Database Management Submission 1

Aditya Gupta(2022031) , Chandan Sah(2022140),
Dhruv Sharma(2022170) ,Tanmay Singhal(2022535)

(Group - 5)

January 21, 2024

1 Project Scope

Our online retailing software is a tool that facilitates ordering a variety of products used in our everyday lives by partnering with local grocery shops in the city and sending its delivery drivers to collect the articles ordered by the customers from these stores to be delivered to the customer within 15 minutes. It provides convenience for customers by eliminating unnecessary travel time. Customers can place and receive their orders in the comfort of their homes instead of often traveling to the store only to find out the product they choose is unavailable. It enables local retail stores to expand their customer base, manage their resources more efficiently, provide quicker services, and give them a competitive advantage by coming across as forward-thinking businesses and catching the eye of potential investors. Finally, it increases safety and accountability by allowing access to the real-time location of the delivery drivers' verified drivers, encouraging delivery drivers and customers to rate each other and including a helpline, chat support, or other means of communication for immediate assistance.

We plan on storing the following data:

1.1 Customer Information:

- Personal details such as name, contact details, and delivery address.
- Their mode of payment for the particular delivery.
- The ratings given to them by past delivery drivers (if any).

1.2 Delivery Partner Information:

- Personal details such as name and phone number.
- Details of their vehicle: type, model, color, license plate number.
- Their accumulated rating.

1.3 Trip Information:

- Pick-up and drop-off location. (along with a visual map, not necessarily real-time)
- Trip distance and duration.
- Rating and feedback from customers.

- Payment details and total bill.
- Trip status.
- Any notes provided by the customer.

1.4 Vendor's Information:

- Personal details such as name, contact details, shop address , etc.
- Total transactions done by that Vendor , the feedback of the Vendor , Current rating of the vendor , etc .

2 Functional Requirements

2.1 User Authentication and Authorization

- Users should be able to create an account.
- Users must be able to log in securely.
- Different user roles (e.g., customer, admin) with appropriate access levels.

2.2 Product Catalog

- Display a categorized list of products.
- Allow users to search and filter products.
- Each product should have details such as name, description, price, and availability.

2.3 Shopping Cart

- Users should be able to add/remove items to/from their shopping cart.
- Display the total price in the shopping cart.
- Users can proceed to checkout from the shopping cart.

2.4 Order Management

- Users should be able to view their order history.
- Admins should be able to manage and process orders.

2.5 Delivery Partner Information

- Admin will be able to see the names and vehicle details of the delivery partner.
- Admin can also view the real-time location of the delivery partners.

2.6 User Profile Management

- Users should be able to edit their profile information, including name, contact details, and delivery address.
- Provide an option for users to reset their passwords securely.
- Users can upload and update their profile pictures.

2.7 Product Reviews and Ratings

- Users should have the ability to leave reviews and ratings for products they have purchased.
- Display average ratings for each product based on customer feedback.
- Allow users to view and filter product reviews.

2.8 Promotions and Discounts

- Implement a promotional system, allowing admins to create and manage discounts or coupon codes.
- Users can apply promo codes during the checkout process to avail discounts.

2.9 Real-time Order Tracking

- Provide a real-time tracking feature for users to monitor the status and location of their delivery.
- Send push notifications or emails to users with updates on their order status.

2.10 Inventory Management

- Admins should be able to update product availability and stock levels in real-time.
- Receive notifications for low-stock products and generate automated reorder suggestions.

3 Non-functional Requirements

3.1 Performance

- The app should handle a minimum of 1000 simultaneous users.
- Page load times should be within 3 seconds.

3.2 Security

- User data should be stored securely.
- Transactions must be encrypted using industry-standard protocols

3.3 Scalability

- The system should be scalable to accommodate future growth.

4 Project Constraints

4.1 Responsiveness and Compatibility:

- The user interface must be responsive and adapt to various device screens, including desktops, tablets, and smartphones.
- The application must be compatible with major web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

4.2 Third-party Integrations:

- Integration with external services or APIs should be carefully managed to ensure compatibility and reliability.
- Regular checks and updates should be performed to accommodate changes in third-party services.

5 Technical Requirements

5.1 Front-end

The front-end development for our app will be implemented using HTML5, CSS3, and JavaScript to create a responsive and interactive user interface. Bootstrap will be leveraged to streamline design consistency and responsiveness. The development team will prioritize cross-browser compatibility, optimize performance for faster loading times, and adhere to accessibility standards. Security best practices will be employed, and performance monitoring tools will be integrated to identify and address issues promptly. The front-end architecture will be designed for scalability, allowing for future enhancements and an expanding user base. Through these technical requirements, the team aims to deliver a visually appealing, accessible, and high-performing front-end for the app.

5.2 Back-end

The back-end development for our app will be powered by Python, with the choice of either the Django or Flask framework. The need for a robust and scalable server-side architecture drives this decision. Django offers a comprehensive, high-level framework suitable for complex applications, providing built-in features like an ORM, authentication, and admin interfaces. On the other hand, being a micro-framework, Flask offers a more lightweight approach, providing flexibility and allowing developers to choose components based on specific project requirements. The back-end will handle data processing, business logic, and interaction with the database, ensuring seamless communication with the front-end components. The development team aims to create a stable and efficient back-end foundation for our app by adopting Python and a well-established web framework.

5.3 Database

The database management system chosen for our app is MySQL, a widely used relational database management system (RDBMS) known for its reliability, scalability, and strong community support. MySQL will be the backend datastore, efficiently managing and storing the app's relational data. Leveraging the SQL capabilities of MySQL, the back-end will interact with the database to execute queries, retrieve and store information, and ensure data integrity.

5.4 Version Control

Version control using Git, comprehensive documentation, and rigorous testing, including continuous integration, will ensure code quality and maintainability.

6 Contribution

1. **Aditya Gupta(2022031)** - Making of the latex document , organizing and reviewing the information . Wrote the content of the technical requirements and idea behind the technology which will be used in the project . Contributed in the technical requirements .

2. **Chandan Sah(2022140)** - Wrote the project constraints and contributed in the functional and non-functional requirements . Helped in documentation .

3. **Dhruv Sharma(2022170)** - Wrote the project scope with the contribution of the other members , reviewed the latex document .

4. **Tanmay Singhal(2022535)** - Wrote the functional and non-functional requirements , helped in modifying the technical requirements .

Latex Link
