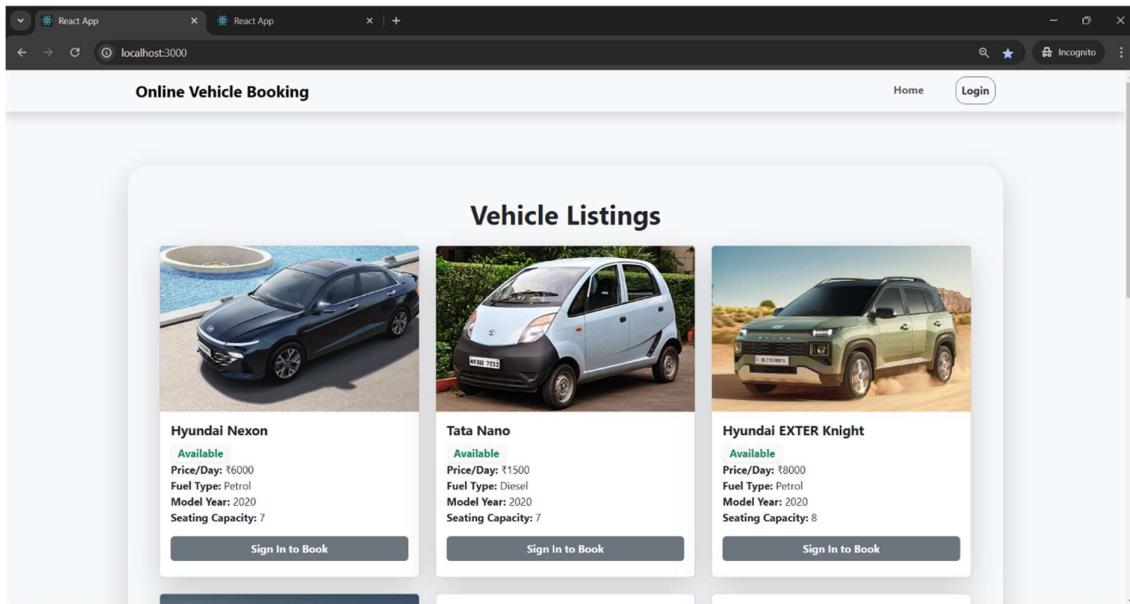
The screenshot shows a browser window for 'React App' at 'localhost:4000/BookingRequest'. The title bar reads 'Online Vehicle Booking' and 'Admin Dashboard'. The dashboard displays three summary boxes: 'Total Users' (3 Registered Users), 'Total Vehicles' (7 Listed Vehicles), and 'Total Bookings' (17 Booking Requests). Below these is a table titled 'Booking Requests' with columns: Name/License_No, Contact No./Email, Vehicle Name/License_ID, Request Date, From Date, To Date, Status, and Actions. The table lists eight booking requests, each with a status of either 'Rejected' or 'Approved'.

Name/License_No	Contact No./Email	Vehicle Name/License_ID	Request Date	From Date	To Date	Status	Actions
User(DL-1420110012345)	9876543210/user@gmail.com	Tata Nano/DL9TCC1369	11/27/2024	11/29/2024	11/30/2024	Rejected	
User3(DL-1420110012346)	9276543211/user3@gmail.com	Hyundai Nexon/DL7TC01873	11/27/2024	11/28/2024	11/29/2024	Approved	
User(DL-1420110012345)	9876543210/user@gmail.com	Tata Nano/DL9TCC1369	11/25/2024	11/26/2024	11/27/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai EXTER Knight/DL9TCC1396	11/20/2024	11/21/2024	11/22/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai Alcazar/KA22EF7348	11/19/2024	11/21/2024	11/22/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai IONIQ 5/MF412MF412	11/19/2024	11/19/2024	11/19/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai IONIQ 5/MF1223VF34	11/19/2024	11/19/2024	11/19/2024	Rejected	

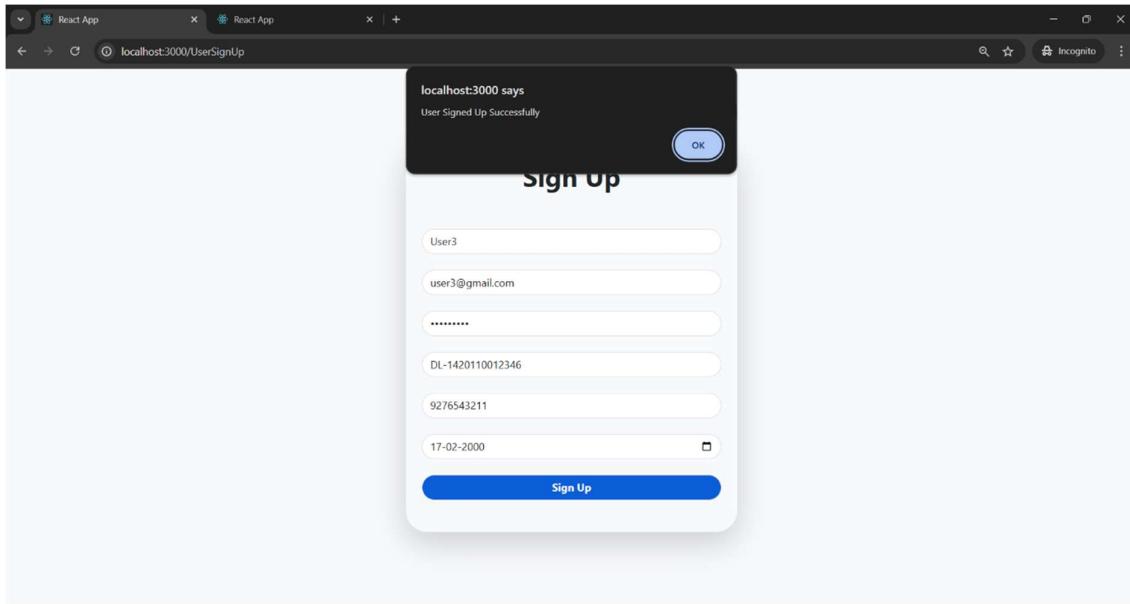
7.7. Adding different Admin at Admin side



