Created by : Rameen Rashid RollNo: 00474272

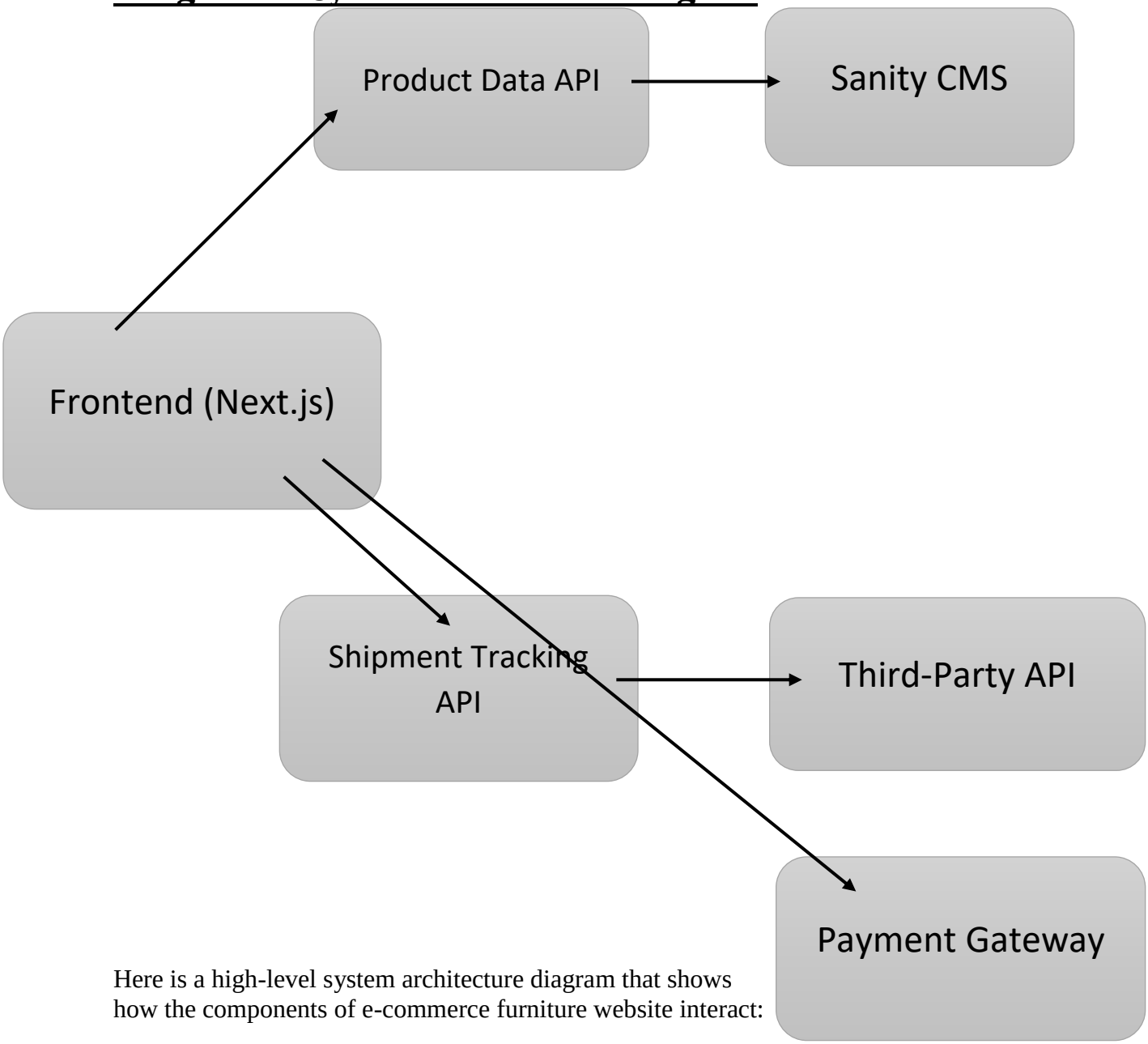
HACKATHON DAY 2 : PLANNING THE TECHNICAL FOUNDATION

1.Technical Requirements:

