Technical Constraints and Non-Functional Requirements for Rehnuma

Technical Constraints:

1. Using MongoDB, a NoSQL database, may not be ideal if complex transactions or strong consistency are required.
2. The project is constrained to React.js (frontend), Node.js (backend), and Python (optimization).
3. Relies on Google Maps API, meaning API limits, pricing, and possible latency must be managed.
4. It is designed for phones and computers so the UI/UX must be responsive across different screen sizes.
5. User authentication and protection of user profiles are necessary for user safety.
6. The system must handle multiple users and large datasets without excessive computational load.

Non-Functional Requirements:

1. The app should be smooth, meaning the optimized itinerary should be generated within 8-10 seconds of the user requesting it.
2. The system should be up and running at all times of the day.
3. The system should be able to handle the increasing number of customers and the data of the growing supermarkets without compromising on the smoothness.
4. The user data(profile and location details) should be protected.
5. The UI must be easy to use for users from different walks of life.
6. The code should be well documented so that it can be easily maintained.

7. Supermarket price and inventory data should be updated every 24 hours to give the users real-time data about the inventory of supermarkets.
8. Google Maps API must be monitored to prevent exceeding free-tier limits or incurring unexpected costs.
9. The system should log errors and provide users with meaningful error messages in case of failures.