DAY 2 ACTIVITIES: TRANSITIONING TO TECHNICAL PLANNING

TECHNICAL REQUIREMENTS

Frontend Features

Frameworks: Use Next.js for building dynamic and high-performing user interfaces.

Styling:

Use Tailwind CSS or CSS for responsive designs..

State Management: Integrate for managing application states like cart, user preferences, and product filters.

APIs: Utilize APIs for fetching product details, reviews, and user profiles.

Responsive Design: Ensure mobile-first design using media queries or CSS frameworks.

Customization

Product Recommendations: Provide tailored product suggestions based on user preferences, skin types, and concerns entered during onboarding or profile setup.

Consultations

Virtual Try-On:
Use augmented reality (e.g., ModiFace) to let users virtually apply lipsticks, eyeshadows, or foundations.

Ingredient Transparency:
Clearly list product ingredients and explain their benefits.
Highlight allergens and certifications (e.g., vegan, cruelty-free).

Community Reviews:
Enable users to review and upload photos or videos of them using the products.

Subscription Services:
Offer consultations for building personalized skincare or makeup routines, bundled with product subscriptions.

Backend Features

Store and manage data for:

Products (details, tags, categories).

Users (profiles, preferences, and consultation history).

Orders (order history, statuses).

APIs .

Product API:

GET /products: Retrieve all cosmetics products.

GET /products/:id: Retrieve a specific product by ID.

POST /products: Create a new cosmetic product.

PUT /products/:id: Update a cosmetic product.

DELETE /products/:id: Delete a product.

User API:

GET /users: Retrieve a list of users.

GET /users/:id: Retrieve specific user details.

POST /users/register: Register a new user.

POST /users/login: Authenticate a user.

Order API:

GET /orders: Retrieve all orders.

GET /orders/:id: Retrieve a specific order.

POST /orders: Create a new order.

Sanity Integration:

Sanity API: Fetch and manage cosmetic product data stored in Sanity’s headless CMS.

Product Content: Sanity will store the cosmetics product details (images, descriptions, ingredients, pricing, etc.).

Admin Panel

Use a **Headless CMS** (e.g., **Sanity CMS**) for:

Managing product catalog, including descriptions, tags, and images.

Adding or editing eco-friendly tags for sustainable products.

Order Tracking: View and update the status of customer orders.

Customer Data Management: Access user profiles, consultation history, and order data.

Loyalty Program Configuration: Set rules and manage reward systems for customer retention.

1.Structure the components, routes, UI, constants, and schema types for a cosmetics product site:

Components

1. Navbar
2. Footer
3. Product Card
4. Product Details Page
5. Review Section
6. Product Search Bar
7. Blog Section
8. Subscription Form
9. Social Media Links
10. Product Filters (Category, Price, Brand)
11. Sale Banner

