Wheel o Rent

A PROJECT REPORT

BY

TEAM NO. 01

Divyansh Jain (E23CSEU1042)

Akshat Awasthi (E23CSEU1036)

Parth Azad (E23CSEU1032)

Dhruv (E23CSEU1031)

A blue and red text on a black background

Description automatically generated

SUBMITTED TO

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY, BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

April 2025

# DECLARATION

I/We hereby declare that the work which is being presented in the report entitled Wheel O Rent , is an authentic record of my/our own work carried out during the period from JAN, 2025 to April, 2025 at School of Computer Science and Engineering and Technology, Bennett University Greater Noida.

The matters and the results presented in this report has not been submitted by me/us for the award of any other degree elsewhere.

Signature of Candidate

Divyansh Jain

(E23CSEU1042)

Akshat Awasthi

(E23CSEU1036)

Parth Azad

(E23CSEU1032)

Dhruv

(E23CSEU1031)

# ACKNOWLEDGEMENT

I/We would like to take this opportunity to express my/our deepest gratitude to my/our mentor, **Dr. Neeraj** for guiding, supporting, and helping me/us in every possible way. I/we was/were extremely fortunate to have him as my/our mentor as he provided insightful solutions to problems faced by me/us thus contributing immensely towards the completion of this capstone project. I/We would also like to express my/our deepest gratitude to VC, DEAN, HOD, faculty members and friends who helped me/us in successful completion of this capstone project. Any other name you can mentioned here.

Divyansh Jain

(E23CSEU1042)

Akshat Awasthi

(E23CSEU1036)

Parth Azad

(E23CSEU1032)

Dhruv

(E23CSEU1031)

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

LIST OF ABBREVIATIONS viii

ABSTRACT ix

1. INTRODUCTION 1

1.1. Problem Statement 1

2. Background Research 2

2.1. Proposed System 2

2.2. Goals and Objectives 2

3. Project Planning 3

3.1. Project Lifecycle 3

3.2. Project Setup 3

3.3. Stakeholders 4

3.4. Project Resources 4

3.5. Assumptions 5

4. Project Tracking 5

4.1. Tracking 5

4.2. Communication Plan 6

4.3. Deliverables 7

5. SYSTEM ANALYSIS AND DESIGN 8

5.1. Overall Description 8

5.2. Users and Roles 8

5.3. Design diagrams/ UML diagrams/ Flow Charts/ E-R diagrams 9

5.3.1. Use Case Diagrams 9

5.3.2. Class Diagram 10

5.3.3. Activity Diagrams 11

5.3.4. Sequence Diagram 12

5.3.5. Data Architecture 13

6. User Interface 14

6.1. UI Description 14

6.2. UI Mockup 14

7. Algorithms/Pseudo Code 15

8. Project Closure 16

8.1. Goals / Vision 16

8.2. Delivered Solution 16

8.3. Remaining Work 16

REFERENCES 17

LIST OF TABLES

Table Page

Table 1: Goal and Objectives 2

TABLE 2: SETUIP 3

Table 3: sTAKEHOLDER 4

Table 4: pROJECT rESOURCES 4

Table 5: aSSUMPTIONS 5

Table 6: tRACKING 5

Table 7: Regularly Scheduled Meetings 6

Table 8: Information To Be Shared Within Our Group 6

Table 9: Information To Be Provided To Other Groups 6

Table 10: Information Needed From Other Groups 7

Table 11: Deliverables 7

Table 12: uSER AND rOLES 8

LIST OF FIGURES

Figure 1: use-case diagram 9

Figure 2: cLASS 11

Figure 3: aCTIVITY 12

Figure 4: sEQUENCE 13

Figure 5: dATA aRCHITECTURE 13

Figure 6: ui mOCKup 14

LIST OF ABBREVIATIONS

Abbreviation Explanation of the Abbreviation

⁠MERN MongoDB, Express.js, React.js, Node.js

⁠UI User Interface

API Application Programming Interface

⁠RBAC Role-Based Access Control

⁠EV Electric Vehicle

⁠DB Database

⁠MVP Minimum Viable Product

ABSTRACT

This report describes the design, development, and impact of a web-based vehicle rental platform developed using the MERN stack. The primary motivation is to provide a fair, transparent, and student-friendly alternative to expensive and inconsistent local transportation. Students can rent vehicles on an hourly or daily basis, complete with features like digital license verification, secure payments, and real-time availability. This system provides technical innovation along with tangible social benefits, especially for college students with limited mobility options.