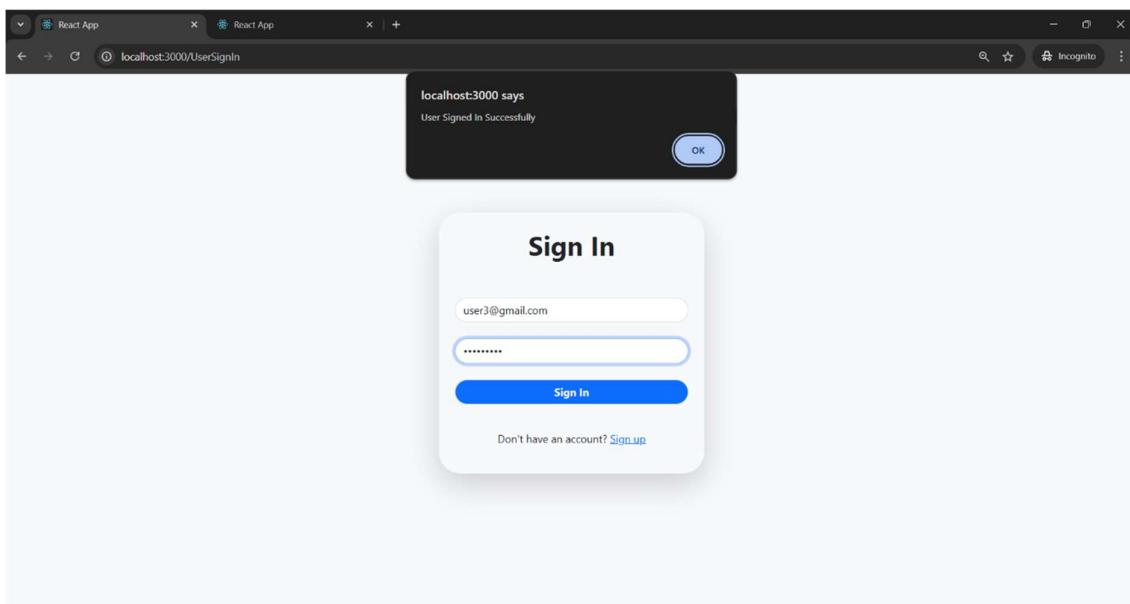
7.8. Vehicle Listing at User side before Sign In



