Module Breakdown Member Responsibilities Module

1: Online Store

Objective: Provide a responsive e-commerce website for customers to browse products, add items to a cart, check out securely, and read/write reviews.

- **Anjana:**
    - **Front-end:** Develop the product catalog pages (product listing, details, images, specifications).
    - **Back-end:** Implement APIs for fetching product data from the database.
- **Sanidu:**
    - **Front-end:** Build the shopping cart interface and checkout pages.
    - **Back-end:** Develop secure checkout APIs, including integration with a payment gateway (e.g., Stripe, PayPal).
- **Ishan:**
    - **Front-end:** Implement advanced filtering and search functionality on the product listing pages.
    - **Back-end:** Create APIs for product search, filtering, and applying promotions/discounts.
- **Avishka:**
    - **Front-end:** Develop the customer reviews and ratings components with a responsive design.
    - **Back-end:** Build the API endpoints to handle review submissions, retrieval, and moderation.

Module 2: Inventory Management

Objective: Track stock levels, trigger low-stock alerts, support barcode/QR code integration, and provide inventory dashboards.

- **Anjana:**
    - **Back-end:** Design and implement the core inventory tracking API (CRUD operations for products, stock updates).
    - **Front-end:** Integrate with the product catalog pages to display current stock information.
- **Sanidu:**

    - **Back-end:** Develop low-stock alert logic and notifications API.
    - **Front-end:** Create dashboard components for real-time stock alerts and inventory summaries.
- **Ishan:**
    - **Full-stack:** Implement barcode/QR code integration (back-end API to process scanned codes and front-end components to trigger scans).
- **Avishka:**
    - **Back-end:** Build modules to track historical inventory data and trends.
    - **Front-end:** Develop data visualization components (charts/graphs) for inventory analytics.

Module 3: Customer Relationship Management (CRM)

Objective: Manage customer profiles, loyalty programs, communication (email/SMS notifications), and feedback mechanisms.

- **Avishka:**
    - **Back-end:** Develop APIs for managing customer profiles (registration, login, profile updates).
    - **Front-end:** Build user interfaces for profile management.
    - **Back-end:** Implement the loyalty program logic and rewards system API.
    - **Front-end:** Create components to display loyalty status and reward points to customers.
    - **Back-end:** Integrate email/SMS notification services to send order confirmations and promotional messages.
    - **Front-end:** Develop a notification center/dashboard for customers to view messages.
    - **Back-end:** Create APIs for collecting and managing customer feedback and support tickets.
    - **Front-end:** Implement feedback submission forms and display customer reviews on the Store.

Module 4: Repair s Service Tracking

Objective: Enable customers to submit repair/service requests and track the status of their service jobs.

- **Sanidu:**
  - Develop the repair/service request submission form.
  - Create API endpoints for submitting and storing repair requests.
  - Implement logic for tracking repair job statuses (e.g., Pending, In Progress, Completed).
  - Build a status tracking component so customers can monitor the progress of their repair requests.
  - Integrate a notification system to alert customers on status changes.
  - Add real-time notification displays (pop-ups, dashboard alerts) regarding service updates.
  - Set up APIs to archive service history and provide detailed logs for each repair.
  - Create historical service record views for both the customer and administrative interfaces.

Module 5: Order s Delivery Management

Objective: Handle order processing, delivery tracking, and manage returns/refunds.

- **Anjana:**
  - Develop APIs for processing new orders and updating order status.
  - Build order summary pages for customers (order confirmation, details, history).
  - Create logic for shipping integration (generating shipping labels, tracking shipments) and APIs for delivery status updates.

- Develop order tracking components for customers and admin dashboards.
- Implement APIs for handling return and refund requests according to business policies.
- Design user interfaces to initiate and track returns/refunds.
- Assist with integrating order C delivery functionalities, ensuring data consistency across the Store and admin panels.
- Provide additional validations, logging, and error handling for the entire order lifecycle.

Module 6: Data Analytics s Reporting

Objective: Provide insights through real-time dashboards, sales/inventory reports, and predictive analytics tools.

- **Ishan:**
  - Develop APIs to aggregate and provide sales data (daily, weekly, monthly reports).
  - Create dashboard components displaying key performance indicators (KPIs).
  - Implement logic to track and analyze inventory trends over time.
  - Develop visualization components (charts, graphs) for inventory analytics.
  - Build APIs to analyze customer behavior and generate detailed reports.
  - Create user interfaces for exporting and viewing customer analytics (CSV/Excel export options).
  - Integrate predictive analytics features (e.g., forecasting future demand based on historical data).
  - Enhance the reporting module with interactive elements for deeper data exploration.

Module 7: Administration s Integration

Objective: Serve as the central management console, handling system configuration, security, and third-party integrations.

- **Anjana:**
  - **Back-end:** Implement authentication and role-based access control using Spring Security with JWT.
  - **Front-end:** Build the login/authorization UI and basic admin dashboard layout.
- **Sanidu:**
  - **Back-end:** Develop APIs for payment gateway integration and related financial configurations.
  - **Front-end:** Create administrative components for managing payment options and viewing transaction summaries.
- **Ishan:**
  - **Back-end:** Set up APIs for integrating third-party services (e.g., shipping, notifications, data backups).
  - **Front-end:** Develop admin panels to configure these integrations and monitor their status.
- **Avishka:**
  - **Back-end:** Build APIs for system configuration settings (e.g., automated notifications, system backups) and ensure proper logging and error handling.
  - **Front-end:** Enhance the admin dashboard with tools for system monitoring, configuration, and overall data management.

Final Notes on Collaboration

- **API Contracts s Documentation:**
  Every member is responsible for documenting the APIs they create using Swagger/OpenAPI. This ensures seamless integration between front-end and back-end across all modules.
- **Shared Codebase:**
  Use a version control system (like Git) with clearly defined branches and regular pull requests to ensure each member's work integrates smoothly.
- **Regular Stand-ups s Syncs:**
  Schedule frequent meetings to discuss integration points, test functionalities across modules, and resolve any dependency conflicts.

- **CI/CD s Testing:**
  Each member is expected to write unit tests (using JUnit for Spring Boot and Jest/React Testing Library for React components) and participate in integration testing. Utilize a CI/CD pipeline (e.g., GitHub Actions, GitLab CI) to automate build, testing, and deployment processes.