1. INTRODUCTION

* Transportation plays a pivotal role in the everyday lives of students, particularly those who commute daily to attend classes or access essential services. In many developing regions, students often rely on local transport such as auto-rickshaws and three-wheelers, which frequently suffer from inflated and inconsistent pricing. Our project, the Vehicle Rental Web Application, is designed to offer a reliable, affordable, and transparent alternative to local transport through a car rental system tailored specifically to student needs.
* This report outlines the conceptualization, technical development, features, and the anticipated social and economic impacts of the platform. It highlights how modern web technologies can be leveraged to solve real-world problems in a secure, scalable, and user-friendly manner.
  1. Problem Statement

Transportation plays a pivotal role in the everyday lives of students, particularly those who commute daily to attend classes or access essential services. In many developing regions, students often rely on local transport such as auto-rickshaws and three-wheelers, which frequently suffer from inflated and inconsistent pricing. Our project, the Vehicle Rental Web Application, is designed to offer a reliable, affordable, and transparent alternative to local transport through a car rental system tailored specifically to student needs.

This report outlines the conceptualization, technical development, features, and the anticipated social and economic impacts of the platform. It highlights how modern web technologies can be leveraged to solve real-world problems in a secure, scalable, and user-friendly manner.

1. Background Research

Existing platforms like Zoomcar or Drivezy serve the public and don't offer features or pricing optimized for student needs. Additionally, their identity verification systems are complex and not student friendly. There’s a lack of dedicated platforms that allow short-term, affordable rentals backed by educational partnerships.

* 1. Proposed System

A modern MERN stack-based rental platform that allows students to:

•⁠ ⁠Register securely

•⁠ ⁠Rent vehicles hourly or daily

•⁠ ⁠Upload driving license digitally

•⁠ ⁠Get verified and approved by an admin

•⁠ ⁠Pay securely online

•⁠ ⁠Track rental history and upcoming bookings

* 1. Goals and Objectives

Table : Goal and Objectives

|  |  |
| --- | --- |
| **#** | **Goal or Objective** |
| 1 | ⁠Ensure digital-first, mobile-responsive access |
| 2 | ⁠Implement RBAC for safe access to resources |
| 3 | ⁠Reduce transportation costs for students |
| 4 | ⁠Create a socially conscious digital product |
| 5 | ⁠Improve convenience, security, and traceability |

1. Project Planning

This section covers the details of the project planning. Selecting the lifecycle of the development, project stakeholders, resources required, assumptions made (if any) are detailed in the sections below.

* 1. Project Lifecycle

The project follows Agile with 3-week sprints, encompassing:

•⁠ ⁠Sprint 1: Requirement Analysis & UI Design

•⁠ ⁠Sprint 2: Frontend & Backend Development

•⁠ ⁠Sprint 3: Integration & Testing

•⁠ ⁠Sprint 4: Deployment & Documentation

* 1. Project Setup

Table : Setup

|  |  |
| --- | --- |
| **#** | **Decision Description** |
| 1 | ⁠Codebase managed on GitHub |
| 2 | ⁠Backend with Node.js and Express.js |
| 3 | ⁠Frontend with React.js and Tailwind CSS |
| 4 | MongoDB Atlas for cloud database |

* 1. Stakeholders

Table : Stakeholders

|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| Students | Primary Users |
| Admins | ⁠Admins: Verify users and manage content |
| Vehicle Owners | Optionally allowed in future releases |
| ⁠University Officials | For optional integration |

* 1. Project Resources

Table : Project Resources

|  |  |  |
| --- | --- | --- |
| **Resource** | **Resource Description** | **Quantity** |
| Database Server | A database server provided by the sponsoring company. | 1 |
| Capstone Team | Our team of students who will be the primary developers of the project. | 4 |
| Jim Somebody | The mentor who will be able to provide us with technical assistance. | 1 |
| Mac Workstation | An OS X workstation with X Code for developing the OS X version of the software. | 1 |
| Android Phone | An Android phone to be used as test hardware for the mobile version of the software. | 2 |