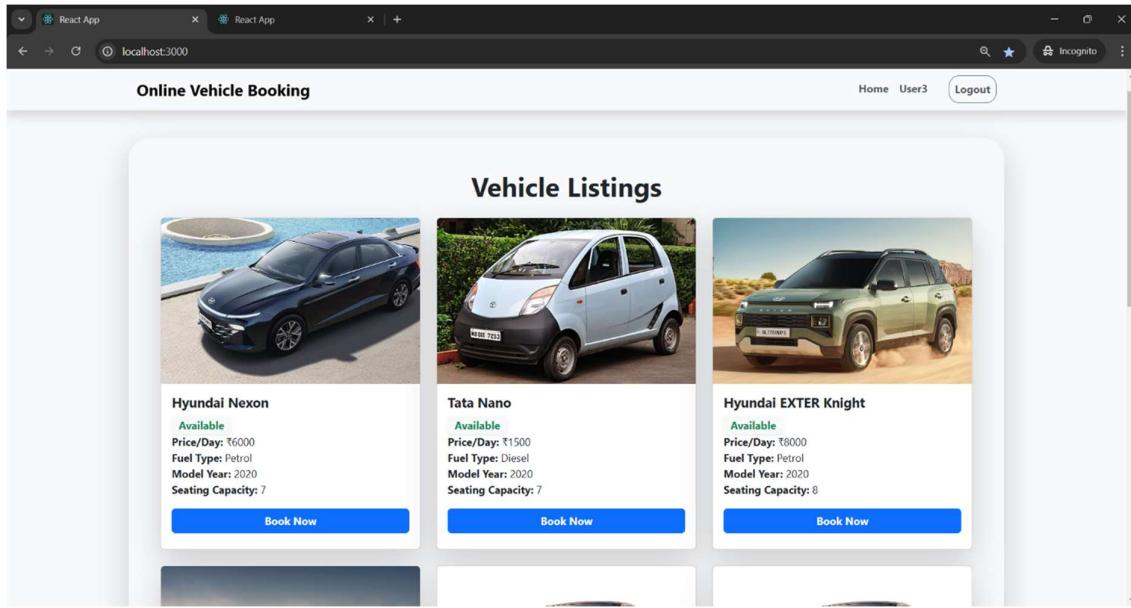
7.9. User Sign Up



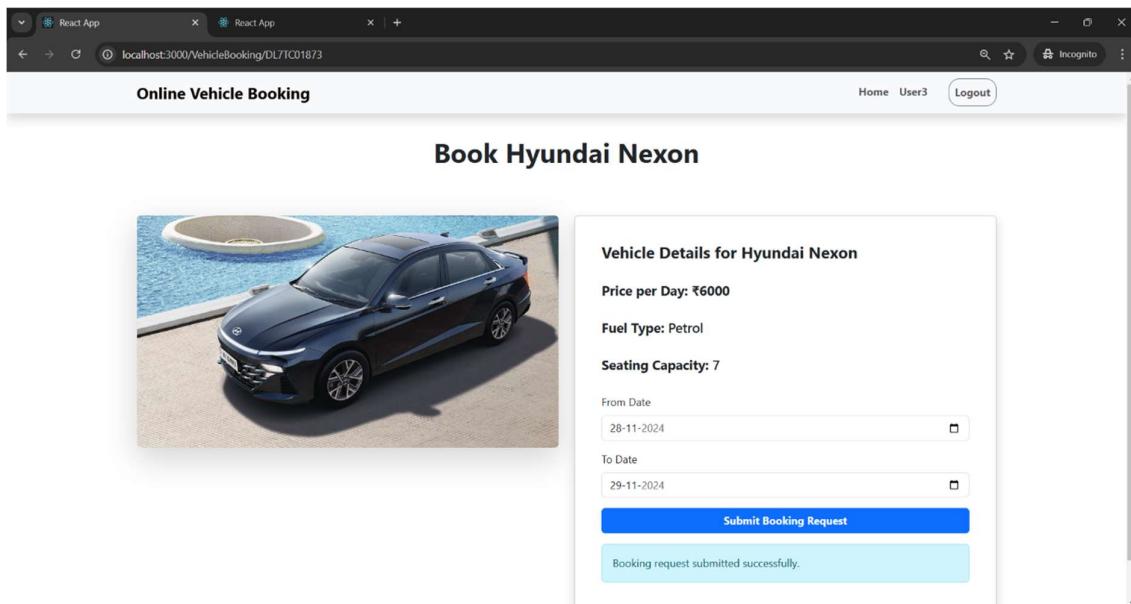
7.10. User Sign In



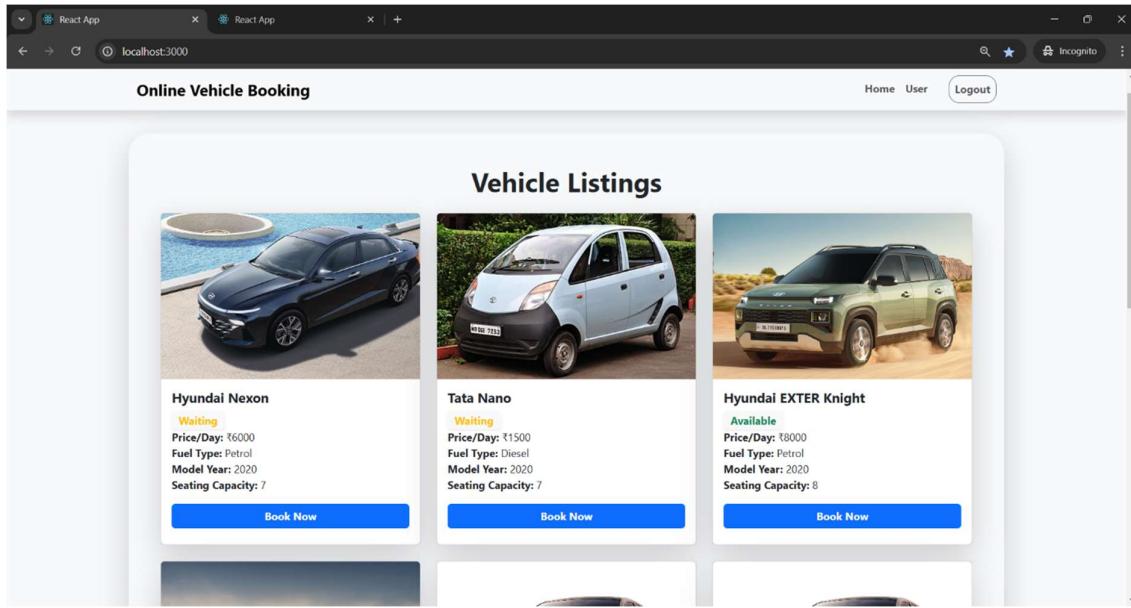
7.11. Vehicle Listing at User side after Sign In



7.12. Vehicle Booking



7.13. Vehicle Listing after Booking



7.14. Vehicle Booking Requests in Admin Dashboard

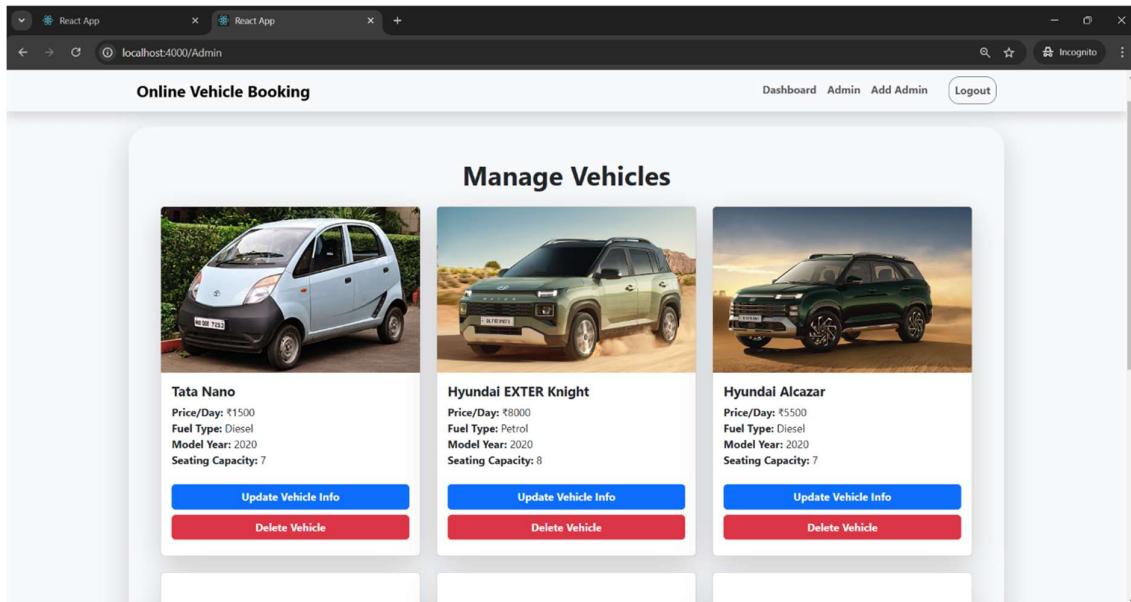
The screenshot shows a web browser window titled "React App" at "localhost:4000/BookingRequest". The main content is titled "Online Vehicle Booking" and "Booking Requests". A table lists the booking requests:

Name(License_No)	Contact No./Email	Vehicle Name/License_ID	Request Date	From Date	To Date	Status	Actions
User(DL-1420110012345)	9876543210/user@gmail.com	Tata Nano/DL9TCC1369	11/27/2024	11/29/2024	11/30/2024	Pending	<button>Approve</button> <button>Reject</button>
User3(DL-1420110012346)	9276543211/user3@gmail.com	Hyundai Nexon/DL7TC01873	11/27/2024	11/28/2024	11/29/2024	Pending	<button>Approve</button> <button>Reject</button>
User(DL-1420110012345)	9876543210/user@gmail.com	Tata Nano/DL9TCC1369	11/25/2024	11/26/2024	11/27/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai EXTER Knight/DL9TCC1396	11/20/2024	11/21/2024	11/22/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai Alcazar/KA22EF7348	11/19/2024	11/21/2024	11/22/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai IONIQ 5/MF412MF412	11/19/2024	11/19/2024	11/19/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai IONIQ 5/MF1223VF34	11/19/2024	11/19/2024	11/19/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Tata Nano/DL9TCC1369	11/19/2024	11/20/2024	11/21/2024	Completed	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai IONIQ 5/MF412MF412	11/19/2024	11/19/2024	11/19/2024	Completed	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai Alcazar/KA22EF7348	11/19/2024	11/20/2024	11/21/2024	Rejected	
User2(DL-2420110012345)	9276543210/user2@gmail.com	Tata Nano/DL9TCC1369	11/19/2024	11/19/2024	11/20/2024	Rejected	

