Muhammad Mustafa

Project: Mini Ride Booking System

Table of Contents

[**Project Overview** 1](#_Toc202718519)

[**Tech Stack Used & Why** 1](#_Toc202718520)

[**Why React?** 1](#_Toc202718521)

[**Assumptions Made** 2](#_Toc202718522)

[**Ride Status Progression Logic (Simulated Workflow)** 2](#_Toc202718523)

[**ERD Diagram** 3](#_Toc202718524)

[**Project GitHub link** 3](#_Toc202718525)

[**Demo video** 3](#_Toc202718526)

**Mini Ride Booking System – Design Document**

# **Project Overview**

A simple Mini Ride Booking System web app prototype designed for testing in smaller cities. The app allows users to simulate booking rides, tracking ride status, and viewing past rides — all without the need for GPS, backend integration, or live driver management.

# **Tech Stack Used & Why**

|  |  |  |
| --- | --- | --- |
| **Layer** | **Technology** | **Reason** |
| Frontend | React.js | Component-based architecture, ideal for rapid prototyping and dynamic UI |
|  | HTML5/CSS3 | Core structure and styling for the app |
|  | JavaScript (ES6) | Used for app logic, state management, and timers |
| Storage | LocalStorage | Lightweight way to store user session and ride history temporarily |
| Tooling | Create React App | Quick setup tool for bootstrapping a modern React project |

# **Why React?**

React allows us to dynamically update views like ride status and ride history using simple, isolated components. It supports clean state-driven logic, which was essential for simulating ride status changes without a backend.

# **Assumptions Made**

|  |  |
| --- | --- |
| **Area** | **Assumption** |
| **Authentication** | Dummy login only — user enters a name; no password or API validation |
| **Ride Booking** | Driver assignment is simulated — no real drivers or matching logic used |
| **Locations** | Locations are entered as free text (e.g., "Lahore", "Airport") |
| **Ride Types** | Limited to: Bike, Car, and Rickshaw |
| **Multi-user Support** | Single-user session only (no multi-login or server-side auth) |
| **Responsiveness** | Basic layout for desktop view — mobile responsiveness not yet implemented |
| **Data Storage** | Data is stored locally in the browser (localStorage) |

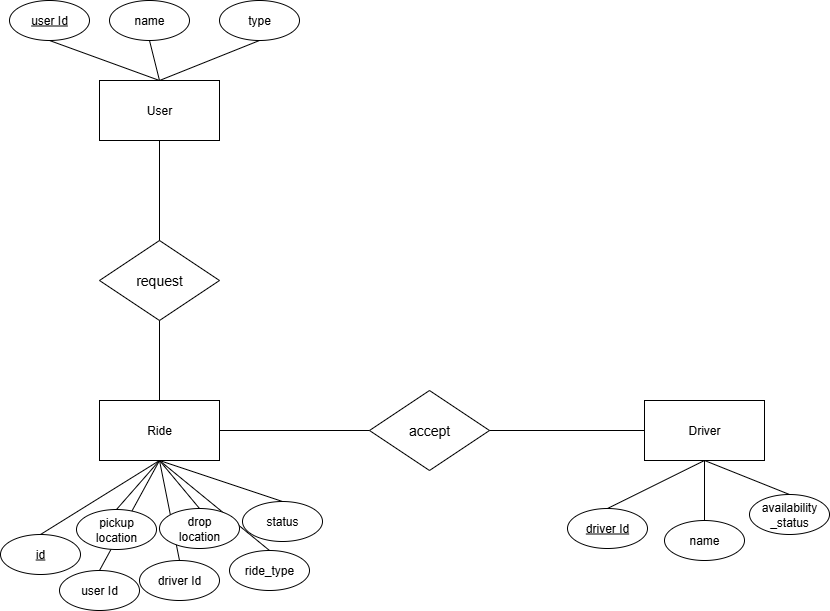
# **Ride Status Progression Logic (Simulated Workflow)**

We used a **time-based simulation** to mimic the real-life flow of a ride request. The system automatically updates the status of a ride after set time intervals without user interaction.

|  |  |
| --- | --- |
| **Time After Booking** | **Ride Status** |
| Immediately | **Requested** — Ride request is sent and pending driver confirmation |
| After 3 seconds | **Accepted** — Driver accepts the ride |
| After 6 seconds | **In Progress** — Ride begins |
| After 10 seconds | **Completed** — Ride ends and moves to ride history |

This functionality is implemented using setTimeout() in React and makes the app feel more realistic while keeping it simple and backend-free.

# **ERD Diagram**



# **Project GitHub link**

<https://github.com/Mustafa25008/Ride-hailing-app.git>

# **Demo video**

<https://youtu.be/_FnuS_XMh8U>