- **User Management:**
  - User registration and login.
  - User profile management (address, payment details, order history).
- **Product Management:**
  - Product catalog with categories, subcategories, and filters (Category, Product Type, Price, Brand).
  - Product details page with images, descriptions, dimensions, quantity, and features
  - Inventory management with stock updates.
- **Shopping Cart and Checkout:**
  - Add to cart, edit, and remove items.
  - Secure and responsive checkout process.
  - Multiple payment gateways (credit/debit cards, easypaisa, jazzcash, digital wallets).
  - Discount codes and promotions.
- **Order Management:**
  - Order tracking and history for customers.
  - Admin dashboard for order processing and status updates.
- **Shipping and Delivery:**
  - Integration with shipping services for real-time delivery updates.
  - Multiple delivery options and shipping cost calculations.
- **Search and Navigation:**
  - Advanced search with filters (Category, Product Type, Price, Brand)
  - Breadcrumb navigation and intuitive menu structure.
- **Reviews and Ratings:**
  - Customer reviews and product ratings.
  - Moderation tools for managing reviews.
- **Wishlist and Favorites:**
  - Save products for future purchase.
- **Content Management System (CMS):**
  - Manage blog posts, promotional banners, and content pages.
- **Customer Support:**
  - Contact forms and FAQ pages.

- **Responsiveness:**
  - Mobile-friendly design (responsive UI/UX).
- **Third-Party Integrations:**
  - Payment gateways credit/debit cards, easypaisa, jazzcash, digital wallets).
  - Shipping services (TCS ,M&P Express, Leopards).
  - Marketing tools (email campaigns, social media).

2.High-Level System Architecture Diagram



Here is a high-level system architecture diagram that shows how the components of e-commerce furniture website interact:

1. **Frontend (Next.js):** The user interface where customers browse products and interact with the website.
2. **Product Data API:** Connects the frontend to the **Sanity CMS** for fetching product data.
3. **Sanity CMS:** Manages product details, customer data, and order records.
4. **Shipment Tracking API:** Handles shipment information and connects to third-party logistics services.
5. **Third-Party API:** Supports external services like shipping and logistics.

6. **Payment Gateway:** Processes secure transactions initiated by the frontend.

3.API Endpoints for Furniture E-commerce Website

API Endpoints

User Authentication

- **Endpoint:** `/auth/register`
- **Method:** `POST`
- **Purpose:** Registers a new user account
- **Response Example:**

```
{"id": "1", "name": "RameenRashid", "email": "rameenrashid15@gmail.com"
}
```

- **Endpoint:** `/auth/login`
- **Method:** `POST`
- **Purpose:** Logs in a user and returns an authentication token
- **Response Example:**

```
{ "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9" }
```

Product Management

- **Endpoint:** `/products`
- **Method:** `GET`
- **Purpose:** Fetches all product details
- **Response Example:**

```
 [
  { "id": "1", "name": "Pseudo Sofa", "price": 980, "stock": 10 },
  { "id": "2", "name": "The Dandy Chair", "price": 250, "stock": 5 }
]
```

- **Endpoint:** `/products/{id}`
- **Method:** `GET`
- **Purpose:** Fetches details of a single product by ID
- **Response Example:**

```
{ "id": "1", "name": "Pseudo Sofa", "price": 980, "description":
"Comfortable 3-seater sofa", "stock": 10 }
```

- **Endpoint:** `/products`
- **Method:** `POST`
- **Purpose:** Adds a new product (Admin only)

- **Response Example:**
  { "id": "3", "name": "Leather Recliner", "price": 699.99, "stock": 7 }

Cart Management

- **Endpoint:** `/cart`
- **Method:** `GET`
- **Purpose:** Fetches the user's current cart items
- **Response Example:**

```
 [
  { "productId": "1", "name": "Pseudo Sofa", "quantity": 1, "price":
980}
]
```

- **Endpoint:** `/cart`
- **Method:** `POST`
- **Purpose:** Adds a product to the cart
- **Response Example:**

```
{ "productId": "1", "quantity": 1, "status": "Added to cart" }
```

Order Processing

- **Endpoint:** `/orders`
- **Method:** `POST`
- **Purpose:** Places a new order
- **Response Example:**

```
{ "orderId": "123", "status": "Processing", "total": 980 }
```

- **Endpoint:** `/orders/{id}`
- **Method:** `GET`
- **Purpose:** Retrieves order details
- **Response Example:**

```
{ "orderId": "123", "status": "Shipped", "deliveryDate": "2025-01-16" }
```

Payment Processing

- **Endpoint:** `/payments`
- **Method:** `POST`
- **Purpose:** Processes the payment for an order

- **Response Example:**

```
{ "paymentId": "pay_7890", "status": "Success", "amount": 980}
```

Shipment Tracking

- **Endpoint:** `/shipment/{orderId}`
- **Method:** `GET`
- **Purpose:** Tracks the shipping status of an order
- **Response Example:**

```
{ "orderId": "123", "status": "Out for delivery", "estimatedDelivery":
"2025

-01-25" }
```