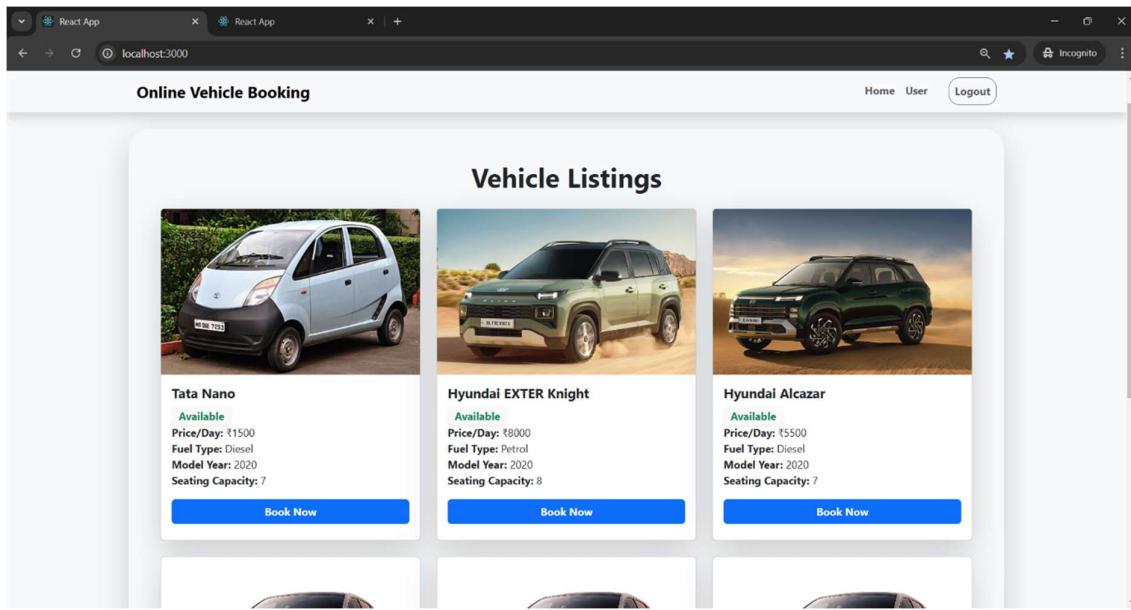
7.15. Vehicle Booking Requests in Admin Dashboard after handling the Booking Request

Booking Requests							
Name/License_No	Contact No./Email	Vehicle Name/License_ID	Request Date	From Date	To Date	Status	Actions
User(DL-1420110012345)	9876543210/user@gmail.com	Tata Nano/DL9TCC1369	11/27/2024	11/29/2024	11/30/2024	Rejected	
User3(DL-1420110012346)	9276543211/user3@gmail.com	Hyundai Nexon/DL7TC01673	11/27/2024	11/28/2024	11/29/2024	Approved	
User(DL-1420110012345)	9876543210/user@gmail.com	Tata Nano/DL9TCC1369	11/25/2024	11/26/2024	11/27/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai EXTER Knight/DL9TCC1396	11/20/2024	11/21/2024	11/22/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai Alcazar/KA22EF7348	11/19/2024	11/21/2024	11/22/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai IONIQ 5/MF412MF412	11/19/2024	11/19/2024	11/19/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai IONIQ 5/MF1223VF34	11/19/2024	11/19/2024	11/19/2024	Rejected	
User(DL-1420110012345)	9876543210/user@gmail.com	Tata Nano/DL9TCC1369	11/19/2024	11/20/2024	11/21/2024	Completed	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai IONIQ 5/MF412MF412	11/19/2024	11/19/2024	11/19/2024	Completed	
User(DL-1420110012345)	9876543210/user@gmail.com	Hyundai Alcazar/KA22EF7348	11/19/2024	11/20/2024	11/21/2024	Rejected	
User2(DL-2420110012345)	9276543210/user2@gmail.com	Tata Nano/DL9TCC1369	11/19/2024	11/19/2024	11/20/2024	Rejected	
User2(DL-2420110012345)	9276543210/user2@gmail.com	Hyundai IONIQ 5/MF1223VF34	11/19/2024	11/19/2024	11/20/2024	Completed	
User2(DL-2420110012345)	9276543210/user2@gmail.com	Hyundai IONIQ 5/MF412MF412	11/19/2024	11/19/2024	11/19/2024	Rejected	

7.16. Vehicle Listing at Admin side after handling the Booking Request



7.17. Vehicle Listing at User side after handling the Booking Request



8. Triggers, Procedures/Functions, Nested query, Join, Aggregate queries

8.1. Procedure 1

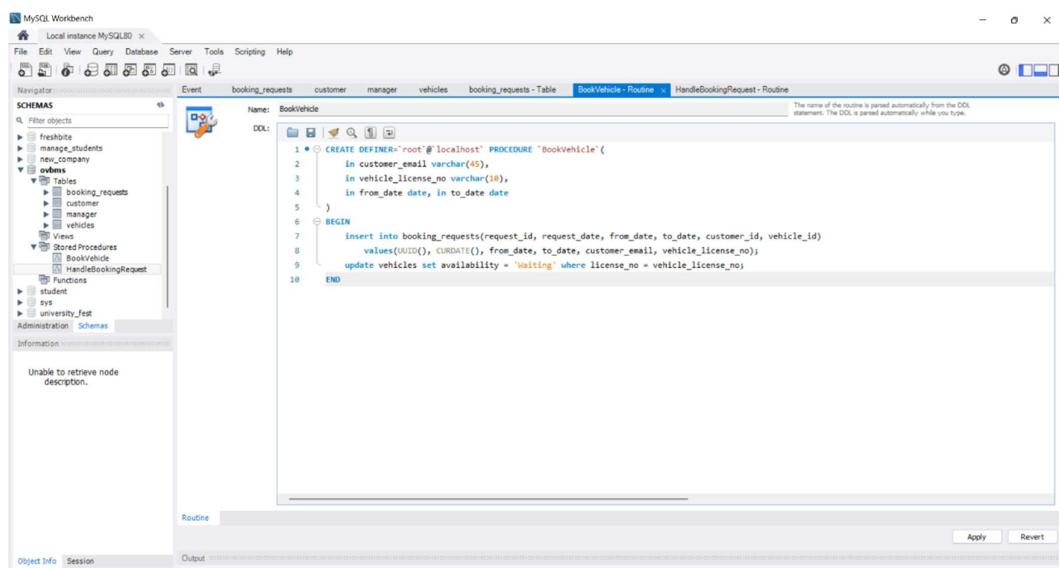
Procedure for Vehicle Booking:

