

On Job Training (OJT) Assessment Report

**DLithe Consultancy Services Pvt. Ltd.**

**On Job Training (OJT) Assessment 2**

**Trainee**/**InternName**: Abdul Ahad

**Reg. no: 4MT21CS003**

**Period**: 4 Months (January 17th - June 17th)

**Job Assignment**: Python Full Stack.

**Organization**: DLithe Consultancy Services Pvt. Ltd. **Supervisor Name**: Kaveri S B

**Observations:**

Our Project is cleanly broken into logical modules (Vehicle Listing, Booking, Customer Management, UI), which will help with parallel development, easier debugging, and code maintainability.

Using **Flask (Python)** as the backend and **React** as the frontend strikes a good balance between simplicity and flexibility. Flask's lightweight nature allows you to quickly build RESTful APIs to serve the React frontend.

## Submitted to

Signature of Training Supervisor Signature of Cohort Owner

**Date**: **Date**:

## Letter of Transmittal

**To:**

Head of Technology,

DLithe Consultancy Services,

Mangalore

**Dear Madam,**

I am writing to formally submit my report on the **PythonFull Stack** job training I recently underwent, undertaken as part of my professional development within our esteemed organization.

The comprehensive report details the training program, key learnings, and recommendations for incorporating Python Full Stack into our operations. Throughout the training program, I actively participated in practical exercises and engaged in discussions with fellow participants from Different Colleges.

This enabled me to gain a comprehensive understanding of current trends, challenges, and best practices in Java development. I believe that the knowledge and skills acquired during the training will significantly enhance our organization's operations and competitiveness in the ever-evolving business landscape.

Sincerely,

**Name:** Abdul Ahad

**Reg. No:** 4MT21CS003

**College Name:** Mangalore Institute Of Technology And Engineering

**Department:** Computer Science And Engineering

Check the content and update this index page

Table of Contents

[Introduction 5](#_heading=h.gjdgxs)

[Applications of Backend 6](#_heading=h.1fob9te)

[Benefits 6](#_heading=h.3znysh7)

[Challenges 7](#_heading=h.2et92p0)

[Conclusion 7](#_heading=h.tyjcwt)

# Introduction

The **Car Rental Management System** is a web-based application designed to simplify and modernize the management of car rental services. The system aims to digitize core activities such as vehicle listings, rental bookings, and customer management, offering an efficient and user-friendly experience for both administrators and customers. Built using **React.js** for the frontend and **Flask (Python)** for the backend, the application provides a seamless interface to manage rental operations, reduce manual tasks, and improve service quality.

The primary use cases of the system include managing vehicle inventories, facilitating customer reservations, and maintaining customer profiles and booking histories. Administrators can easily add, update, and remove vehicle listings with details like model, rental price, and availability. Customers will be able to browse available vehicles, make reservations, and manage their bookings through a responsive interface. Additionally, an admin dashboard will allow staff to oversee daily operations, handle customer queries, and view analytics to support business decisions.

By leveraging modern web technologies and a modular architecture, this system will streamline car rental workflows, minimize booking conflicts, and improve data management. The goal is to offer a scalable and reliable solution that enhances both operational efficiency and customer satisfaction, making car rental services more accessible and convenient for all users.

## b. Introduction To Software Development Process

The development of the **React-Based Car Rental Management System** follows a structured software development process to ensure systematic planning, design, implementation, testing, and deployment. By following these clearly defined stages, the project can achieve its objectives efficiently while ensuring quality and maintainability.

In this initial phase, we identify and understand the functional and non-functional requirements of the system. The goal is to gather all necessary information about vehicle management, booking processes, and customer management to ensure that the final product meets user expectations.

**System Design**

The system’s architecture and design are defined in this phase. The application structure, module interactions, database schema, and API design are outlined to provide a blueprint for development. Visual representations like mind-maps and flow diagrams help clarify the design.