* 1. Assumptions

Table : Assumptions

|  |  |
| --- | --- |
| **#** | **Assumption** |
| A1 | ⁠Internet access is available |
| A2 | ⁠Students own or can access driving licenses |
| A3 | ⁠Basic mobile/computer literacy exists among users |
| A4 | ⁠The institution may assist in verification |

1. Project Tracking
   1. Tracking

Table : Tracking

|  |  |  |
| --- | --- | --- |
| **Information** | **Description** | **Link** |
| Code Storage | Project code will be stored in Github repository. | [Link](https://github.com/AkshatAwa/Wheelorent) |
| Bug Tracking | Bug tracking will be done with Trac. | [Link](https://github.com/AkshatAwa/Wheelorent) |
| Project Documents and Assignments | Weekly reports, specification and design documents, etc. will be stored in our Github repository. | [Link](https://github.com/AkshatAwa/Wheelorent) |
| Continuous Integration | Continuous integration will be done with github | [Link](https://github.com/AkshatAwa/Wheelorent) |

* 1. Communication Plan

Table 7: Regularly Scheduled Meetings

|  |  |  |
| --- | --- | --- |
| Meeting Type | Frequency/Schedule | Who Attends |
| Conference Call/Skype | Weekly | Project team and mentor |
| Team Meeting | Weekly | Project team |
| Short Meeting | Weekly in class | Project team |
| Sprint Planning Meeting | Start of each sprint | Project team and mentor |
| Sprint Retrospective Meeting | End of each sprint | Project team |
| Sprint Review Meeting | End of each sprint | Project team, ***mentor, and sponsor*** |

Table 8: Information To Be Shared Within Our Group

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Project team | Task assignments & General scrum information | Weekly | Team meetings, listing in Project Specification. |

Table 9: Information To Be Provided To Other Groups

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Sponsor and mentor | Final deliverables | At completion of project | Project specification doc., code, Power Point presentation |
| Sponsor and mentor | Weekly report | Weekly | Email and Trac site access |
| Sponsor and mentor | Project baselines ***(optional)*** | At the end of each sprint | Onsite customer demo, access to repository |

Table 10: Information Needed from Other Groups

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Sponsor and mentor | Requirement changes | Start of each sprint | Conference call or meeting with sponsor and mentor. |
| Nathan Olson | Availability of test server | Start of second sprint | Email |

* 1. Deliverables

Table 11: Deliverables

|  |  |
| --- | --- |
| **#** | **Deliverable** |
| 1 | **Wireframes and Ui Designs** |
| 2 | MVP with all core features |
| 3 | Final Deployment |
| 4 | ***Project Report*** |
| 5 | Final report (final PowerPoint presentation, 3 minute video, and final sprint) |

1. SYSTEM ANALYSIS AND DESIGN
   1. Overall Description
2. The system is a multi-role web application built using the MERN stack. It consists of:
   1. A frontend (React.js) for users and admins to interact with the system
   2. A backend server (Node.js & Express.js) handling API requests, business logic, and authentication
   3. A NoSQL database (MongoDB) to store users, cars, rentals, and transactions
   4. Integration with third-party APIs such as Auth0 for authentication, Stripe for payments, and Cloudinary for image uploads
3. The system is divided into three major modules:
   1. **User Module** – Registration, login, profile management, car browsing, and booking
   2. **Admin Module** – User management, car approval/rejection, rental monitoring
   3. **Rental Module** – Time-based car renting with hourly and daily rates, secure license verification, and rental status tracking
   4. Users and Roles

Table 12: User and Roles

|  |  |
| --- | --- |
| **User** | **Description** |
| Student / User | User can do following things:   1. Register/login 2. Browse available cars 3. Rent cars hourly or daily 4. Upload Driving License 5. View Rental history 6. Give your car on rent |
| Admin | Admin can do following things:   1. Manage user accounts 2. Verify driving licenses 3. Approve or Reject Car listings 4. Monitor Rental transaction |

* 1. Design diagrams/Architecture/ UML diagrams/ Flow Charts/ E-R diagrams
     1. Use Case Diagram