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `HandleBookingRequest`(
    in requestId varchar(255),
    in action varchar(15),
    in managerId varchar(45))
BEGIN
    if action = 'Approve' then
        update booking_requests set
            request_status = 'Approved',
            action_type = 'Approve',
            action_date = curdate(),
            manager_id = managerId
        where request_id = requestId;

    elseif action = 'Reject' then
        update booking_requests set
            request_status = 'Rejected',
            action_type = 'Reject',
            action_date = curdate(),
            manager_id = managerId where request_id = requestId;
    END IF;
END ;

```



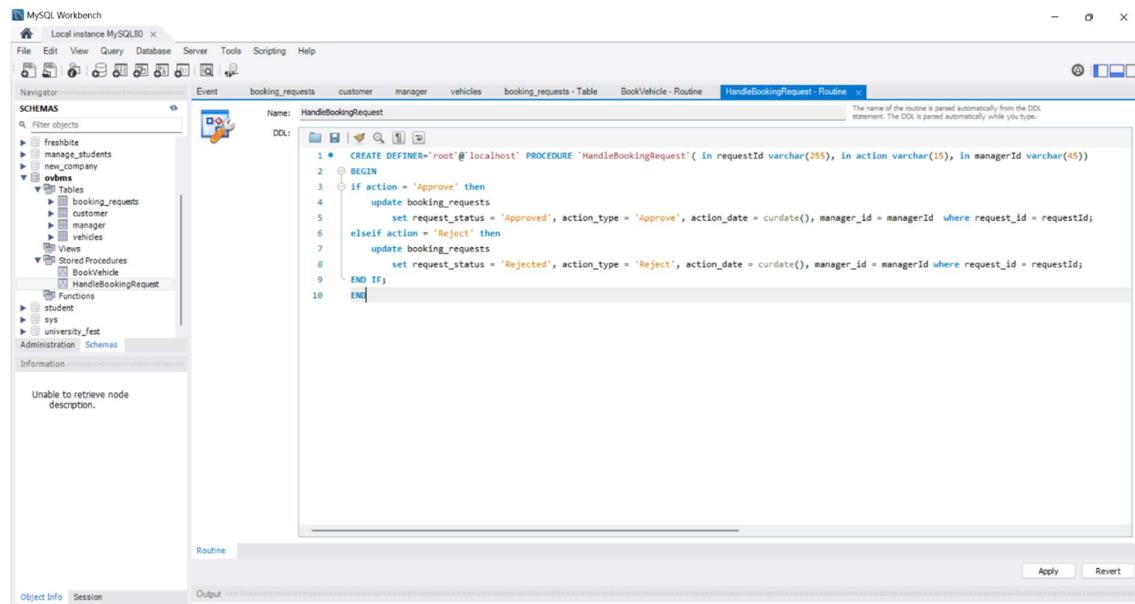
8.2. Procedure 2

Procedure for handling the Booking Requests:

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `HandleBookingRequest`(
    in requestId varchar(255),
    in action varchar(15),
    in managerId varchar(45))
BEGIN
    if action = 'Approve' then
        update booking_requests set
            request_status = 'Approved',
            action_type = 'Approve',
            action_date = curdate(),
            manager_id = managerId
        where request_id = requestId;
    elseif action = 'Reject' then
        update booking_requests set
            request_status = 'Rejected',
            action_type = 'Reject',
            action_date = curdate(),
            manager_id = managerId
        where request_id = requestId;
    END IF;
END ;;

```

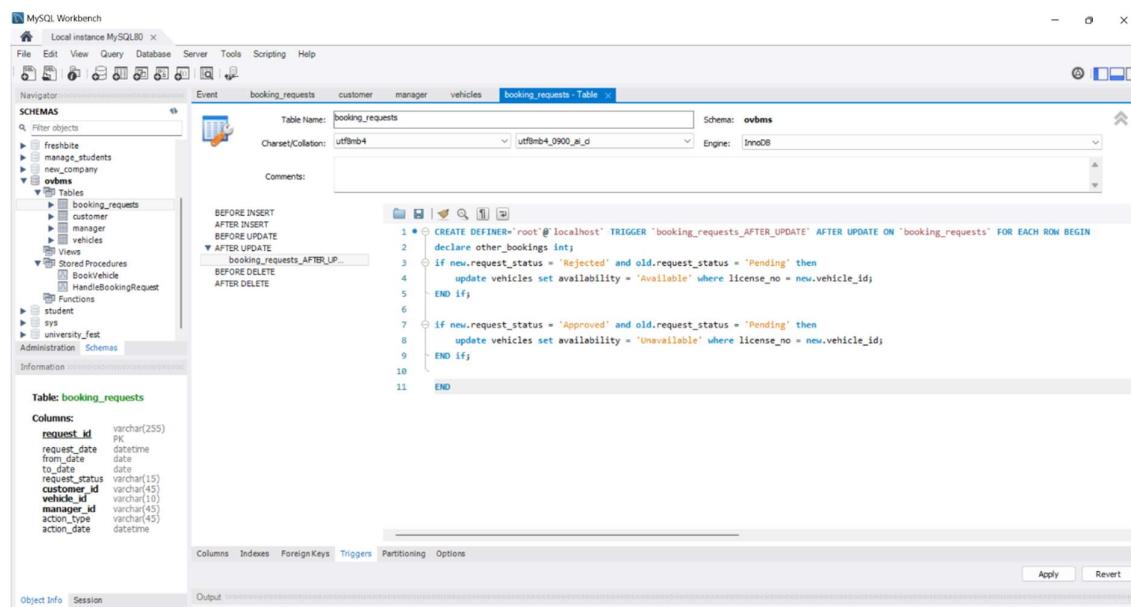


8.3. Trigger

Trigger for handling the data in “Vehicles” tables when the action is taken on Booking Request (after update on “Booking_Requests” table):

```
CREATE DEFINER=`root`@`localhost` TRIGGER `booking_requests_AFTER_UPDATE`
AFTER UPDATE ON `booking_requests`
FOR EACH ROW
BEGIN
    declare other_bookings int;
    if new.request_status = 'Rejected'
    and old.request_status = 'Pending' then
        update vehicles set availability = 'Available'
        where license_no = new.vehicle_id;
    END if;

    if new.request_status = 'Approved'
    and old.request_status = 'Pending' then
        update vehicles set availability = 'Unavailable'
        where license_no = new.vehicle_id;
    END if;
END
```



8.4. Event

Event for handling the data in “Booking_Requests” table and “Vehicles” table whenever the booking span of the vehicle is completed (checks everyday):

```
delimiter $$

CREATE DEFINER=`root`@`localhost` EVENT `CheckCompletedBookings`
ON SCHEDULE EVERY 1 DAY
STARTS CURRENT_TIMESTAMP
DO
BEGIN
    UPDATE vehicles SET availability = 'Available'
    WHERE license_no IN (
        SELECT vehicle_id
        FROM booking_requests
        WHERE request_status = 'Approved'
        AND to_date < CURDATE());
    
    UPDATE booking_requests
    SET request_status = 'Completed'
    WHERE request_status = 'Approved'
    AND to_date < CURDATE();
END $$

delimiter ;
```

