Problem Statement: Design a College-Centric Grocery Delivery System

Objective:

The objective is to develop a Web2-based grocery delivery system tailored to the specific needs of college students, fostering convenience, affordability, and efficiency. This system aims to connect Grocery Sellers (ex- Radhika Supermarket, Old SC etc.) with students, providing a centralized platform for ordering groceries and essential supplies. The system will streamline the shopping experience, reduce time spent on errands, and ensure timely delivery, ultimately enhancing the overall student experience.

Key Features:

Seller Portal:

- **Store Registration:** Sellers can register their stores with essential details (name, address, contact information, operating hours).
- **Product Listing & Management:** Sellers can upload product catalogs, including descriptions, images, and prices. They can also update inventory levels in real-time.
- **Order Management:** Sellers receive instant notifications upon receiving new orders, view order details, and manage order status.
- **Transaction History:** Sellers can access a comprehensive history of past orders and transactions, including payment details and customer information.
- **Max Purchases:** There should also be an option for the sellers to view the student with max shopping (probably a list) in order to give discount coupons, or something of that sort.
- **Promotional Tools:** Sellers can create and manage promotions, discounts, and special offers to attract student customers.
- **Inventory Update:** Sellers can easily update inventory levels.

Student Portal:

- **User Registration & Login:** Students can register using their college ID and credentials, ensuring exclusivity and security.
- **Location Services:** Students can set their delivery location manually or utilize location APIs for automatic detection.
- **Product Browsing & Search:** Students can browse products by category, search for specific items, and view product details, including descriptions, images, and prices.

- **Cart Management & Checkout:** Students can add items to their cart, review their order, and proceed to checkout with secure payment options.
- **Order Tracking:** Students can track the real-time status of their orders, from order confirmation to delivery.
- **Order History:** Students can access a detailed history of their past orders and transactions.
- **Notifications:** Push notifications for order confirmation, delivery updates, and promotional offers.
- **Payment Integration:** Integration of Razorpay (dummy account just for demonstration) or multiple payment gateways (UPI, credit/debit cards) via any other source.
- **Delivery Slot Selection:** Students can choose desired delivery slots.

Bonus Features:

- Scheduled Orders: Students can schedule recurring grocery orders for regular delivery.
- "Essentials" List: Curated lists of commonly purchased items for quick ordering.
- College-Specific Promotions: Exclusive discounts and promotions for college students.
- Chat Feature: A direct chat option between student and seller or delivery personnel.
- AI-powered product recommendations based on student purchase history.
- Implementing a loyalty program.
- Any other feature that you feel missing on either seller or student side.

Submission Requirements:

Participants are required to submit the following:

- **Deployment Link:** A live URL to the deployed web application. This allows evaluators to directly interact with the system and assess its functionality.
- **Working Video Demonstration:** A video (maximum – 3 minutes) showcasing the key features and functionalities of the grocery delivery system. The video should provide a clear walkthrough of the user interface and demonstrate how the system addresses the defined problem statement.
- **Presentation slides (max 5 slides)** explaining the idea, tech stack, and implementation.
- **GitHub Repository Link:** A link to the GitHub repository containing the complete source code of the project. The repository should be well-organized, with clear documentation and comments to facilitate code review.
- **Documentation:** a document explaining the architechture, any third party api implementation, and also the steps to run the project locally.