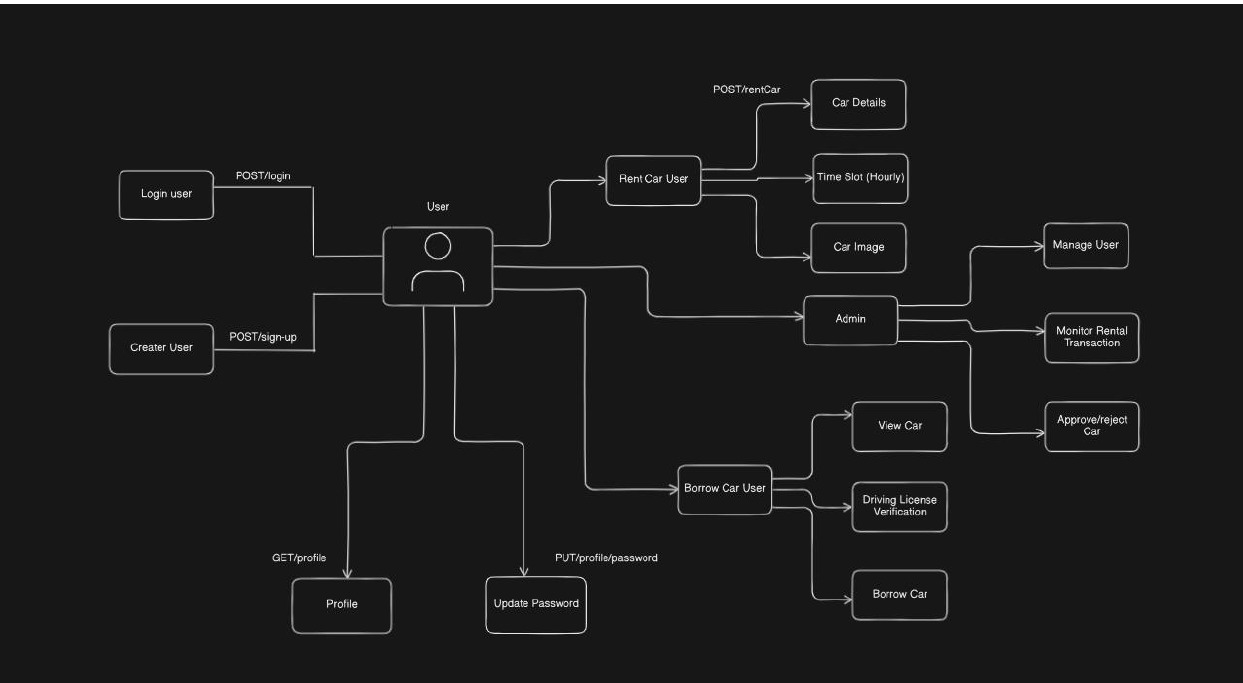


Figure : Use-Case diagram

Illustrates interaction between users (student, admin) and system components. Key use cases include login, register, rent car, upload license, view profile (for users), and manage user, approve cars, monitor transactions (for admin).