8.5. Nested Query & Aggregate Query

Nested Query & Aggregate Query in the Backend used for Admin Dashboard

```
SELECT
    (SELECT COUNT(*) FROM customer) as total_users,
    (SELECT COUNT(*) FROM vehicles) as total_vehicles,
    (SELECT COUNT(*) FROM booking_requests) as total_bookings;
```

8.6. Join Query

Join Query for Listing the Booking Requests in Admin Dashboard with Customer details and Vehicle details

```
select
    br.request_id, br.request_date, br.from_date, br.to_date,
    br.request_status, br.vehicle_id,
    c.name, c.email, c.license_id, c.mobile_no,
    v.vehicle_name, v.price_per_day, v.vehicle_image, v.manager_id
from booking_requests br
join customer c on br.customer_id = c.email
join vehicles v on br.vehicle_id = v.license_no
order by br.request_date desc;
```

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows the project structure for 'ONLINE_VEHICLE_BOOKING_MANAGEMENT_SYSTEM' (abbreviated as OVBMS). It includes 'client', 'src' (containing 'index.js', 'logo.svg', 'reportWebVitals.js', 'setupTests.js'), 'Database' (containing 'Event.sql' and 'ovbms.sql'), and 'server' (containing 'node_modules', 'package-lock.json', and 'package.json').
- Code Editor:** The 'server.js' file is open. The code defines an 'app.get' route for '/BookingRequest' that performs a database query. The query selects data from 'booking_requests' (br), 'customer' (c), and 'vehicles' (v) tables, joining them on specific fields. The results are ordered by 'request_date' in descending order.
- Bottom Bar:** Includes icons for main, Launchpad, AI Code Chat, BLACKBOX Chat, Add Logs, Improve Code, Share Code Link, Generate Commit Message, Search Error, AI Code Chat, Prettier, and a GitHub icon.

9. Code snippets for invoking the Procedures/Functions/Trigger

9.1. Invoking “BookVehicle” Procedure:

9.2. Invoking “HandleBookingRequest” Procedure:

Note: Trigger is also invoked at this part only to handle the data in other tables

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure. The current file, `server.js`, is highlighted in blue.
- Code Editor (Center):** Displays the `server.js` file content. The code handles booking requests, interacting with an AdminDB and a MySQL database named `ovbms`. It includes logic for getting booking requests, posting booking requests, and listening on port 5000.
- Bottom Bar:** Includes icons for main, Launchpad, outline, timeline, and various status indicators like battery and signal strength.

```
File Edit Selection View Go Run ... < > Online_Vehicle_Booking_Management_System Explorer JS server.js

ONLINE... D E P O M ONLINE... JS server.js
OVBMS > server > JS server.js > app.get("/BookingRequest/:requestId") callback > AdminDB.query() callback
387 app.get("/BookingRequest", adminAuthenticateToken, (req, res) => {
    AdminDB.query(q, (err, data) => {
        if (err) return res.json(err)
        return res.json(data)
    })
})

app.post("/BookingRequest/:requestId", adminAuthenticateToken, (req, res) => {
    const { requestId } = req.params
    const { action } = req.body
    const managerId = req.user.id

    const q = 'CALL HandleBookingRequest(?, ?, ?)'
    AdminDB.query(q, [requestId, action, managerId], (err, data) => {
        if (err) return res.status(500).json(err)
        return res.status(200).json(data)
    })
})

app.listen(5000, ()=>{
    console.log("Connected to Backend")
})
336 })
```

10. SQL queries (Create, Insert, Triggers, Procedures/ Functions, Nested query, Join, Aggregate queries) used in the project in the form of .sql file

```

CREATE DATABASE IF NOT EXISTS `ovbms` /*!40100 DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci */ /*!80016 DEFAULT ENCRYPTION='N' */;
USE `ovbms`;
-- MySQL dump 10.13 Distrib 8.0.36, for Win64 (x86_64)
--
-- Host: localhost    Database: ovbms
-- -----
-- Server version 8.0.37

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `booking_requests`
--

DROP TABLE IF EXISTS `booking_requests`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `booking_requests` (
  `request_id` varchar(255) NOT NULL,
  `request_date` datetime NOT NULL,
  `from_date` date NOT NULL,
  `to_date` date NOT NULL,
  `request_status` varchar(15) NOT NULL DEFAULT 'Pending',
  `customer_id` varchar(45) NOT NULL,
  `vehicle_id` varchar(10) NOT NULL,
  `manager_id` varchar(45) DEFAULT NULL,
  `action_type` varchar(45) DEFAULT NULL,
  `action_date` datetime DEFAULT NULL,
  PRIMARY KEY (`request_id`),
  KEY `fk_manager_id_idx` (`manager_id`),
  KEY `fk_br_manager_id_idx` (`manager_id`),
  KEY `fk_customer_id_idx` (`customer_id`),

```

```

    KEY `fk_vehicle_id_idx` (`vehicle_id`),
    CONSTRAINT `fk_br_manager_id` FOREIGN KEY (`manager_id`) REFERENCES
`manager` (`email`) ON UPDATE CASCADE,
    CONSTRAINT `fk_customer_id` FOREIGN KEY (`customer_id`) REFERENCES
`customer` (`email`) ON UPDATE CASCADE,
    CONSTRAINT `fk_vehicle_id` FOREIGN KEY (`vehicle_id`) REFERENCES `vehicles`(
`license_no`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `booking_requests`
-- 

LOCK TABLES `booking_requests` WRITE;
/*!40000 ALTER TABLE `booking_requests` DISABLE KEYS */;
INSERT INTO `booking_requests` VALUES ('03a7e591-a65b-11ef-a411-
b44506948d64','2024-11-19 00:00:00','2024-11-20','2024-11-
21','Rejected','user@gmail.com','KA22EF7348','admin@gmail.com','Reject','2024-
11-19 00:00:00'),('12013110-acbc-11ef-b89b-b44506948d64','2024-11-27
00:00:00','2024-11-28','2024-11-
29','Approved','user3@gmail.com','DL7TC01873','admin@gmail.com','Approve','202
4-11-27 00:00:00'),('127b808c-a63c-11ef-a411-b44506948d64','2024-11-19
00:00:00','2024-11-19','2024-11-
19','Completed','user@gmail.com','MF412MF412','admin@gmail.com','Approve','202
4-11-19 00:00:00'),('27b53886-a63c-11ef-a411-b44506948d64','2024-11-19
00:00:00','2024-11-19','2024-11-
19','Rejected','user2@gmail.com','MF412MF412','admin@gmail.com','Reject','2024-
11-19 00:00:00'),('377f7cc6-acbc-11ef-b89b-b44506948d64','2024-11-27
00:00:00','2024-11-29','2024-11-
30','Rejected','user@gmail.com','DL9TCC1369','admin@gmail.com','Reject','2024-
11-27 00:00:00'),('67acae30-a6f4-11ef-a411-b44506948d64','2024-11-20
00:00:00','2024-11-21','2024-11-
22','Rejected','user@gmail.com','DL9TCC1396','admin@gmail.com','Reject','2024-
11-25 00:00:00'),('6e544a2c-a65e-11ef-a411-b44506948d64','2024-11-19
00:00:00','2024-11-20','2024-11-
21','Completed','user@gmail.com','DL9TCC1369','admin@gmail.com','Approve','202
4-11-19 00:00:00'),('80ac4895-a63a-11ef-a411-b44506948d64','2024-11-19
00:00:00','2024-11-19','2024-11-
19','Rejected','user@gmail.com','MF1223VF34','admin@gmail.com','Reject','2024-
11-19 00:00:00'),('85e8f85d-a643-11ef-a411-b44506948d64','2024-11-19
00:00:00','2024-11-19','2024-11-
20','Completed','user2@gmail.com','MF1223VF34','admin@gmail.com','Approve','20
24-11-19 00:00:00'),('8a5055b5-a63a-11ef-a411-b44506948d64','2024-11-19
00:00:00','2024-11-19','2024-11-
19','Rejected','user@gmail.com','MF412MF412','admin@gmail.com','Reject','2024-
11-19 00:00:00'),('add803bf-ab30-11ef-a411-b44506948d64','2024-11-25
00:00:00','2024-11-26','2024-11-

