

Project Design

Initial Design:

At the beginning of the project, our goal was to create an efficient system for managing furniture inventory, ensuring fast and easy retrieval of information related to both the common attributes (such as dimensions and material) and special features of each piece. Initially, we thought the best approach was to create three separate tables:

1. **Furniture Table** – This table contains the common attributes for all furniture items. Its primary key is the model number of each item.
2. **Inventory Table** – This table tracks inventory details and includes columns for the model number and quantity. In this table, the model number serves as both a foreign key (referencing the Furniture Table) and the primary key.
3. **Individual Furniture Tables** – Each type of furniture has its own table that includes the model number (as the primary key) along with the specific features unique to that item.

Issues with the Initial Design:

Despite the initial design seeming reasonable and in line with our goals, we quickly realized that accessing the data required multiple steps. For example, in order to retrieve specific information about a furniture item in stock, we would first have to go to the inventory table, then to the specific furniture table, and finally, access the main furniture table to get all the relevant data. This approach created significant complexity when trying to filter or retrieve information, especially as the volume of data grew. The need to navigate between multiple tables made the system less efficient.

Design Change:

After realizing the complexity of the initial design, we decided to combine the furniture and inventory tables into one table. This new design merged the inventory data (such as quantity) with the common attributes of each furniture item, while the unique attributes (special features) for each furniture piece were stored in a separate field json type called "Details."

Advantages: By combining the furniture and inventory data into a single table, we simplified the structure and reduced the number of queries needed to access relevant information. This made data retrieval more efficient and easier to manage. Storing the unique attributes in the "Details" field allowed for flexibility, enabling us to handle special features without complicating the main table structure.

We also faced issues with Data Storage:

Initially, we considered using a **remote server** to store the data. However, we were unable to find a suitable free option that would work well for the project. Consequently, we opted to use JSON files for data storage. Unfortunately, we soon discovered that this approach was

inefficient and not ideal for our needs. For example, when displaying all inventory items with their complete details—such as color, dimensions, and other specific attributes, as might be shown on a homepage—the process required several steps. First, you must iterate through all model numbers listed in the inventory JSON file. Then, for each model number, you need to search the general furniture table to retrieve shared attributes (like name and category). Finally, you have to open the specific JSON file for that type of furniture, search for the model number there, extract its unique attributes, and merge the data. This multi-step process leads to slower performance and increased complexity, particularly as the number of items grows.

Final Solution:

After further evaluation, we decided to switch to **SQLite** using **SQLAlchemy**. This transition allowed us to utilize a structured query language (SQL) to efficiently query the database, significantly improving the speed and accuracy of data retrieval. SQLAlchemy's integration with SQLite provided a much more efficient way to handle data and perform queries, making it the ideal choice for our project.

Conclusion:

Throughout the process, we learned that the design needed to evolve to keep pace with the growing complexity of the data. By combining the furniture and inventory tables, and switching to SQLite for better data storage and retrieval, we achieved a more efficient and scalable solution. These changes not only simplified the database structure but also enhanced the performance of our application, allowing for faster and more reliable queries. Initially, we thought the original design would work best, but ultimately, the iterative process of testing and reevaluating led us to a more practical solution.

Innovative Enhancements and Feature Highlights

Our project showcases our creative approach and thoughtful design through several key enhancements:

Distinct Admin and User Interfaces

- **Admin Registration:** Admins register via a dedicated link reserved exclusively for authorized personnel. This provides access to specialized API endpoints (e.g., updating inventory quantities) that are restricted to admin users only.
- **User Registration:** Regular users sign up through a separate link, ensuring that non-admins cannot mistakenly gain admin privileges. Every API call enforces role validation, confirming that the role is either "admin" or "user."

Access Control

- **Public vs. Private Features:**

Browsing: Visitors can browse and view furniture details without needing to log in.

Shopping Cart: Actions like adding items to the cart require user authentication, protecting the integrity of user-specific operations.

Enhanced Furniture Attributes

- **Detailed Specifications:**

General Furniture: We defined valid material types (e.g., wood, metal, plastic, leather, fabric) to ensure consistent and accurate descriptions.

Sofas: Feature a specific attribute for upholstery types (e.g., leather, fabric, velvet, synthetic).

Bookshelves: Automatically include features like the number of shelves and the maximum capacity weight per shelf.

User Account Management

- **Personal Information Updates:** Users can update key personal details such as address, full name, phone number, email, and even an encrypted password. This feature allows users to correct any inaccuracies from the initial registration, enhancing overall user experience.

These enhancements not only add robust functionality to our online store but also ensure a secure, user-friendly, and scalable application.