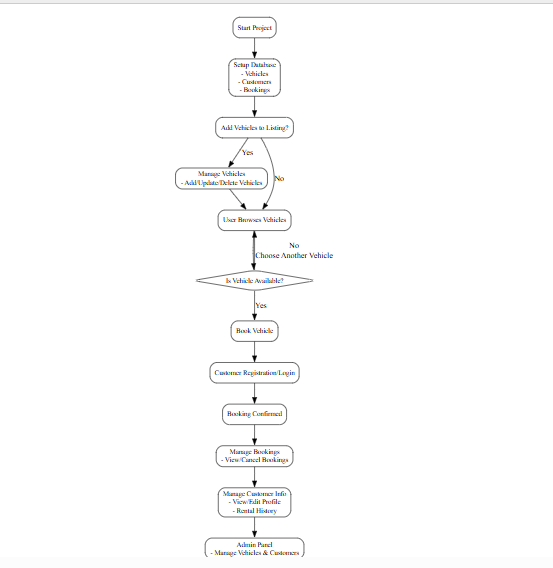


Fig: Flow Chart

# 

# proj m.png

# Fig : Mind Map

# Technologies Used To Implement The Project

**Backend Python (Flask) and MySQL** find numerous applications across various domains in software development. Here are some common applicat

**Front-end Development:Technologies Used:** React.js, HTML5, CSS3ions:, JavaScript,

For the Car Rental Management System, **React.js** was used to create a responsive, interactive, and dynamic user interface. React’s component-based architecture allowed for reusable UI components such as vehicle listings, booking forms, and customer dashboards. Styling and layout were enhanced using **Material-UI** and **CSS3**, while **HTML5** and **JavaScript** provided the foundation for frontend development.

These technologies together ensured a smooth user experience, enabling functionalities like browsing vehicles, making bookings, and managing customer accounts efficiently.

**Back-End Development:Technologies Used:** Python (Flask), MySQL

The backend of the project was implemented using **Python Flask**, a lightweight and flexible micro-framework ideal for building RESTful APIs quickly and efficiently. Flask handled HTTP requests and responses, business logic processing, and communication with the database. **MySQL** served as the relational database to store and manage structured data such as vehicle details, customer records, and booking histories.

This stack provided a scalable and secure backend capable of supporting all core functionalities of the car rental management system — including vehicle management, booking system, and customer data handling.

**API’s Details**

**Technologies Used:** Flask RESTful API, Flask-SQLAlchemy, Flask-JWT (optional for authentication)

The backend APIs were developed using **Flask RESTful** for clean and structured API routes. **Flask-SQLAlchemy** acted as the ORM (Object Relational Mapper), simplifying interaction between Python and the MySQL database. These APIs handled tasks such as managing vehicle listings (add/update/delete),processing booking requests and cancellations and user history

# Conclusion

In conclusion, the **React-Based Car Rental Management System** successfully streamlines the process of managing vehicle listings, rental bookings, and customer information through an efficient and user-friendly web application. By leveraging a modern technology stack — React.js for the frontend and Python Flask with MySQL for the backend — the system delivers a seamless experience for both administrators and customers. Core functionalities such as vehicle management, booking handling, and customer account management were implemented effectively, addressing the practical needs of a car rental service.

Throughout the project, careful attention was given to modular design, scalability, and ease of use. The use of RESTful APIs ensured smooth communication between the frontend and backend, while the database design supported secure and reliable data storage. The responsive user interface enabled users to browse vehicles, make reservations, and manage their accounts intuitively. Backend services ensured robust processing of bookings and management of customer data, contributing to an all-in-one solution for car rental operations.

Overall, the project not only meets the defined objectives but also demonstrates how modern web technologies can simplify traditional business processes. With scope for future enhancements — such as integrating payment calable and efficient car rental management platform that can be expanded to meet evolving user needs.

gateways, advanced analytics, and user authentication — the system lays a strong foundation for a s

# Challenges

During the development of the Car Rental Management System, several challenges were encountered that required thoughtful solutions. Integrating the React frontend with the Flask backend posed initial difficulties, especially in ensuring smooth communication through RESTful APIs and handling cross-origin resource sharing (CORS) issues. Designing a scalable database schema that could efficiently manage complex relationships between vehicles, bookings, and customers also required careful planning. Additionally, implementing features like real-time booking status updates and ensuring data consistency between multiple modules added to the complexity. Balancing frontend responsiveness with backend performance, while maintaining clean code structure and ensuring user-friendly design, was a continuous challenge throughout the project.

# 

# Appendix

**GitHub Link:** [**https://github.com/A-one01/DLithe-Internship-Project**](https://github.com/A-one01/DLithe-Internship-Project)

**PPT Link:**