```

```

27', 'Rejected', 'user@gmail.com', 'DL9TCC1369', 'admin@gmail.com', 'Reject', '2024-
11-25 00:00:00'), ('ba0871d1-a64b-11ef-a411-b44506948d64', '2024-11-19
00:00:00', '2024-11-19', '2024-11-
20', 'Rejected', 'user2@gmail.com', 'DL9TCC1369', 'admin@gmail.com', 'Reject', '2024-
11-19 00:00:00'), ('c9da1da0-a534-11ef-a8a4-b44506948d64', '2024-11-18
00:00:00', '2024-11-18', '2024-11-
19', 'Completed', 'user@gmail.com', 'MF412MF', 'admin@gmail.com', 'Approve', '2024-
11-18 00:00:00'), ('e6568d9d-a309-11ef-a8a4-b44506948d64', '2024-11-15
00:00:00', '2024-11-17', '2024-11-
18', 'Rejected', 'user@gmail.com', 'DL9TCC1369', 'admin@gmail.com', 'Reject', '2024-
11-15 00:00:00'), ('e728937a-a669-11ef-a411-b44506948d64', '2024-11-19
00:00:00', '2024-11-21', '2024-11-
22', 'Rejected', 'user@gmail.com', 'KA22EF7348', 'admin@gmail.com', 'Reject', '2024-
11-19 00:00:00'), ('e7f8db69-a534-11ef-a8a4-b44506948d64', '2024-11-18
00:00:00', '2024-11-19', '2024-11-
22', 'Rejected', 'user@gmail.com', 'MF1223VF34', 'admin@gmail.com', 'Reject', '2024-
11-18 00:00:00'), ('fa54d88c-a363-11ef-a8a4-b44506948d64', '2024-11-15
00:00:00', '2024-11-18', '2024-11-
18', 'Completed', 'user@gmail.com', 'DL9TCC1369', 'admin@gmail.com', 'Approve', '202
4-11-15 00:00:00');
/*!40000 ALTER TABLE `booking_requests` ENABLE KEYS */;
UNLOCK TABLES;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR
_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
/*!50003 CREATE*/ /*!50017 DEFINER=`root`@`localhost`*/ /*!50003 TRIGGER
`booking_requests_AFTER_UPDATE` AFTER UPDATE ON `booking_requests` FOR EACH
ROW BEGIN
declare other_bookings int;
if new.request_status = 'Rejected' and old.request_status = 'Pending' then
    update vehicles set availability = 'Available' where license_no =
new.vehicle_id;
END if;

if new.request_status = 'Approved' and old.request_status = 'Pending' then
    update vehicles set availability = 'Unavailable' where license_no =
new.vehicle_id;
END if;

END */;;

```

```

DELIMITER ;
/*!50003 SET sql_mode      = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;

-- 
-- Table structure for table `customer`
-- 

DROP TABLE IF EXISTS `customer`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `customer` (
  `email` varchar(45) NOT NULL,
  `name` varchar(45) NOT NULL,
  `password` varchar(255) NOT NULL,
  `license_id` varchar(16) NOT NULL,
  `mobile_no` varchar(10) NOT NULL,
  `dob` date NOT NULL,
  PRIMARY KEY (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `customer`
-- 

LOCK TABLES `customer` WRITE;
/*!40000 ALTER TABLE `customer` DISABLE KEYS */;
INSERT INTO `customer` VALUES
('user@gmail.com', 'User', '$2a$10$r1RUIFDnQZ4yFKrkPs5glepf/itAsIfCuirP325S.lgIqcA/Z/j0e', 'DL-1420110012345', '9876543210', '2004-04-01'), ('user2@gmail.com', 'User2', '$2a$10$p3zZnsiKqKIdLQOk2RXrW0gfx18cq9PLIGk9yJlhGtLwl.p1Gm5zW', 'DL-2420110012345', '9276543210', '2004-02-15'), ('user3@gmail.com', 'User3', '$2a$10$aDhirlcY3P0DCvEjjpph0.gkqX5/V714NuTqRn21St0pcddSgko8e', 'DL-1420110012346', '9276543211', '2000-02-17');
/*!40000 ALTER TABLE `customer` ENABLE KEYS */;
UNLOCK TABLES;

-- 
-- Table structure for table `manager`
-- 

DROP TABLE IF EXISTS `manager`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `manager` (