Routes

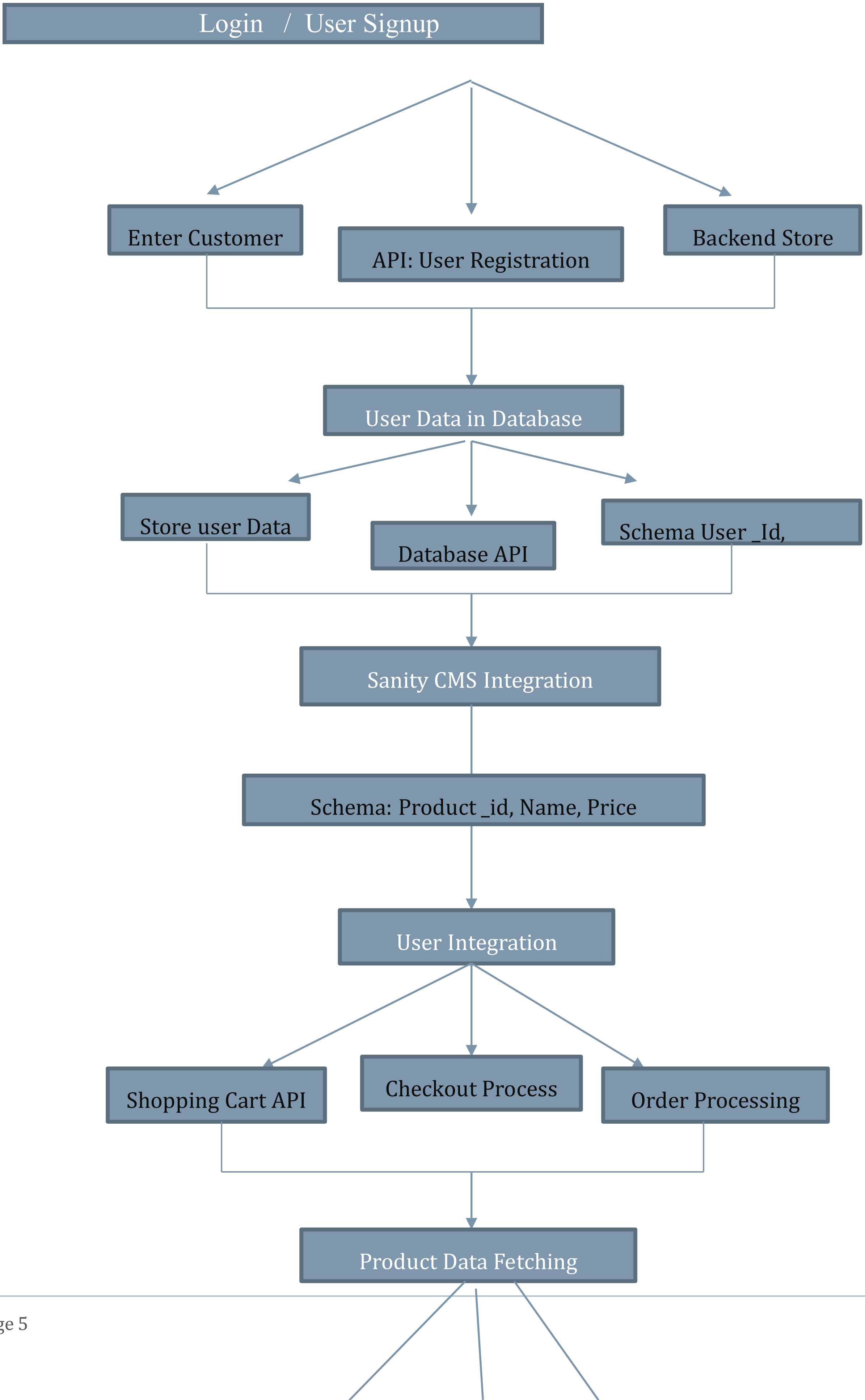
1. Blog
2. Cart
3. Checkout
4. Contact
5. Shop
6. Hero
7. Featured Products 8. Blog Center
8. Customer Reviews
9. About Us
10. Beauty Tips

