Backend Development Task for Interns

Objective:

Develop the **desktop and responsive (mobile view)** functionality for the eCommerce web design template provided in the Figma file: [Ecommerce Web Design.](#) This task focuses on backend development while integrating the design template with dynamic and responsive capabilities.

Tasks Overview:

Each week is structured to guide you through backend development, integrating frontend designs with dynamic content, and ensuring a fully responsive layout.

Week 1: Project Setup and Static Backend Integration

1. **Goal**: Set up the project environment and create static endpoints to serve the frontend templates.
2. **Tasks**:
   - Set up the backend server using **Node.js** and **Express.js**.
   - ○ Organize the project directory with folders for routes, views, public (CSS/JS), and assets.
   - ○ Create static HTML/CSS files for:
     - ■ Home Page
     - ■ Product Listing Page
     - ■ Product Details Page
   - ○ Serve these static pages through Express routes (/, /products, /products/:id).
   - ○ Ensure mobile responsiveness of all pages using media queries in the provided CSS files.
3. **Deliverables**:
   - Functional Node.js server with routes serving static pages.
   - ○ Desktop and mobile view consistency using responsive design techniques.
   - ○ Push code to a GitHub repository.

Week 2: Database Integration and Dynamic Content

1. **Goal**: Connect to a database and render dynamic content on the frontend.
2. **Tasks**:

> Set up a **MongoDB database** (or Firebase Realtime Database if preferred).

- ○ Create a collection for **Products** with attributes like id, name, price, category, image, description, and stock.
- ○ Populate the database with sample product data.
- ○ Use **EJS** (or another templating engine) to dynamically render:
  - ■ Home Page: Display featured products from the database.
  - ■ Product Listing Page: Fetch and display all products from the database in a grid layout.
  - ■ Product Details Page: Fetch and display detailed information for a single product.
- ○ Implement a **search bar** (functional, server-side) to filter products by name or category.

3. **Deliverables**:

> Fully integrated backend with dynamic data rendering.

- ○ Responsive design for all pages (desktop and mobile).
- ○ Code committed to GitHub repository.

Week 3: Backend Features and Final Integration

1. **Goal**: Add backend features and finalize the project for deployment.
2. **Tasks**:

> Implement **user authentication** using **JWT** or **Firebase Auth**.
> - ■ Create login and signup pages.
> - ■ Restrict access to certain pages (e.g., adding products, admin actions).

- ○ Add a feature to **add products** using a form and save them to the database. ○ Implement **pagination** for the Product Listing Page to handle large datasets. ○ Deploy the project to a platform like **Render**, **Heroku**, or **Vercel**.

3. **Deliverables**:

> Authentication system implemented and tested.

- ○ Pagination and add product functionality working as expected.
- ○ Deployed project with live URL.
- ○ Final code uploaded to the GitHub repository.

Tools and Technologies:

- **Node.js** and **Express.js** for backend development.
- **MongoDB** (or Firebase) for the database.
- **EJS** (or any templating engine) for dynamic content rendering.
- **HTML5/CSS3/JavaScript** for frontend integration.
  - **GitHub** for version control and code submission.
- OR YOU CAN USE TECHNOLOGY STACK OF YOUR OWN CHOICE

Submission Instructions:

1. Create a GitHub repository named ecommerce-backend-design.
2. Push your code weekly based on deliverables.
3. Share the repository link and deployed URL (Week 3) with your for review.

Evaluation Criteria:

1. **Functionality**:
   All routes are functional and serve dynamic data.
   ○ Responsive design on both desktop and mobile.
2. **Code Quality**:
   Organized folder structure and readable code.
   ○ Proper error handling in backend routes.
3. **Database Design**:
   Efficient schema for managing products.
   ○ Secure and functional authentication.
4. **Timely Submission**:
   Complete weekly tasks on time and push to GitHub.

Deadline

- **Final Submission Date: 14th January 2026**
- **You will have to submit all of these tasks collectively on 14th January 2026**

Good luck! Ensure the integration is seamless and both desktop and mobile views are responsive.