Marketplace Project Day 2 Technical Foundation

1. Overview

The focus of Day 2 is to shift from the business strategies outlined on Day 1 to establishing a solid technical structure for the marketplace. This includes designing the system architecture, outlining API requirements, and preparing documentation for workflows and data schemas. The project leverages Sanity CMS for backend management and incorporates third-party APIs to handle payments and shipment tracking.

2. DefineTechnical Requirements

Frontend :

- **Framework: Next.js**
  - A React-based framework designed for building server-rendered and statically generated web applications.
- **Component Library: shadcn**
  - A collection of customizable and accessible components to ensure consistent UI design.
- **Styling: Tailwind CSS**
  - A utility-first CSS framework that facilitates rapid UI development.
- **Shipping Integration: ShipEngine**
  - An API for generating shipping labels, calculating rates, and tracking shipments.
- **Payment Integration: Stripe**
  - A secure payment gateway for processing online transactions.

Pages:

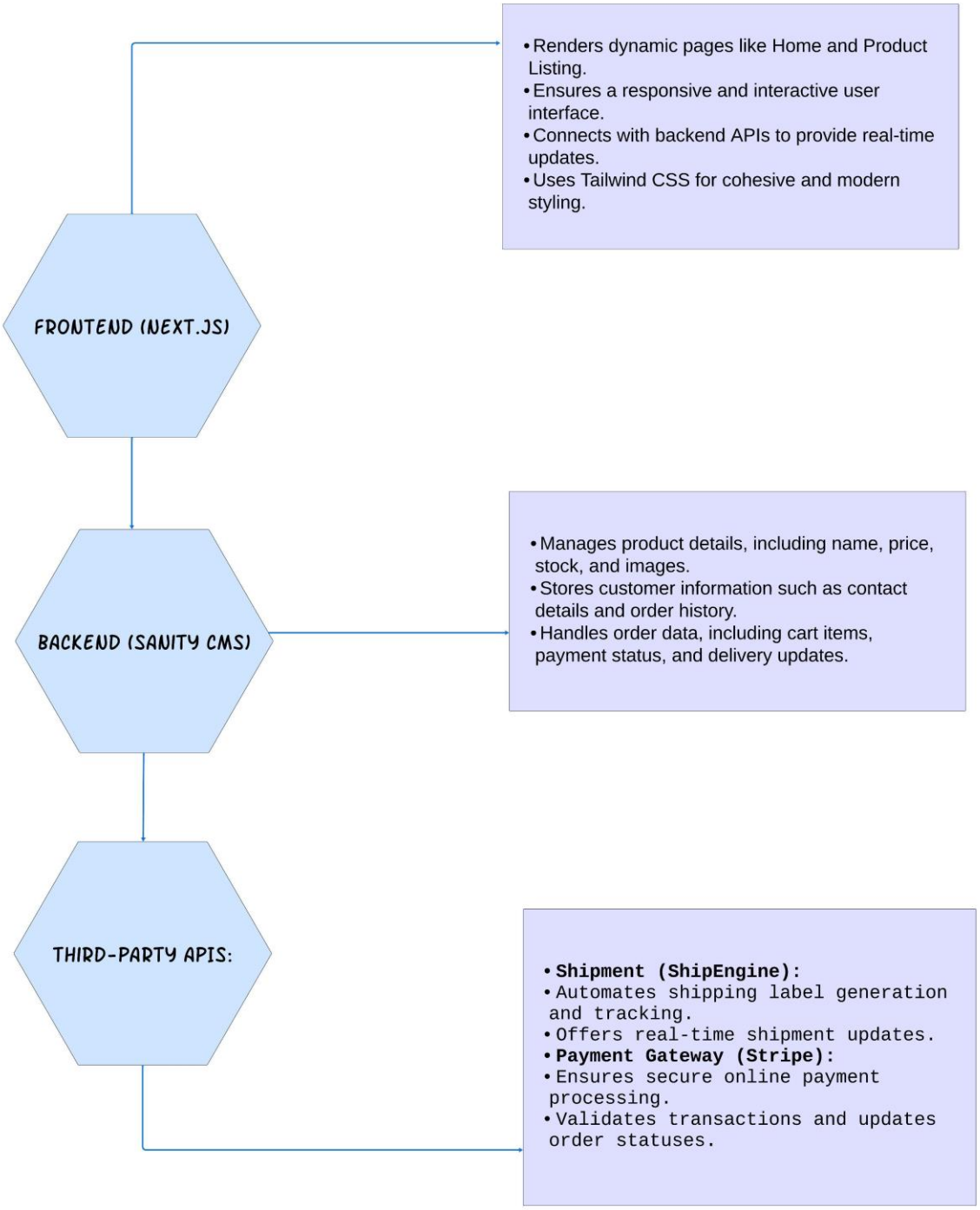
- **Home:**
  - Serves as the landing page, showcasing the marketplace's mission, featured products, and highlights.
- **Product Listing:**
  - Displays all available products, with options for filtering and sorting.
- **Product Details:**
  - Offers comprehensive details about a product, including images, pricing, and descriptions.
- **Cart:**

    - Enables users to review and manage selected products before proceeding to checkout.
- **Checkout:**
    - A page where users can enter shipping information and complete the payment process.
- **Order Confirmation:**
    - Provides a summary of the order details and confirms successful transactions.

Backend:

- **CMS:** Sanity CMS
    - Provides a flexible and efficient way to manage content such as products, orders, and customers.
    - Enables developers to design and customize schemas to meet specific business needs.
    - Supports real-time updates, ensuring the frontend reflects the latest data instantly.
- **Third-Party APIs:**
    - **ShipEngine:**
        - Integrates shipping functionalities, including creating labels and tracking packages.
        - Reduces complexity by automating shipping rate calculations.
        - Provides real-time shipment updates to users.
    - **Stripe:**
        - Manages secure payment processing for various payment methods.
        - Handles complex financial transactions with fraud protection.
        - Provides detailed payment status updates for seamless order management.

3. System Architecture Design



4. API Endpoints

1. **Products**
   - **Endpoint:** `/products`
   - **Method:** GET
   - **Description:** Fetch all available products from Sanity.
   - **Response Example:** `{ "id": 1, "name": "Product A", "price": 100 }`
2. **Orders**

- **Endpoint:** `/orders`
- **Method:** POST
- **Description:** Create a new order in Sanity.
- **Payload Example:** `{ "customer": "Harry", "items": [...], "paymentStatus": "Paid" }`

3. **Shipment**
   - **Endpoint:** `/shipment`
   - **Method:** GET
   - **Description:** Fetch order tracking status.
   - **Response Example:** `{ "orderId": 123, "status": "In Transit", "ETA": "2 days" }`

5. Sanity Schemas

Product Schema:

```
export default {
  name: "product",
  type: "document",
  fields: [
    { name: "name",
      type: "string",
      title: "Product Name"
    },
    { name: "price",
      type: "number",
      title: "Product Price"
    },
    { name: "stock",
      type: "number",
      title: "Available Stock"
    },
    { name: "size",
        type: "string",
        title: "Product size"
      },
    { name: "description",
      type: "text",
      title: "Product Description"
    },
    { name: "image",
      type: "image",
      title: "Product Image"
    },
  ],
};
```

THANK YOU

Prepared by Bilal Hassan