# **User Documentation**

## **Workshop Management System**

**Team Member names:**

**Praveen Aluru**

**Niharika Balusu**

**Yashasri Syamala**

**Mary Lavanya Thumma**

**Contents**

[1 Overview 3](#_Toc173433248)

[1.1 System Users 3](#_Toc173433249)

[1.2 Key Features and Functionalities 3](#_Toc173433250)

[2 Getting Started 4](#_Toc173433251)

[2.1 Requirements 4](#_Toc173433252)

[2.2 Installation 4](#_Toc173433253)

[3 Configuration 4](#_Toc173433254)

[3.1 Environment Variables 4](#_Toc173433255)

[4 Application Structure 5](#_Toc173433256)

[6 Running the Application 6](#_Toc173433259)

[6.1 Development 6](#_Toc173433260)

1. **Overview**

The **Ride Management System** (RMS) is a comprehensive digital platform designed to streamline the management and operation of rides for transportation services. This system enhances user experience by providing tools for efficient ride management, real-time updates, and improved operational efficiency. The RMS facilitates seamless interaction between three primary user roles: **Admin**, **Driver**, and **Rider**. Admins manage the overall system, monitor ride statuses, and manage driver assignments; Drivers are responsible for accepting ride requests, managing rides, and ensuring passenger safety; and Riders can access ride information, book rides, and track their journey. Built with modern technologies, the platform integrates features such as real-time ride status updates, automated booking, digital payment processing, and detailed ride tracking. The system caters to various transportation environments, from ride-sharing services to traditional taxi services, providing tools for ride creation, operational oversight, rider engagement, and safety compliance. Through its user-friendly interfaces and automated processes, the RMS significantly enhances user satisfaction while reducing operational costs.

**1.1 System Users**

**The system caters to three main user groups:**

**Admin**

* Manage the overall system and monitor ride statuses.
* Manage driver assignments and access control.
* View detailed ride analytics and reports.
* Set up and manage ride schedules, maintenance plans, and safety protocols.

**Driver**

* Accept ride requests and manage rides.
* Monitor ride status and manage operational activities.
* Ensure passenger safety during the ride.
* Access ride analytics and reports.
* Handle digital payment processing for rides.

**Rider**

* View available rides and their status.
* Book rides through the platform.
* Track their ride in real-time.
* Provide feedback and view ratings and reviews for drivers.
* Access digital payment methods and manage bookings.

**1.2 Key Features and Functionalities**

The Ride Management System incorporates a comprehensive suite of features designed to streamline ride management and enhance the user experience across all roles. At its core, the system provides robust administrative controls, allowing system administrators to manage ride schedules, monitor ride statuses, and oversee driver assignments. It offers specialized tools for drivers to manage ride operations, monitor safety protocols, and handle bookings efficiently. For riders, the system provides real-time updates on ride statuses and estimated times, seamless booking processes, and access to digital payment options. The platform includes automated booking and queue management, ensuring optimal utilization of driver resources and minimizing wait times. An integrated payment system allows for easy processing of payments through multiple methods, while detailed ride tracking provides insights into ride performance and safety. Security is maintained through role-based access control and secure authentication, while the system’s automated features handle notifications, ride status updates, and maintenance scheduling. The platform’s scalable architecture ensures reliable performance and data integrity, making it suitable for any size of transportation service while providing user-friendly interfaces accessible across various devices.

1. **Getting Started**
   1. **Requirements**

To run the application, ensure you have the following software installed:

* + - **Python**: 3.13.0 or latest version
    - **Node.js**: 20 or latest version
    - **MongoDB**: v4.4 or latest version
  1. **Installation**

1. Unzip the project
2. Navigate to the project directory/folder:

cd source\_code

1. pip install pipenv
2. pipenv shell
3. pipenv install
4. **Configuration**
   1. **Environment Variables**

The application uses a file to manage configuration settings. The following variables should be defined:

* + - **DATABASE URL**: The URL connection string for MongoDB.

1. **Application Structure**

The project’s framework is Flask and the file structure is as follows:

. frontend: contains all files of UI

. backend: contains all backend files

1. **Running the Application**
   1. **Development**

To start the application in development mode, run:

npm start: Frontend Server

py run.py: Backend Server