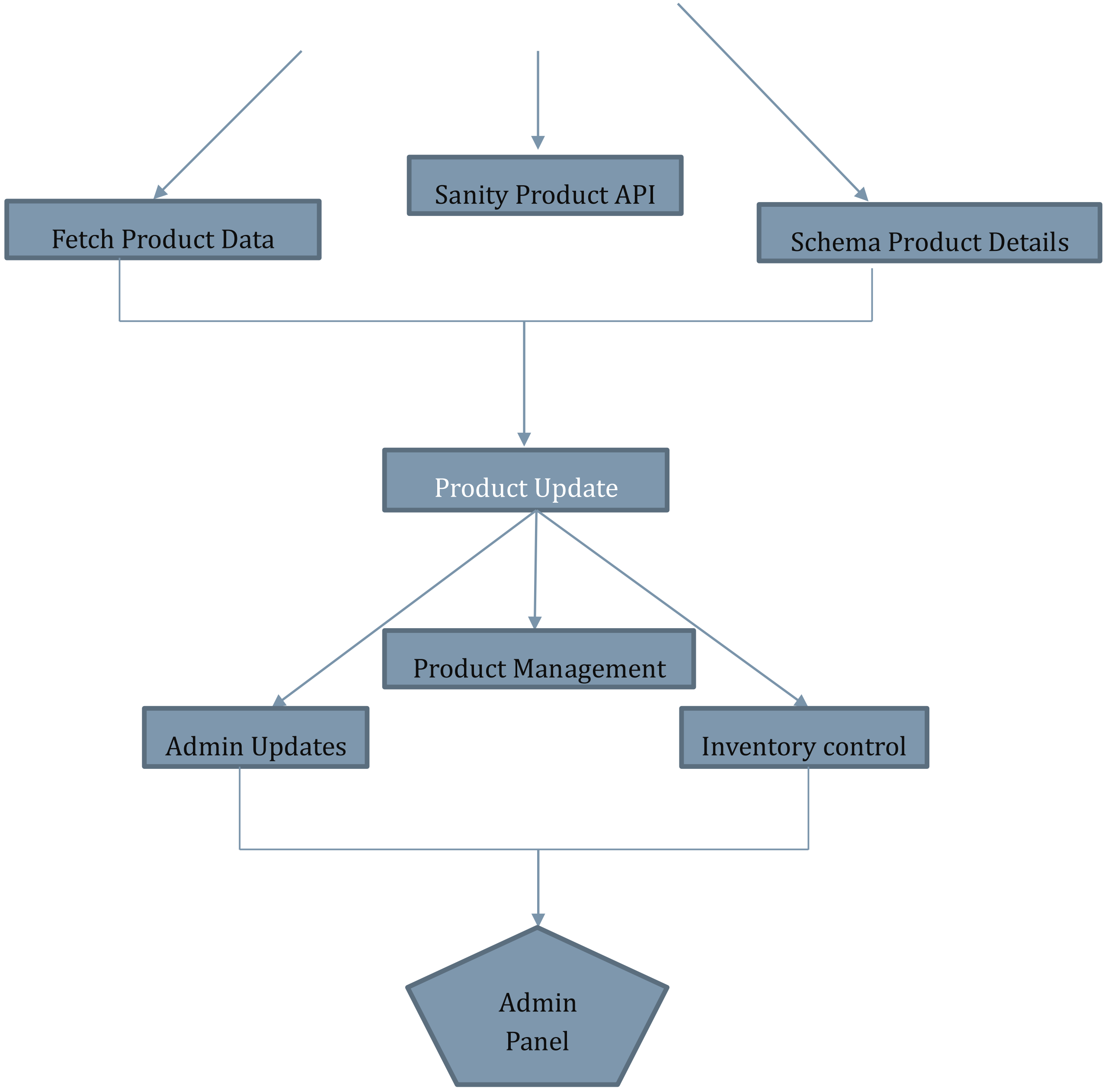
UI Elements

1. Badge
2. Button
3. Card
4. Input Field
5. Label
6. Add Product Button
7. Product Sorting (by price, rating)
8. Rating Stars
9. Cart Totals
10. Product Image Gallery
11. Product Quantity Selector

Constants

1. Radio-Group (Skin Type, Preference, etc.)
2. Select (Brand, Category)
3. Separator
4. Tabs (for product details, ingredients, reviews)
5. Textarea (for reviews)
6. Icon (for different cosmetic product types)
7. Product Discounts
8. Price Formatting


Login / User Signup
Enter Customer
API: User Registration
Backend Store
User Data in Database
Store user Data
Database API
Schema User _Id,
Sanity CMS Integration
Schema: Product _id, Name, Price
User Integration
Shopping Cart API
Checkout Process
Order Processing
Product Data Fetching


Sanity Product API
Fetch Product Data
Schema Product Details
Product Update
Product Management
Admin Updates
Inventory control
Admin
Panel