* + 1. Class Diagram

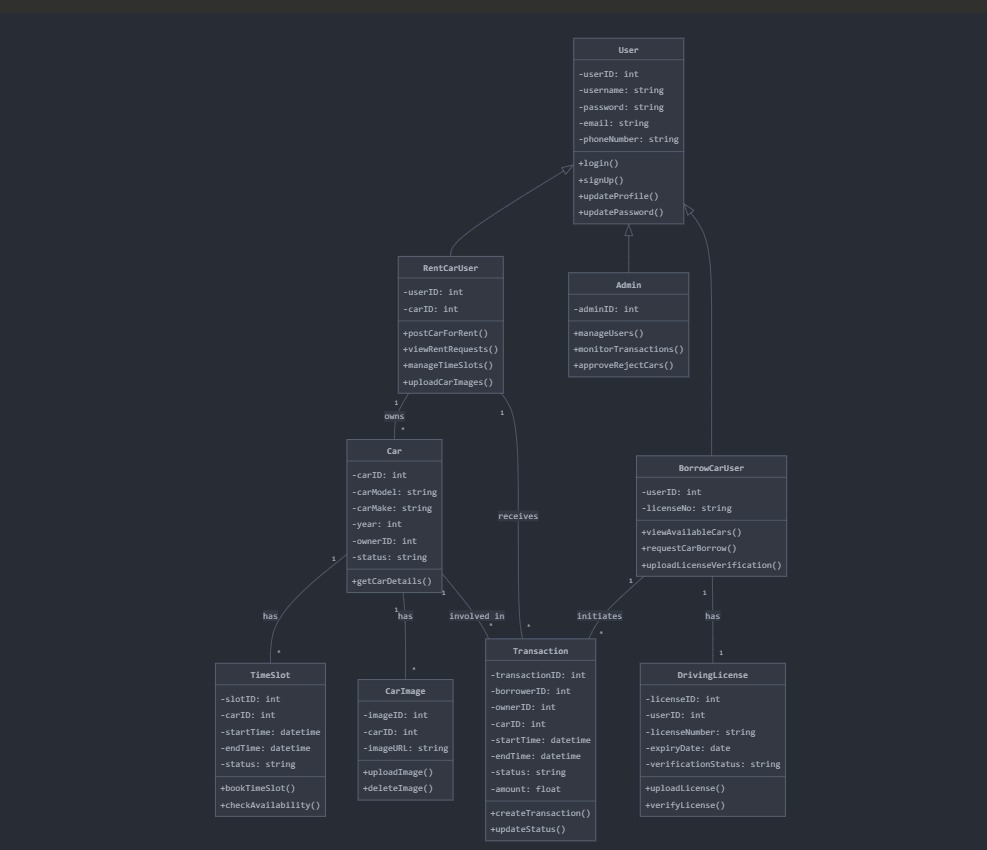


Figure : Class Diagram

Main classes:

* **User** (attributes: name, email, password, role, license)
* **Car** (attributes: title, image, description, availability, rate)
* **Rental** (attributes: userId, carId, timeSlot, duration, paymentStatus)
* **Admin** (inherits User, additional permissions) Relationships: One-to-many between User and Rental, and Car and Rental.
  + 1. Activity Diagrams

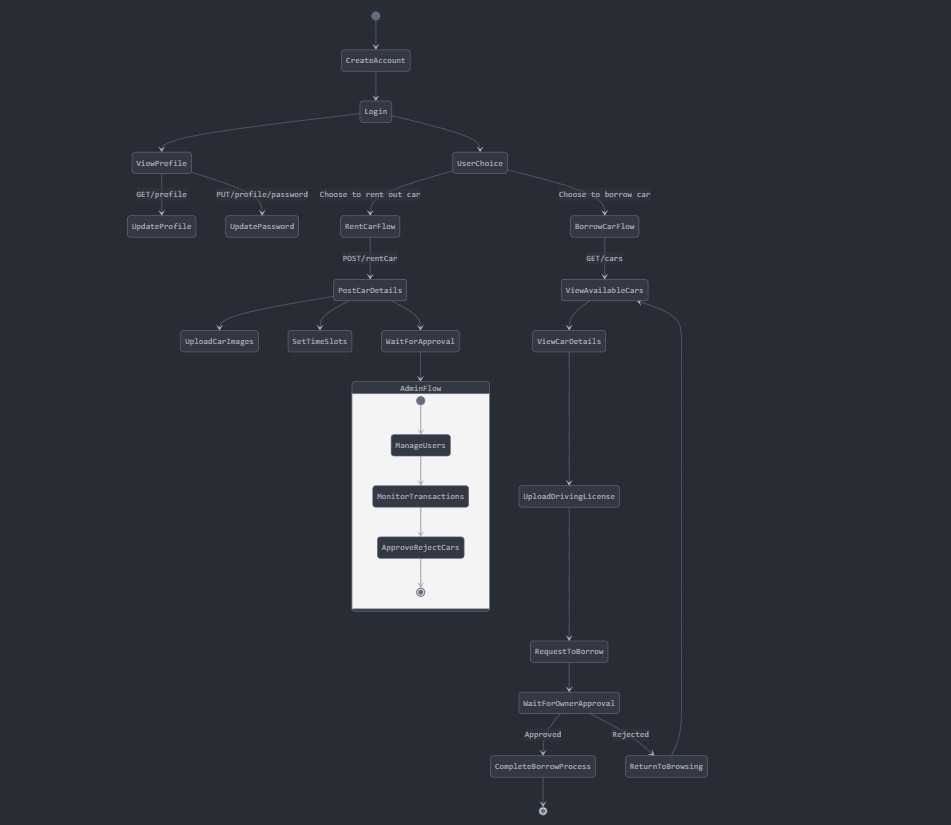


Figure : Activity Diagram

Activity flow for booking a car:

