"SMILE" SKIN CARE

Objective:

Build a responsive e-commerce platform for skincare and wellness, focusing on a seamless shopping experience and educational content (blogs).

XX_Module_A: Design & Branding

Before coding, students must define the visual identity.

- **Logo Creation:** Design a logo that reflects "Wellness" and "Clean Beauty."
- **Style Guide:** Create a document specifying:
    - **Typography:** (e.g., Serif for headings, Sans-serif for body).
    - **Color Palette:** Use calming tones (Pastles,Nudes, Sage Greens, or Soft Pinks) inspired by Nykaa/Tira.
    - **Iconography:** Consistent set for "Cart," "Profile," and "Search."
- **UI/UX Design:** Deliver high-fidelity wireframes in **Figma** or Adobe XD.
    - **Mobile-First Approach:** Design for 390x844px before scaling to Desktop.
    - **Required Screens:** Home (Product Feed), Product Detail, Blog List, Cart, and Order History.

XX_Module_B: Backend API Development

Provide a Postman collection or Swagger documentation.

Function	Method	Description
Auth	POST /auth/login	Support Email/Password or Mobile/OTP logic.
Products	GET /products	Fetch all products with category filters.
Cart	POST /cart/add	Add product ID and quantity to user's session/DB.
Blogs	GET /blogs	Retrieve list of articles and content.
Orders	POST /checkout	Move items from Cart to Order History.
History	GET /orders	View past purchases for the logged-in user.

XX_Module_C: Frontend Implementation

The visual bridge between the user and the data.

- **Tech Stack:** Recommended React.js, Next.js, or Vue.js.
- **Key Deliverables:**
    - **Home Screen:** Dynamic grid displaying product cards (Image, Price, "Add to Cart" button).
    - **Responsive Navigation:** A sticky header with categories and search, following the Tira/Nykaa layout.
    - **State Management:** Use Redux, Context API, or Zustand to handle the "Cart" state across the site.
    - **Blog Integration:** A clean, readable layout for long-form text and images.

Database

Create following database tables or structure –

1. User
    a. Name
    b. Email
    c. Paasowrd/otp
    d. Phone Number
    e. Addresses (multiple)
2. Order
    a. Id
    b. Product_id
    c. Time of ordering.
    d. Amount paid
3. Cart (can be filled in temporarily)
    a. Id
    b. Product_id
    c. Price

4. Blogs
    a. Title
    b. Subtitle
    c. Table of Contents
    d. Headings of TOC
    e. Description of TOC
    f. SEO meta fields(multiple)
    g. Product_categories (foreign_key)
5. Product Categories
    a. Id
    b. Name
    c. Description
6. Products
    a. Id
    b. Name
    c. Description
    d. Expiry date
    e. Price
    f. Allergens
    g. image

Submission

1. Submit your database export with data by name XX_Module_B/smile_db.sql (or any other format If you are using any other format)
2. Submit postman collection or swagger documentation inside XX_Module_B folder
3. Host this website on a remote Url
4. Share atleast one git commit message for moule B and C in respective folders, inside read.me file.
5. Push code to github repository in the above-mentioned folder structure.

Marking Summary

1. **Responsiveness:** Does the site look as good on a phone as it does on a laptop?
2. **API Integration:** Does the frontend correctly talk to the backend, or is the data hardcoded?
3. **Visual Consistency:** Does the final site match the Figma Style Guide?
4. **Submission:** Are all the modules Submitted
5. **Authentication:** Is User authentication working fine.
6. **Cart:** Can Items be added and removed to cart
7. **Product Listing:** Can products be seen, on home screen with all details
8. **Blogs:** Can user navigate to blogs and see blogs listing and read them
9. **Order History:** Can user navigate to Order history and see their orders.
10. **Page refresh:** Does data persist on page refresh.