**Problem Statement: Online Cab Booking System (MERN Stack)**

In today’s fast-paced world, users need a seamless and user-friendly platform to book cabs on demand. This capstone project involves developing an **Online Cab Booking System** using the **MERN (MongoDB, Express.js, React, Node.js)** stack, which facilitates real-time cab booking, registration, login, and secure payment through Stripe integration.

**Objectives:**

* Design and develop a web-based cab booking platform.
* Provide functionality for admin to manage cars and drivers.
* Enable users to register, log in, and book cabs based on availability.
* Integrate a secure and testable payment system.
* Store booking and user information in MongoDB.

**🔧 Functional Requirements:**

1. **User Module:**
   * User registration and login.
   * Dashboard view with booking options.
   * Book a cab by selecting start and end dates.
   * View available cars and assign drivers.
   * View booking history and status.
2. **Admin Module:**
   * Admin login.
   * Add and manage cars (car name, model, type).
   * Add and manage drivers.
3. **Payment Module:**
   * Integrate Stripe API in test mode.
   * Dummy card details for testing transactions.
4. **Database:**
   * MongoDB for storing users, drivers, car details, and bookings.

**Generative AI Tools Integration Suggestions:**

To **enhance your existing Cab Service MERN project** with **AI-enabled tools**, here’s a **step-by-step guide** along with **practical enhancements** you can add using generative AI tools across the **Design, Development, Optimization, Testing, and Feature Enhancement** phases.

**How to Use AI Tools in Your Existing Project (Step-by-Step)**

**🖌️ Phase 1: Design Enhancement**

**🔧 Tools: Figma, Uizard, Microsoft Designer**

**🔁 Steps:**

1. **Export existing UI Screenshots** or wireframes from your React frontend.
2. **Upload them to Figma or Uizard** to convert into editable designs.
3. Use **Uizard’s AI prompt feature** to generate improved UI (e.g., “Redesign this dashboard to look modern and clean for a cab service app”).
4. Use **Microsoft Designer** to create:
   * Homepage hero banners.
   * Promotional visuals.
   * Admin UI image content.

**Outcome:**

* Modern UI improvements.
* Rapid prototyping of new screens (e.g., Driver Tracking UI, Smart Suggestions Panel).

**Phase 2: AI-Assisted Development**

**🔧 Tools: GitHub Copilot, ChatGPT**

**Steps:**

1. **Enable GitHub Copilot** in VS Code.
2. Start typing your React components, routes, or backend service methods.
   * E.g., Type getCabAvailability = async () => → Copilot suggests logic automatically.

@workspace **Need React components, routes, or backend service methods for getCabAvailability with complete working code**

@workspace **need all functionality in existing workspace only**

1. Use **ChatGPT** prompts like:
   * “Generate a Mongoose schema for car booking with carId, userId, startDate, and endDate.”
   * “React component for selecting a car and making a Stripe payment.”
2. For logic-heavy parts (like fare calculation), ask AI to optimize or rewrite them using github copilot.

The prompt message for github copilot

“Here is the current fare calculation logic in my React/Node.js project. Please review and optimize it for better readability, performance, and scalability. If possible, refactor it to handle edge cases like surge pricing, night charges, or discounts more cleanly”

**✅ Outcome:**

* Faster development with fewer errors.
* Cleaner and scalable code using AI pair programming.

**Phase 3: Code Optimization & Documentation**

**🔧 Tools: GitHub Copilot and Chat GPT**

Ask **GitHub Copilot**:

* “What is the time complexity of my cab booking route in Express?”

Prompt message

“@workspace What is the time complexity of my cab booking route in Express”

* “Optimize this Mongoose find query.”

Prompt message

“@workspace Optimize this Mongoose find query.”

**Phase 4: AI-Powered Testing**

**🔧 Tools: TestRigor**

**Steps:**

1. **Connect your app URL** to TestRigor.
2. Use **natural language to define test cases**, e.g.:
   * “Login as user, book a cab, go to payment, and verify booking confirmation.”
3. Run **automated AI-generated test cases** across:
   * Registration
   * Booking
   * Admin car management
   * Stripe payment

**✅ Outcome:**

* Save time writing manual test cases.
* Catch regressions with automated AI QA.