1. User logs in
2. Searches for available cars
3. Selects car and rental duration
4. Uploads license if not uploaded
5. Proceeds with payment
6. Booking confirmed, rental stored
   * 1. Sequence Diagram



Figure : Sequence Diagram

Sequence of interactions:

* User → Login → API → Auth validation → Success
* User → Select Car → Rent API → Validate license → Payment → Confirm rental
* Admin → Approve car → Notify user
  + 1. Data Architecture

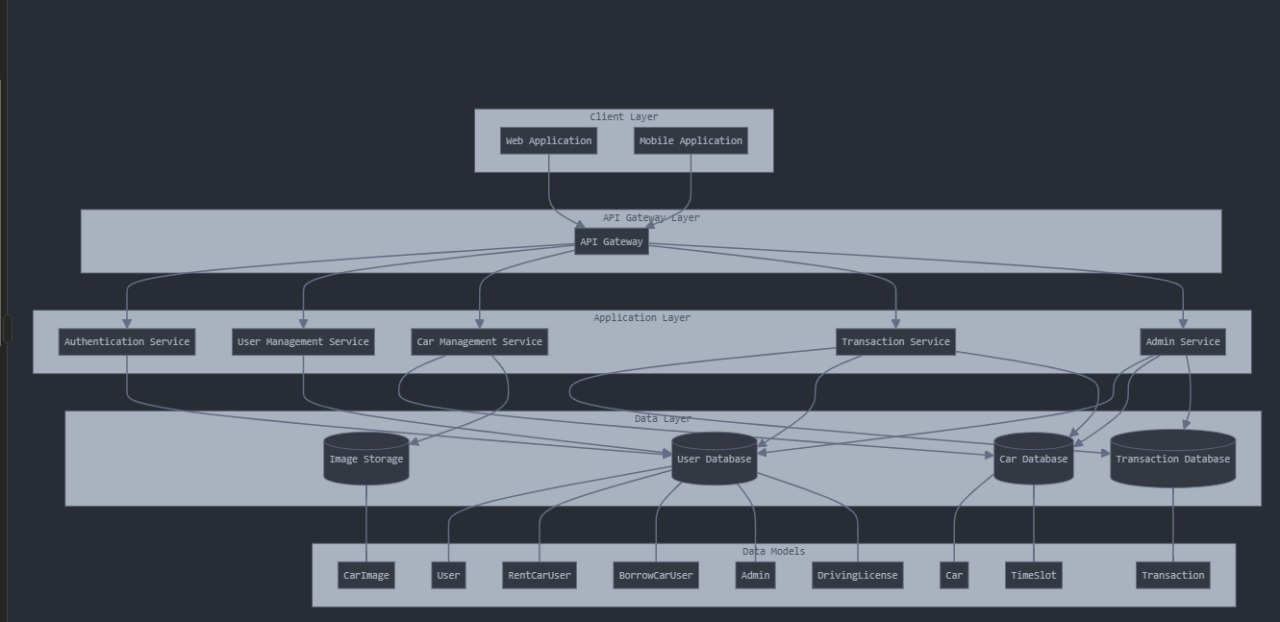


Figure : Data Architecture Diagram

MongoDB database with collections:

* **Users** – stores user data with roles and encrypted credentials
* **Cars** – details of listed cars and availability
* **Rentals** – transaction history, booking details

1. User Interface
   1. UI Description

The user interface is designed for minimal friction with mobile responsiveness. TailwindCSS is used for styling. Forms are user-friendly, and dashboards provide clear feedback and car details.

