Feasibility Report

Team Members

G.Pramodh -23MCCE26

D.Sai Kumar - 23MCCE24

Technical Feasibility

1.) Technology Availability

for **front-end** HTML5 , CSS3 , JS , React are useful , for **back-end** Nodejs and Expressjs are useful , for managing data MySql and PostgreSQL.

2.) Algorithm Implementation

for clustering K-means clustering and for optimised route we can use travelling salesman (TSP) , multiple travelling salesman (MTSP) , vehicle routing problem (VRP).

Economic Feasibility

we can use free servers like **Mysql** , **PostgreSq**l for database , for map API's **OpenStreetMap** + **Leaflet** they are completely free.

Operational Feasibility

1.) Operations

- Administrators can manage partners and monitor operations efficiently.
- Delivery partners require minimal technical training.
- Customers use a simple and user-friendly dashboard.

2.) Process Integration

The system integrates smoothly into daily operations, including:

- Subscription selection and automatic order generation
- Cut-off time enforcement
- Partner assignment and route optimization
- Delivery confirmation
- Salary calculation and reporting

3.) User Acceptance

- Customers gain flexibility and real-time tracking.
- Delivery partners receive structured routes and transparent earnings.
- Administrators achieve better operational visibility and control.

Schedule Feasibility

Phase	Duration
Requirement Analysis	3 days
System Design	3 days
Development	4 weeks
Testing	2 week
Deployment	1 Week

Estimated Total Duration: 9 - 10 Weeks

Conclusion

The proposed delivery management system is technically feasible using modern web technologies, optimisation algorithms, and open-source tools. It is economically viable due to the use of free databases and mapping services, minimizing development and operational costs. The system integrates smoothly into organizational workflows with minimal training and high user acceptance. With an estimated timeline of 9–10 weeks, the project is practical and recommended for implementation.

Functional Requirements

- Users can register, log in securely, and access dashboards based on their roles (Admin, Delivery Partner, or Customer).
- Customers can choose subscription plans (daily, monthly, or yearly), customise their fruit bowls, select predefined delivery time slots, and manage their subscriptions by pausing, resuming, or skipping deliveries before the cut-off time.
- Daily orders are automatically generated from active subscriptions and locked once the defined cut-off time is reached.
- Administrators can manage delivery partners by adding or removing accounts, activating or deactivating them, and setting partner availability for specific delivery slots.
- Customer orders are automatically assigned to delivery partners using clustering algorithms, ensuring that routes begin and end at the depot.

- Delivery routes are optimised using routing algorithms, and route durations are validated against time slot constraints before schedules are finalised.
- Administrators have the ability to manually review and override delivery assignments when necessary.
- Delivery partners can view their assigned routes, mark deliveries as completed, and record delivery timestamps.
- Earnings for delivery partners are calculated based on completed deliveries, and monthly salary reports are generated accordingly.
- Administrators have access to dashboards and reports to monitor deliveries, revenue, and partner performance.
- Customers can view their delivery status, subscription details, and payment history at any time.
- All user, subscription, order, and delivery data is securely stored and managed within the database.

Non - Functional Requirements

- Secure authentication and authorisation are enforced through encrypted credentials and role-based access control.
- Data integrity is maintained by preventing unauthorised modifications to orders after the defined cut-off time.
- The platform is designed to scale efficiently, handling a growing number of customers, delivery partners, and daily orders without significant performance issues.
- Delivery schedules and optimised routes are generated within an acceptable processing time to ensure operational efficiency.
- High availability and reliability are maintained, particularly during active delivery time slots.
- Consistent performance is ensured under normal operational loads.
- The interface is designed to be user-friendly and intuitive for administrators, delivery partners, and customers.
- User, subscription, and payment information are securely stored to ensure data confidentiality.
- Fault tolerance is supported through system error logging and the maintenance of detailed delivery execution records.
- Accurate delivery logs and salary calculations are maintained to support auditing and reporting requirements.
- The architecture is modular and maintainable, allowing future enhancements such as real-time traffic integration or delivery capacity constraints.
- Compatibility is ensured across modern web browsers, with a responsive design that adapts to different device screen sizes.