```

```

`email` varchar(45) NOT NULL,
`name` varchar(45) NOT NULL,
`password` varchar(255) NOT NULL,
PRIMARY KEY (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `manager`
-- 

LOCK TABLES `manager` WRITE;
/*!40000 ALTER TABLE `manager` DISABLE KEYS */;
INSERT INTO `manager` VALUES
('admin@gmail.com','Admin','$2a$10$JZHSUXa2KJ52oNB.8QaeXucRZ0n50pGCLFXH0rt8Nk
Ah5a2BDTxi'), ('admin2@gmail.com','Admin
2','$2a$10$8uDDK7B87GcQDlTm1c7U8enTAZf1KlMu7UHOCh8yLU8LLVAbxwZFe'), ('manager@g
mail.com','Manager','$2a$10$AhkVm6h3LCwiHr9360Nwy.HOsV./g.f/3cdrCjfhAww9Am4nM.
bJy');
/*!40000 ALTER TABLE `manager` ENABLE KEYS */;
UNLOCK TABLES;

-- 
-- Table structure for table `vehicles`
-- 

DROP TABLE IF EXISTS `vehicles`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8mb4 */;
CREATE TABLE `vehicles` (
  `license_no` varchar(10) NOT NULL,
  `vehicle_name` varchar(45) NOT NULL,
  `model_year` year NOT NULL,
  `price_per_day` decimal(10,0) NOT NULL,
  `seating_capacity` int NOT NULL,
  `fuel_type` varchar(10) NOT NULL,
  `vehicle_image` longtext,
  `vehicle_overview` varchar(225) DEFAULT NULL,
  `manager_id` varchar(45) NOT NULL,
  `availability` varchar(45) NOT NULL DEFAULT 'Available',
  PRIMARY KEY (`license_no`),
  KEY `fk_manager_id_idx` (`manager_id`),
  CONSTRAINT `fk_manager_id` FOREIGN KEY (`manager_id`) REFERENCES `manager`(`email`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- 

```

```
-- Dumping data for table `vehicles`
--

LOCK TABLES `vehicles` WRITE;
/*!40000 ALTER TABLE `vehicles` DISABLE KEYS */;
INSERT INTO `vehicles` VALUES ('DL7TC01873','Hyundai
Nexon',2020,6000,7,'Petrol','https://www.hyundai.com/content/dam/hyundai/in/en
/data/find-a-
car/Verna/Highlights/vernapoolside1.jpg','Good','admin@gmail.com','Unavailable
'),('DL9TCC1369','Tata
Nano',2020,1500,7,'Diesel','https://upload.wikimedia.org/wikipedia/commons/thumb/3/3c/Tata_-_Nano_-_Kolkata_2011-09-15_5184.JPG/420px-Tata_-_Nano_-_
_Kolkata_2011-09-15_5184.JPG','Very Good
Car','admin@gmail.com','Available'),('DL9TCC1396','Hyundai EXTER
Knight',2020,8000,8,'Petrol','https://www.hyundai.com/content/dam/hyundai/in/e
n/data/find-a-car/Exter/Highlights/smallimageexter2_5.jpg','Good
Car','admin@gmail.com','Available'),('KA22EF7348','Hyundai
Alcazar',2020,5500,7,'Diesel','https://www.hyundai.com/content/dam/hyundai/in/
en/data/find-a-car/Alcazar/Gallery/alcazargallerybig1.jpg','Good
Car','manager@gmail.com','Available'),('MF1223VF34','Hyundai IONIQ
5',2024,5000,5,'Electric','https://ioniq5.hyundai.co.in/uploads/attachments/cl
uqpl0yg4jiktasahp8ats4c-front-image.png','Good
Car','manager@gmail.com','Available'),('MF412MF', 'Hyundai IONIQ
5',2020,4000,5,'Electric','https://ioniq5.hyundai.co.in/uploads/attachments/cl
uqpl0yg4jiktasahp8ats4c-front-image.png','Good
Car','manager@gmail.com','Available'),('MF412MF412','Hyundai IONIQ
5',2022,4500,5,'Electric','https://ioniq5.hyundai.co.in/uploads/attachments/cl
uqpl0yg4jiktasahp8ats4c-front-image.png','Good
Car','admin@gmail.com','Available');

/*!40000 ALTER TABLE `vehicles` ENABLE KEYS */;
UNLOCK TABLES;

--

-- Dumping events for database 'ovbms'
--

/*!50106 SET @save_time_zone= @@TIME_ZONE */ ;
/*!50106 DROP EVENT IF EXISTS `CheckCompletedBookings` */;
DELIMITER ;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode            =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR
_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;
```

```

/*!50003 SET @saved_time_zone      = @@time_zone */ ;;
/*!50003 SET time_zone            = 'SYSTEM' */ ;;
/*!50106 CREATE*/ /*!50117 DEFINER=`root`@`localhost`*/ /*!50106 EVENT
`CheckCompletedBookings` ON SCHEDULE EVERY 1 DAY STARTS '2024-11-19 11:42:39'
ON COMPLETION NOT PRESERVE ENABLE DO BEGIN

UPDATE vehicles
SET availability = 'Available'
WHERE license_no IN (
    SELECT vehicle_id
    FROM booking_requests
    WHERE request_status = 'Approved'
    AND to_date < CURDATE()
);

UPDATE booking_requests
SET request_status = 'Completed'
WHERE request_status = 'Approved'
AND to_date < CURDATE();

END */ ;;
/*!50003 SET time_zone      = @saved_time_zone */ ;;
/*!50003 SET sql_mode       = @saved_sql_mode */ ;;
/*!50003 SET character_set_client  = @saved_cs_client */ ;;
/*!50003 SET character_set_results = @saved_cs_results */ ;;
/*!50003 SET collation_connection = @saved_col_connection */ ;;
DELIMITER ;
/*!50106 SET TIME_ZONE= @save_time_zone */ ;

-- 
-- Dumping routines for database 'ovbms'
-- 

/*!50003 DROP PROCEDURE IF EXISTS `BookVehicle` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode             =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR
_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `BookVehicle`(in customer_email
varchar(45), in vehicle_license_no varchar(10), in from_date date, in to_date
date)
BEGIN

```

```

insert into booking_requests(request_id, request_date, from_date, to_date,
customer_id, vehicle_id) values(UUID(), CURDATE(), from_date, to_date,
customer_email, vehicle_license_no);
update vehicles set availability = 'Waiting' where license_no =
vehicle_license_no;
END ;;
DELIMITER ;
/*!50003 SET sql_mode      = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `HandleBookingRequest` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results    = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8mb4 */ ;
/*!50003 SET character_set_results = utf8mb4 */ ;
/*!50003 SET collation_connection  = utf8mb4_0900_ai_ci */ ;
/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
/*!50003 SET sql_mode      =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR
_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `HandleBookingRequest`(
  in requestId varchar(255), in action varchar(15), in managerId varchar(45))
BEGIN
if action = 'Approve' then
update booking_requests set request_status = 'Approved', action_type =
'Approve', action_date = curdate(), manager_id = managerId where request_id =
requestId;
elseif action = 'Reject' then
update booking_requests set request_status = 'Rejected', action_type =
'Reject', action_date = curdate(), manager_id = managerId where request_id =
requestId;
END IF;
END ;;
DELIMITER ;
/*!50003 SET sql_mode      = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;  
  
-- Dump completed on 2024-11-28 15:16:12
```

11. GitHub Repo Link

GitHub [link](#) for [Online Vehicle Booking Management System](#)