Day 2 Assignment: Technical Plan for Furniture Marketplace

### Step 1: System Architecture Design

The architecture of the system defines how different components will interact to deliver the desired functionality.

1. **Frontend:**
   - **Tech Stack:** React.js, HTML, CSS, JavaScript.
   - **Details:**
     - The frontend will provide a user-friendly interface where customers can browse products, customize their furniture, and place orders.
     - Features include filtering products by category, adding items to the cart, and responsive design for desktop and mobile devices.

2. **Backend:**
   - **Tech Stack:** Node.js, Express.js.
   - **Details:**
     - The backend will handle core logic, user authentication, API endpoints, and order management.
     - Middleware will be used for validation and secure data flow.

3. **Database:**
   - **Tech Stack:** MongoDB.
   - **Details:**

- MongoDB will store user profiles, product details, and order information.
- Data indexing will optimize search performance.

4. **External Services:**
   - **Sanity CMS:** Content management for storing product data.
   - **Stripe API:** Secure payment processing.
   - **Twilio API:** Order notifications via SMS.

### Step 2: Workflows and Diagrams

1. **User Registration Workflow:**
   - User visits the registration page and fills in details.
   - Backend validates and stores the data securely.
   - A welcome email/SMS is sent to the user.

2. **Order Placement Workflow:**
   - User browses and selects products.
   - Customizations (like material or color) are added.
   - Items are added to the cart.
   - During checkout, payment is processed using Stripe, and order details are stored.

### Step 3: APIs and Schema Design

1. **APIs:**

API Name	Endpoint	Method	Purpose
Register User	/api/users/register	POST	Register a new user.
Login User	/api/users/login	POST	Authenticate and log in a user.
Fetch Products	/api/products	GET	Retrieve all products.
Place Order	/api/orders	POST	Create a new order and process payment.

2. **Database Schema:**

- Users Collection:

```
{
  "_id": "unique_user_id",
  "name": "John Doe",
  "email": "john.doe@example.com",
  "password": "hashed_password",
  "address": "123 Street, City",
  "phoneNumber": "1234567890"
}
```

### Step 4: Diagram for Sanity CMS Integration

Sanity CMS serves as a central content management system. The backend fetches product data using the Sanity API, and the frontend dynamically displays it.

Diagram:

```
Frontend <--> Backend <--> Sanity CMS
     \         /
      \--> MongoDB <--/
```
