DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

CU CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Report File FULL STACK

Student Name: Sukhjinder Singh

UID: 23BAI70078

Branch: BE-AIT-CSE

Section/Group: 23AIT-KRG-G2

Semester: 5th

Date of Performance: 25th Aug, 2025

Subject Name: Full Stack

Subject Code: 23CSP-339

Project Report: WheelX – A Car Rental Web Service

---

1. AIM

The aim of this project is to build **WheelX**, an online **Car Rental Web Service** where users and administrators can log in and manage car rental operations.

User Features:

- Register and log in securely.
- Browse available cars for rent.
- Book and manage car rentals.
- View booking history.

Admin Features:

- Log in as admin with secure authentication.
- Add, update, or delete car listings.
- Manage user bookings and rental status.
- Upload car images using Cloudinary API.

This project helps students understand **frontend–backend integration**, **authentication using JWT**, and **database management with PostgreSQL**.

---

2. TECH STACK

Frontend

- **React.js** → Interactive and dynamic user interface.

- **Tailwind CSS** → Modern, responsive styling.

Backend

- **Spring Boot (Java)** → Backend framework for API development and logic handling.
- **JWT (JSON Web Token)** → Used for secure authentication and authorization.
- **Cloudinary API** → Stores and manages car images on the cloud.

Database

- **PostgreSQL** → Stores user data, car details, and booking information.

Tools

- **VS Code & IntelliJ IDEA** → Code editors for frontend and backend.
- **Postman** → Testing REST APIs.
- **Browser** → Running and testing the web application.

Security

- **JWT (JSON Web Token)** → Ensures secure login for both users and admins.

---

3. THEORY / BACKGROUND

The **WheelX Car Rental System** follows the **client-server architecture**:

- **Client (React + Tailwind CSS):**
  Displays car listings, handles user/admin login, and manages booking interactions.
- **Server (Spring Boot):**
  Provides REST APIs for login, registration, car management, and booking services.
- **Database (PostgreSQL):**
  Stores user credentials, car details, images (links via Cloudinary), and booking records.

Cloudinary Integration:
Used to upload and store images of cars securely and efficiently in the cloud.

JWT Authentication Flow:

1. User/Admin logs in with credentials.
2. Server generates a JWT token.
3. Token is used to authenticate and authorize all further API requests securely.

REST API Endpoints:

- `POST /auth/login` → Authenticate user or admin and generate JWT.
- `POST /auth/register` → Register new users.
- `GET /cars` → Fetch all available cars.
- `POST /cars` → Add a new car (Admin only).
- `PUT /cars/:id` → Update car details (Admin only).
- `DELETE /cars/:id` → Delete a car (Admin only).
- `POST /bookings` → Create a booking.

- `GET /bookings/:userId` → Fetch user bookings.

---

4. PROCEDURE / IMPLEMENTATION

Step 1: Setup Environment

1. Install **Node.js**, **Java (JDK)**, and **PostgreSQL**.
2. Create a folder named `wheelx-app`.
3. Setup **React frontend** and **Spring Boot backend** projects separately.

Step 2: Backend Development (Spring Boot + JWT + Cloudinary)

- Initialize Spring Boot project using Spring Initializr.
- Add dependencies:
  `Spring Web`, `Spring Data JPA`, `PostgreSQL Driver`, `Spring Security`, and `JWT`.
- Configure Cloudinary API for image upload.
- Create REST APIs for authentication, car management, and booking services.
- Connect backend to PostgreSQL database.

Step 3: Frontend Development (React + Tailwind CSS)

- Create UI components for:
  - Login/Register pages.
  - Car listings and booking forms.
  - Admin dashboard for car management.
- Use **Axios** or **Fetch API** to connect frontend with backend APIs.
- Apply Tailwind CSS for responsive and modern design.

Step 4: Run and Test Application

- **Start backend server:**
- `mvn spring-boot:run`
- **Start frontend:**
- `npm start`
- **Test APIs using Postman** for authentication, car listing, and booking.
- Open app in browser and verify all user and admin features.

---

5. CONCLUSION

The **WheelX Car Rental Web Service** demonstrates a full-stack web application with secure authentication and real-time interaction.

It provides practical experience in:

- **Frontend:** Responsive design with React and Tailwind CSS.
- **Backend:** Secure REST APIs with Spring Boot and JWT.
- **Database:** Data persistence and management using PostgreSQL.
- **Cloud Integration:** Storing and retrieving images using Cloudinary API.

This project showcases skills in **web security, API communication, and cloud integration** — essential concepts for building scalable, production-ready web applications.
