Plus Points in Implementation (Overall Evaluation Criteria)

1. Authentication:
- Implement robust user authentication protocols to ensure secure access.

2. Cost Estimation - Time and Space:
- Conduct a thorough analysis of time and space complexity in the system.
- Utilize efficient algorithms and data structures to optimize both time and space requirements.

3. Handling System Failure Cases:
- Implement fault-tolerant mechanisms to address system failures.
- Employ backup and recovery strategies for data integrity.
- Develop comprehensive error recovery procedures to minimize downtime.

4. Object-Oriented Programming Language (OOPS):
- Choose a robust OOPS language for structured and modular code.
- Leverage OOPS principles such as encapsulation, inheritance, and polymorphism for maintainability and extensibility.

5. Trade-offs in the System:
- Clearly define and document trade-offs made during system design.
- Evaluate and communicate the rationale behind architectural and design decisions.
- Consider trade-offs in terms of performance, scalability, and maintainability.

6. System Monitoring:
- Implement comprehensive monitoring tools to track system performance.
- Utilize real-time dashboards and logging mechanisms to promptly identify and address issues.

7. Caching:
- Integrate caching mechanisms to enhance system response times.
- Utilize caching for frequently accessed data to reduce database load.
- Implement cache eviction policies for optimal resource utilization.

8. Error and Exception Handling:
- Develop a robust error and exception handling framework.
- Provide meaningful error messages for effective debugging.
- Regularly review and update error-handling strategies based on system usage patterns.

Instructions:

1. Read and Understand the Problem Statement:
- Carefully read the problem statement provided. Understand the requirements, inputs, expected outputs, and any constraints mentioned.

2. Choose a Programming Language:
- Select a programming language you are comfortable with and that is suitable for solving the problem described in the case study.

3. Design Your Solution:

- Plan the overall structure of your solution. Consider the algorithms, data structures, and any potential optimizations needed.

4. Write the Code:
- Implement your solution in code. Follow best practices for coding standards, such as meaningful variable names, proper indentation, and comments where necessary.
- Break down the problem into smaller functions or modules to improve code readability and maintainability.

5. Test Your Code:
- Test your code thoroughly with different sets of input data, including edge cases and boundary conditions.
- Ensure that your code produces the expected outputs for all test cases.

7. Document Your Code :
- Consider adding documentation or comments to explain the logic and purpose of your code, especially for complex parts or algorithms.

8. Submit Your Solution:
- Once you're satisfied with your code and it meets all the requirements, submit your solution on GitHub and share the GitHub link.

9. Demonstration:
- Include a demonstration video showcasing key features of the ride-sharing platform.
- Alternatively, use screenshots to visually highlight the user interface and functionality.

Vendor Cab and Driver Onboarding & Vendor Hierarchy Management:

A **Vendor Cab and Driver Onboarding System** enables seamless **multi-level vendor management, vehicle onboarding, document verification, and hierarchical access control**. This system ensures that **Super Vendors and Sub Vendors** can efficiently manage fleets, onboard drivers, and maintain compliance, while providing structured delegation capabilities.

I. Multi-Level Vendor Hierarchy

To support **large-scale fleet operations**, the system allows vendors to operate at **multiple hierarchical levels**, ensuring **structured access control and management.**

✅ **Flexible N-Level Hierarchy**

- Vendors can operate at **multiple levels**, forming a **parent-child relationship** (e.g., Super Vendor → Regional Vendor → City Vendor → Local Vendor).
- This hierarchy ensures that **fleet management responsibilities are distributed**, preventing operational bottlenecks.

✅ **Role-Based Access Management**

- Each vendor level has **specific roles and permissions**, ensuring **controlled access to vehicles, drivers, and fleet operations**.
- **Super Vendors** can **define access policies** for their sub-vendors.

🔹 **Example Use Case:**

- A **National Fleet Operator (Super Vendor)** has **Regional Vendors**, who in turn manage **City-Level Vendors** handling **local cabs and drivers**.

🔹 **Outcome:**

- Ensures **scalability** for large vendor networks.
- **Prevents unauthorized access** by implementing a structured hierarchy.

II. Super Vendor Access & Delegation

Super Vendors have complete control over **sub-vendors, fleets, and driver onboarding**, ensuring centralized management.

✅ **Access Delegation to Sub Vendors**

- **Super Vendors** can **grant specific access permissions** to sub-vendors to manage:
  - **Fleet onboarding & assignments**
  - **Driver onboarding & verification**
  - **Operational tasks (booking management, payments, compliance tracking, etc.)**

✅ **Delegation of Authority**

- A **Super Vendor** can **authorize a sub-vendor** to act on their behalf for certain operations.
- The **delegated sub-vendor** can perform **all or selected** administrative actions under the Super Vendor's name.

✅ **Controlled Delegation Rights**

- The **Super Vendor** can:
  - **Enable or revoke delegation at any time**.
  - Restrict access to **specific functions** (e.g., sub-vendor can onboard drivers but not process payments).

🔹 **Example Use Case:**

- A **Super Vendor managing a large fleet** can delegate responsibilities to a **Regional Vendor**, who further assigns tasks to **City-Level Vendors**.
- The Super Vendor ensures **smooth workflow automation** without handling **every small task manually**.

🔹 **Outcome:**

- **Reduces administrative overhead** for Super Vendors.
- **Ensures continuity** of operations even if the Super Vendor is unavailable.

III. Sub-Vendor Fleet & Driver Management

Sub Vendors are responsible for **managing vehicle onboarding, driver assignments, and compliance documentation.**

✅ **Vehicle & Driver Onboarding**

- Sub Vendors can onboard:
  - **Cabs & Commercial Vehicles** – Enter vehicle details such as **registration number, model, seating capacity, fuel type, etc.**
  - **Drivers** – Add drivers and assign them to specific vehicles.

✅ **Driver Document Upload**

- Sub Vendors can upload **essential driver documents**:
  - **Driving License (DL)**
  - **Vehicle Registration Certificate (RC)**
  - **Permit & Pollution Certificate**.

🔹 **Example Use Case:**

- A **City-Level Vendor** manages **local fleet onboarding and assigns drivers** to cabs.
- The system **flags expired documents** and prevents non-compliant vehicles from operating.

🔹 **Outcome:**

- **Ensures regulatory compliance** with up-to-date documentation.
- **Reduces manual document tracking efforts** with automated reminders.

IV. Super Vendor's Complete Control Over Sub-Vendors

Super Vendors have **full visibility and control** over their **entire sub-vendor network, including fleet status, driver assignments, and compliance reports**.

✅ **Centralized Dashboard for Super Vendors**

- A **single dashboard** provides:
  - **Real-time view** of all sub-vendors.
  - **Fleet status updates** (number of active/inactive vehicles).
  - **Pending document verifications & approvals**.
  - **Driver availability tracking**.

✅ **Super Vendor Action Control**

- The **Super Vendor can override sub-vendor actions**, ensuring compliance and smooth operations.
- **System-wide reports & analytics** help **optimize fleet performance**.

🔹 **Example Use Case:**

- A **Super Vendor detects compliance issues** (e.g., missing insurance documents) and **temporarily disables vehicle operations** until resolved.

🔹 **Outcome:**

- **Maintains operational consistency & regulatory compliance** across multiple vendor levels.
- **Gives Super Vendors full visibility and authority** over their entire fleet network.