Furniture Store Client Requirements

This document describes the specifications, by which the front-end client is created. You may use it as a reference during the implementation of a REST API that serves data in the expected format.

1. Implement Furniture Store

You can make requests to the following endpoints:

- Register User (POST): **http://localhost:3030/users/register**
- Login User (POST): **http://localhost:3030/users/login**
- Logout User (GET): **http://localhost:3030/users/logout**

- Create Furniture (POST): **http://localhost:3030/data/catalog**
- All Furniture (GET): **http://localhost:3030/data/catalog**
- Furniture Details (GET): **http://localhost:3030/data/catalog/:id**
- Update Furniture (PUT): **http://localhost:3030/data/catalog/:id**
- Delete Furniture (DELETE): **http://localhost:3030/data/catalog/:id**
- My Furniture (GET): **http://localhost:3030/data/catalog?where=_ownerId%3D%22{userId}%22**
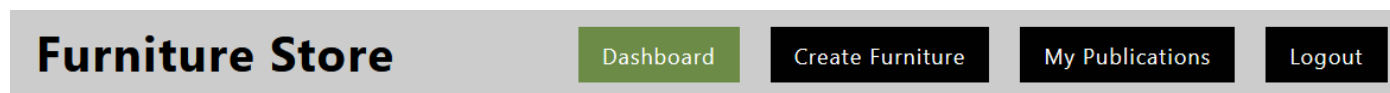
Guest (not logged in)

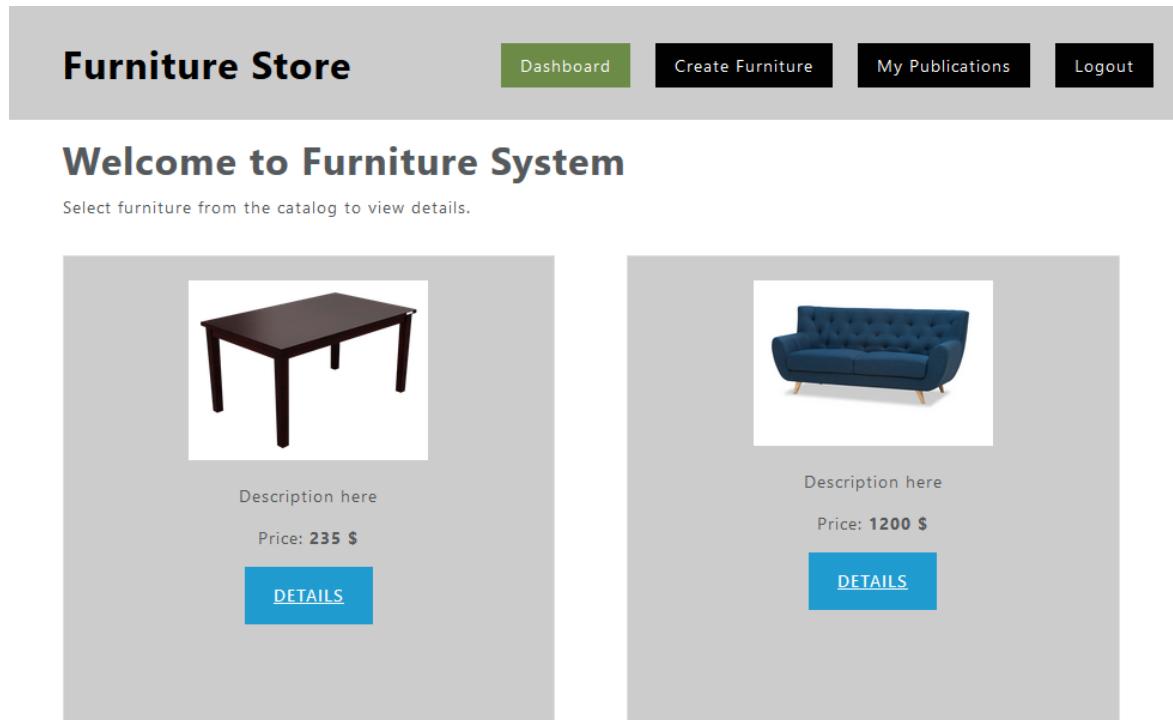**Guest** navigation example:

Users (logged in)

User navigation example:

All Furniture

List **all furniture** inside the **store**. Display appropriate links in the navigation bar, based on the user session.

Clicking on any of the **Details** buttons should **redirect to details**.

Register User

You need to write the functionality for the registration of a new user. By clicking the "**Register**" button you have to load the registration form. Register a user inside the database with an **email**, **password. Password** inside the database must be **hashed** (use bcrypt) and both **passwords** must **match**!

When the "**Register**" button of the form is clicked you need to send a post request.

If the registration is **successful** you can **redirect to the dashboard page**.

Login User

If the **user has already registered**, the user can login by using **the login form**. After **successful** login the user should be **redirected to the dashboard page**. Save the returned token in the session storage, and send it with every request.

Logout User

The **logged-in user** can be **logged out** by clicking the **logout button**. Write the functionality for this action.

Create Furniture

Create New Furniture

Please fill all fields.

Make	Price

Model	Image
Valid input	

Year	Material (optional)
Invalid input	

Description	
	CREATE

Validate fields:

- Make and Model must be **at least 4 symbols long**
- The year must be **between 1950 and 2050**
- The description must be **more than 10 symbols**
- The price must be a **positive number**
- Image **URL is required**
- Material is **optional**

By valid input, you can **add the "is-valid" class** to the input field, and **by invalidating** the **"is-invalid" class.**

If the creation is **successful** show **redirect to the dashboard page**.

Furniture Details

Get the **id** from the **URL** and **display** the information. If **the logged user is the creator** the buttons "**Edit**" and "**Delete**" should be **visible** (can be used), otherwise, they should not be visible.

Furniture Details

![Chair]

Make: **make**

Model: **model**

Year: **year**

Description: **description**

Price: **price**

Material: **material**

EDIT   **DELETE**

Update Furniture

If the **logged-in user** is **the creator** then can **edit** the furniture details. When the form is **loaded all the fields** must be **filled up** with the **information from the server**. **Validation** should be the same as the validation by creating new furniture. **PUT request** must be sent.

Edit Furniture

Please fill all fields.

Make	Price
Table	235

Model	Image
Swedish	/images/table.png

Year	Material (optional)
2015	Wood

Description	
Medium table	**EDIT**

Delete Furniture

By clicking on the "**Delete**" button the app needs **first confirmation for deleting** (you can use alert or another custom-made notification) then sends a **DELETE request** and **deletes the furniture**. Then the app **redirects** to the **dashboard**.

My Furniture

A logged-in user can see a list of their publications by clicking the link **"My Publications"** in the navigation bar. Display a page similar to the main catalog (dashboard), but only show the records that are associated with the currently logged-in user.