To design the system architecture and API endpoints for an e-commerce business specializing in furniture, I'll break it down into two parts: **System Architecture** and **API Endpoints**. I'll also provide two different charts representing the business for better visualization.

**1. System Architecture**

The system architecture of an e-commerce platform for furniture could be a typical **3-tier architecture**:

**Components:**

1. **Frontend (Client-side)**:
   * **UI/UX**: The user interface for browsing products, filtering items, and completing purchases.
   * Built using frameworks like React or Next.js, as you have experience with them.
2. **Backend (Server-side)**:
   * **Business Logic Layer**: Handles the core functionalities like managing products, users, inventory, and orders.
   * **API Layer**: Exposes RESTful APIs for the frontend to interact with the server. Typically built using Node.js (Express.js) or similar frameworks.
3. **Database**:
   * A **relational database** like PostgreSQL or MySQL for storing product data, user details, orders, etc.
4. **Payment Gateway**:
   * Integration with a payment service (e.g., Stripe or PayPal) to handle transactions securely.
5. **Admin Dashboard**:
   * A backend management tool for administrators to add/remove products, view orders, and manage customer data.
6. **Search Engine**:
   * A search engine like **Elasticsearch** can be integrated to allow users to search for products efficiently.

**Architecture Diagram:**

plaintext

CopyEdit

+----------------+ +--------------------+ +-----------------+

| Frontend | ---> | Backend API | ---> | Database |

| (React/Next.js)| | (Node.js/Express) | | (MySQL/PostgreSQL)|

+----------------+ +--------------------+ +-----------------+

| | |

+--------------+ +--------------+ +-----------------+

| Payment | | Admin Panel | | Search Engine |

| Gateway (e.g.,| | (CRUD Ops) | | (Elasticsearch) |

| Stripe/PayPal)| +--------------+ +-----------------+

+--------------+

**2. API Endpoints**

The API endpoints allow communication between the frontend and backend, and they are typically RESTful APIs for different functionalities.

**API Endpoints for Furniture E-commerce:**

* **Authentication**
  + POST /api/auth/login: Login a user.
  + POST /api/auth/register: Register a new user.
  + POST /api/auth/logout: Log the user out.
* **Product Management**
  + GET /api/products: Fetch all products.
  + GET /api/products/{id}: Fetch a single product by its ID.
  + POST /api/products: Add a new product (Admin only).
  + PUT /api/products/{id}: Update an existing product (Admin only).
  + DELETE /api/products/{id}: Delete a product (Admin only).
* **Order Management**
  + POST /api/orders: Create a new order.
  + GET /api/orders/{userId}: Get all orders for a specific user.
  + PUT /api/orders/{id}: Update order status (Admin only).
  + GET /api/orders/{id}: Get details of a specific order.
* **User Profile**
  + GET /api/users/{id}: Fetch user profile details.
  + PUT /api/users/{id}: Update user profile details.
* **Cart Management**
  + POST /api/cart: Add an item to the cart.
  + GET /api/cart/{userId}: Get items in the user's cart.
  + DELETE /api/cart/{itemId}: Remove an item from the cart.
* **Search & Filtering**
  + GET /api/products/search: Search products by keywords.
  + GET /api/products/category/{categoryId}: Filter products by category.
  + GET /api/products/price-range: Filter products within a price range.