Marketplace Technical Foundation - Hekto

Day 2: Planning the Technical Foundation

1. Define Technical Requirements

Frontend Requirements:

- User-friendly interface for browsing products:
  - Add titles and subtitles to help users understand what they are looking for (already done in the UI/UX template).
  - Breadcrumbs already added.
  - Add a back button in the breadcrumb (not yet implemented, needs work).
  - Add "Back to Home" functionality on the logo.
  - Add movement effects using shadows and transform scale on buttons.
  - After adding dynamic products, use a carousel from ShadCN:
    - For mobile and tablet sizes, make the first product slightly move on the x-axis to grab user attention. Otherwise, users might think there is only a single product and swipe up/down.
- Responsive design for mobile and desktop users:
  - Use flex and grid layouts extensively.
  - Hide certain elements for smaller screens.
  - Add a hamburger menu. (already done in the UI/UX template)
- Essential pages:
  - Home : Attract customers with featured products and categories.
  - Product Listing : Let users browse products with filters and pagination.
  - Product Details : Provide detailed product information.
    Features: Images, description, price, add-to-cart button.
  - Cart and Checkout : Ensure a smooth purchase process.
    Features: View items in cart, payment, and order confirmation.
  - Order History : Enable users to view past orders.
    Features: List of orders, status tracking.

Static vs Dynamic Pages:

Page Name	Static	Dynamic
Home	✓	Needs work
Listing Product	✓	Needs work
Product Details	✗	Needs work
Cart	✓	Needs work
Checkout	✓	Needs work
order	✓	Needs work
order complete	✓	No dynamic additions needed

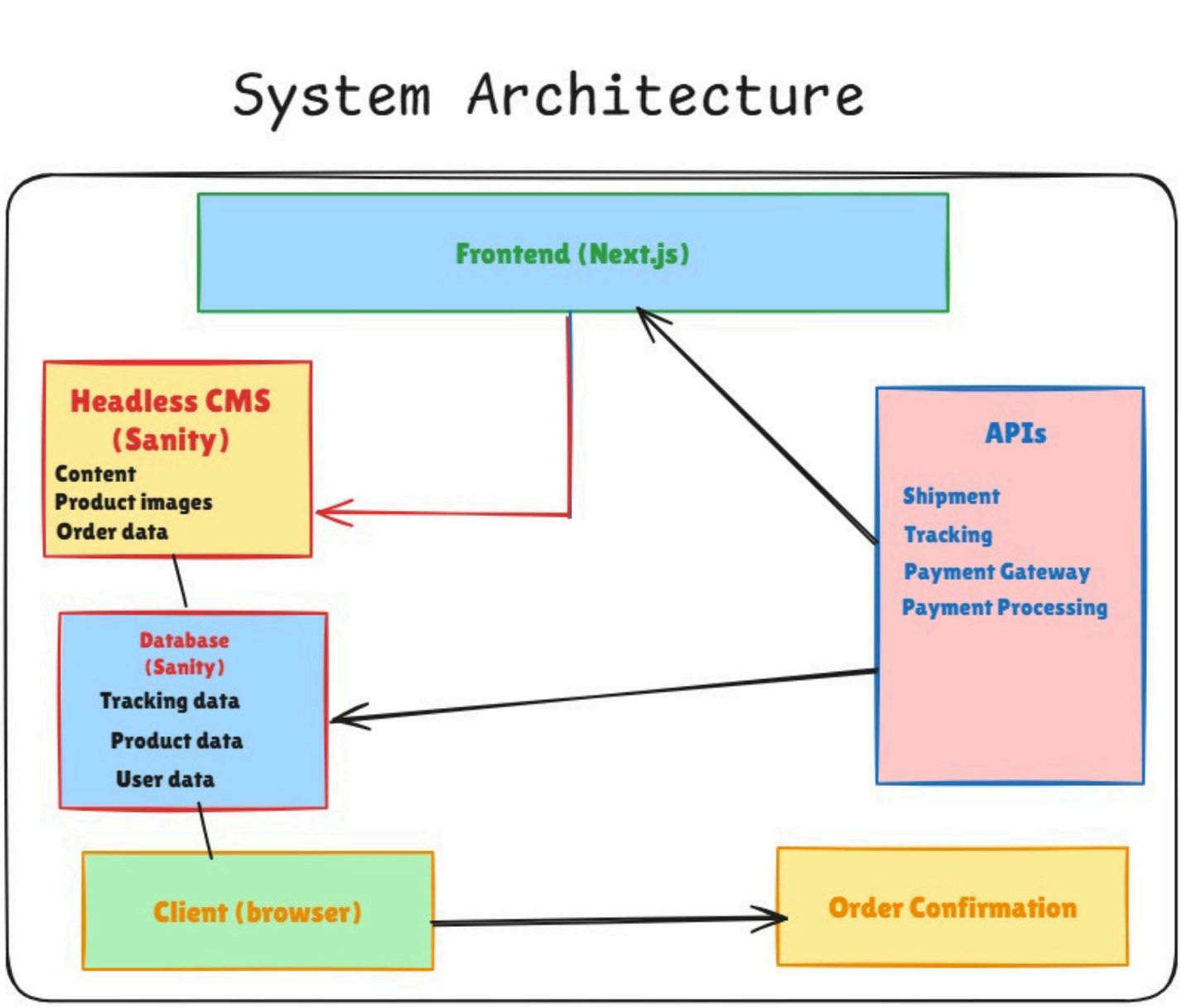
Sanity CMS as Backend:

- Sanity CMS Schemas:
  - Product Schema (High Priority): Fields: Name, description, price, category, stock, images.
  - Order Schema (High Priority): Fields: User details, products, total price, status.
  - User Schema (Medium Priority): Fields: Name, email, phone, address.
  - Review Schema (Low Priority): Fields: Product ID, user ID, rating, comments.
- Use Sanity CMS to manage product data, customer details, and orders:
  - Product data: Needs work on it. We've done it in class projects, so not a hard task. Customer
  - and order details: Need to research how to implement and then implement it.
- Focus on designing schemas in Sanity to align with the business goals from Day 1:
  - Schema will be provided by the teacher on Day 3, but as a backup, I think my schema is totally the same as what I created for my Thursday class project (9 to 12). Just need to add some tags like "Featured," "Latest Products," "Trending," etc., and some extra details.
- For the day one`s goal to make the process transparent, we can have input fields like images/videos, a chat section, and more. Will refine it during creation.

Third-Party APIs:

- ==> Integrate APIs for shipment tracking, payment gateways, and backend services.
  - Need to research and develop this since it's something I have never done before.. Ensure APIs provide the
  - necessary data for frontend functionality.(Will work on it after completing the rest of the tasks.)
  -

2. Design System Architecture:



3. Plan API Requirements:

- Fetch Products : Endpoint: /products (GET).
- Fetch Product Details : Endpoint: /products/:id (GET).
- Create Order : Endpoint: /orders (POST).
- Fetch Orders by User: Endpoint: /users/:id/orders (GET).
- Order Tracking: Endpoint: /shipment/:id (GET).

First, check API endpoints on the browser.

Then use them in my project to take data and send it to Sanity.

Finally, use the data on the UI..