**Name: Ammarah Owais**

**Roll No. : 00015418**

**Tempelet#8**

**DAY 2**

**PLANNING THE TECHNICAL FOUNDATION FOR FURNITURE WEBSITE**

**Technical Requirements:**

* **Responsive Design**: Ensure the website adapts to all screen sizes (desktop, tablet, mobile).
* **Navigation**: Intuitive navigation with a clear menu structure.
* **Fast Load Times**: Optimize images, use lazy loading, and minify CSS/JS for speed.
* **Frontend Technology**:
  + Use modern frameworks like Next.js, React, Angular, for dynamic UIs.
  + HTML5, CSS, and JavaScript for foundational elements.
* **Backend Technology**:
  + Select a backend framework (e.g., Node.js, Django, sanity).
  + Handle API integration and business logic.
* **CMS Integration**:
  + Use a CMS like WordPress, Sanity, or Contently for content management.
* **Database**:
  + Choose a suitable database (e.g., MySQL, PostgreSQL, MongoDB) for storing data.

**4. Functionality**

* **User Authentication**: Allow secure login/logout with session management.
* **Search Functionality**: Include a search bar for content discovery.
* **Forms**: Contact forms, registration forms, etc., with data validation.
* **E-commerce Features** (if applicable):
* **Third-party Integrations**:
  + Social media, analytics, and email services.
* **SEO Optimization**:
  + Meta tags, sitemaps, and schema markup for search engine visibility.

**System architecture for a Sofa furniture market website involves;**

### **1**. Frontend****

The user interface for interacting with the website.

* **Technologies**: React.js, Next.js, Tailwind CSS.
* **Features**:
  + Homepage with featured sofas.
  + Product catalog with categories, filters, and sorting.
  + Product detail pages with high-resolution images, descriptions, and reviews.
  + Shopping cart and checkout.
  + User accounts for order tracking and wish lists.

**2. Backend**

Handles business logic, data processing, and API endpoints.

* **Technologies**: Node.js, Express.js, or REST API.
* **Features**:
  + API for fetching product data.
  + User authentication and authorization.
  + Order management system.
  + Payment processing integration.
  + Review and ratings system.

**3. Database**

Stores and manages data.

* **Entities**:
  + **Products**: Name, category, price, description, inventory, images.
  + **Users**: Name, email, hashed passwords, order history.
  + **Orders**: Order ID, user, product list, status, timestamps.
  + **Reviews**: User, product, rating, comments.

**4. Content Management System (CMS)**

Manages dynamic content like blogs, promotions, and updates.

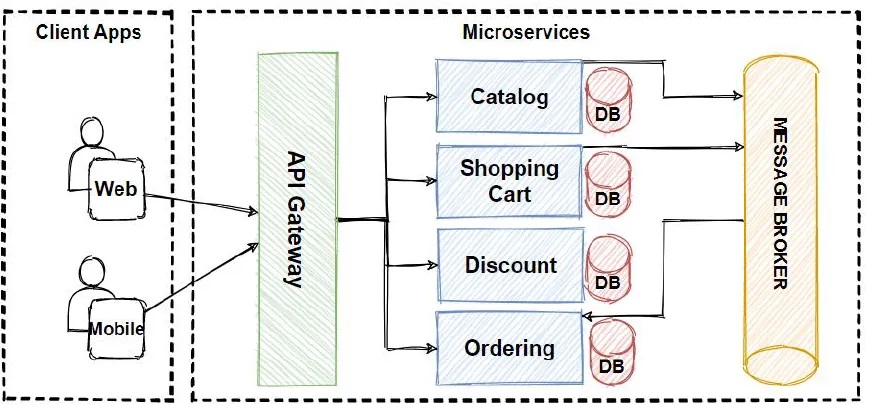
* **Technology**: Sanity.io

**5. Third-Party Integrations**

* **Payment Gateway**: PayPal
* **Shipping API**: FedEx, DHL.
* **Analytics**: Google Analytics,
* **Social Media**: Facebook, Instagram, Twitter integrations

**6.Hosting Environment** for deployment.

**System Architecture Diagram:**



**Plan API Requirements:**

* Endpoint Name: /product

o Method: GET

o Description: Fetch all available products from Sanity

o Response: Product details (ID, name, price, stock, image)

• Endpoint Name: /orders o Method: POST

o Description: Create a new order in Sanity.

o Payload: Customer info, product details, payment status.

• Endpoint Name: /shipment

o Method: GET o Description: Track order status via third-party API.

o Response: Shipment ID, order ID, status, expected delivery date.

**API Endpoints:**

o Provide a table format:

♣ Endpoint: /products

♣ Method: GET

♣ Purpose: Fetches all product details

♣ Response Example: { "id": 1, "name": "Product A", "price": 100 }