* 1. UI Mockup

Mockups built with Figma/Canva. Includes:

* Login/Register Screens
* Car Listings
* Booking Screen
* Admin Dashboard (Attach screenshots in appendix)

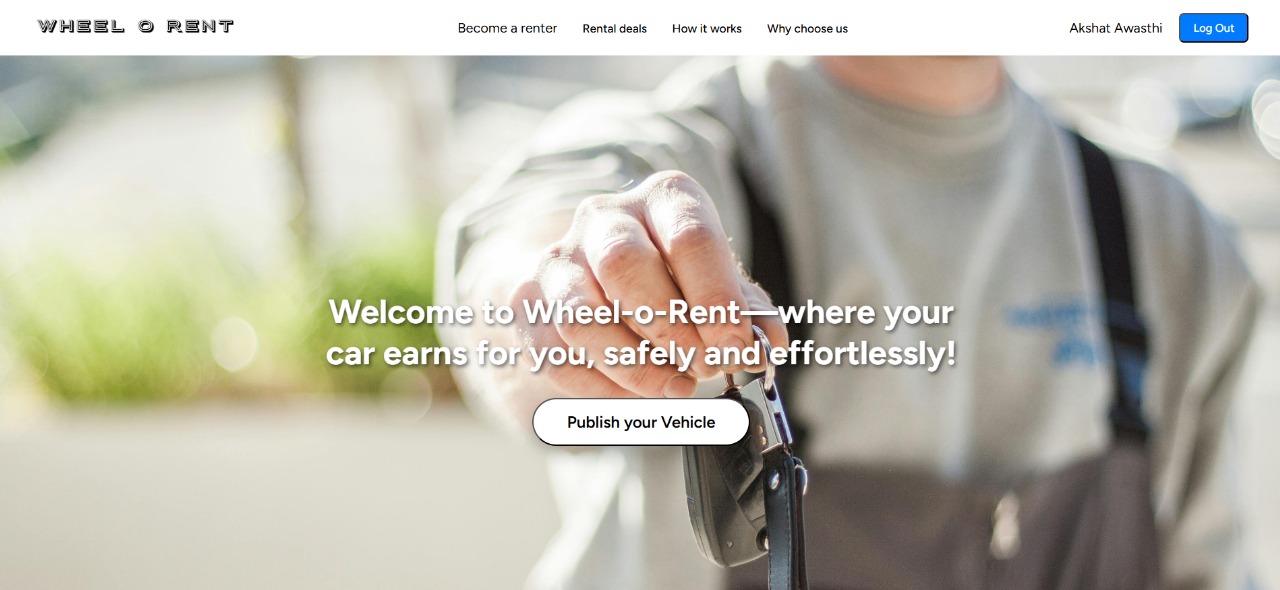


Figure 6: UI of Publishing Page

1. Algorithms/Pseudo Code OF CORE FUNCTIONALITY

**a. Role Check Middleware**

function isAdmin(req, res, next) {

if (req.user.role !== 'admin') {

return res.status(403).send('Forbidden');

}

next();

}

**b. Rental Price Calculation**

function calculatePrice(hours, rateHourly, rateDaily) {

return hours <= 5 ? hours \* rateHourly : rateDaily;

}

**c. Secure License Upload**

**const encrypted = encrypt(req.file.buffer);**

**cloudinary.upload(encrypted);**

1. Project Closure
   1. Goals / Vision

* To enhance campus mobility with affordable, reliable transportation for students.

8.2 Delivered Solution

* + Full-stack MERN application
  + Authenticated rentals with Stripe payments
  + Admin interface for monitoring
  + Encrypted license verification

8.3 Remaining Work

* + Android/iOS app development
  + Vehicle health tracking using IoT
  + Integration with university portals

**REFERENCES**

1. MERN Stack Official Documentation [Link](https://www.mongodb.com/resources/languages/mern-stack-tutorial)
2. MongoDB Schema Best Practices [Link](https://www.mongodb.com/basics/schema-design-best-practices)
3. Stripe Payment Integration Docs [Link](https://stripe.com/docs/payments)
4. Auth0 Authorization Guides [Link](https://auth0.com/docs/get-started)
5. Cloudinary Image Uploading Docs [Link](https://cloudinary.com/documentation/image_upload_api_reference)
6. OWASP Web Security Standards [Link](https://owasp.org/www-project-top-ten/)
7. GitHub Repositories and Issues [Link](https://github.com/search?q=topic%3Aopen-source)