

Module Breakdown & Member Responsibilities

Module 1: Online Storefront

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

- **Member A:**
 - **Front-end:** Develop the product catalog pages (product listing, details, images, specifications).
 - **Back-end:** Implement APIs for fetching product data from the database.
 - **Member B:**
 - **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).
 - **Member C:**
 - **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.
 - **Member D:**
 - **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.

- **Member A:**
 - **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.
- **Member B:**

- **Back-end:** Develop low-stock alert logic and notifications API.
 - **Front-end:** Create dashboard components for real-time stock alerts and inventory summaries.
 - **Member C:**
 - **Full-stack:** Implement barcode/QR code integration (back-end API to process scanned codes and front-end components to trigger scans).
 - **Member D:**
 - **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.

- **Member A:**
 - **Back-end:** Develop APIs for managing customer profiles (registration, login, profile updates).
 - **Front-end:** Build user interfaces for profile management.
- **Member B:**
 - **Back-end:** Implement the loyalty program logic and rewards system API.
 - **Front-end:** Create components to display loyalty status and reward points to customers.
- **Member C:**
 - **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.
- **Member D:**
 - **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 storefront.

Module 4: Repair & Service Tracking

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

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

Module 5: Order & Delivery Management

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

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

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

Module 6: Data Analytics & Reporting

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

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

Module 7: Administration & Integration

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

- **Member A:**
 - **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.
- **Member B:**
 - **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.
- **Member C:**
 - **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.
- **Member D:**
 - **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 & 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 & Syncs:**

Schedule frequent meetings to discuss integration points, test functionalities across modules, and resolve any dependency conflicts.

- **CI/CD & 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.