**Python FSD**

React Capstone Project Solution



**Home Furnishing Application**

**Steps to be followed:**

1. Pull Phase 1(Python with Django) and Phase 2(React JS) project from Git to local machine.
2. Create Docker image for both project and Docker compose file
3. Then push the code to git hub account
4. Login to AWS account
5. Create EC2 instance
6. Install all required software to deploy ie git, java, jenkin, docker and docker compose
7. Login to Jenkin dashboard
8. Create Jenkin job responsible to pull the project from git and run the docker compose file
9. Test the application

**Step 1: Install dependencies required for the project**

1. Ensure that the following tools are installed in your system:

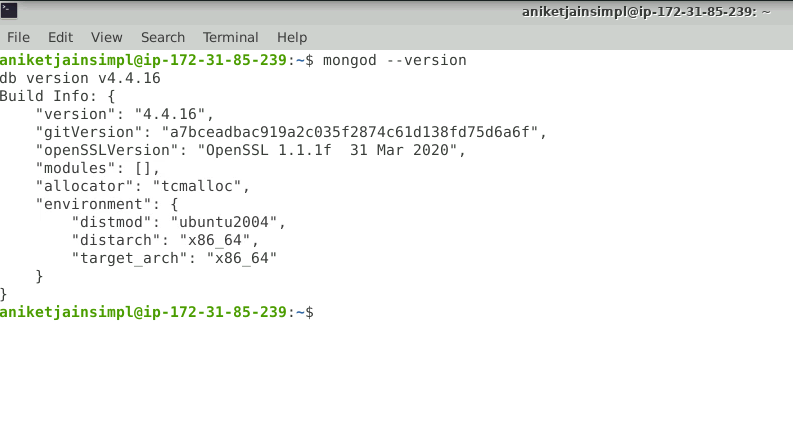
* Python 3.x
* Visual Studio Code
* Django
* Django
* PyMongo
* MongoDB
* Browser
* NodeJS
* create-react-app

Note: The guides to install these are available in the course component, and you have already used them in the training.

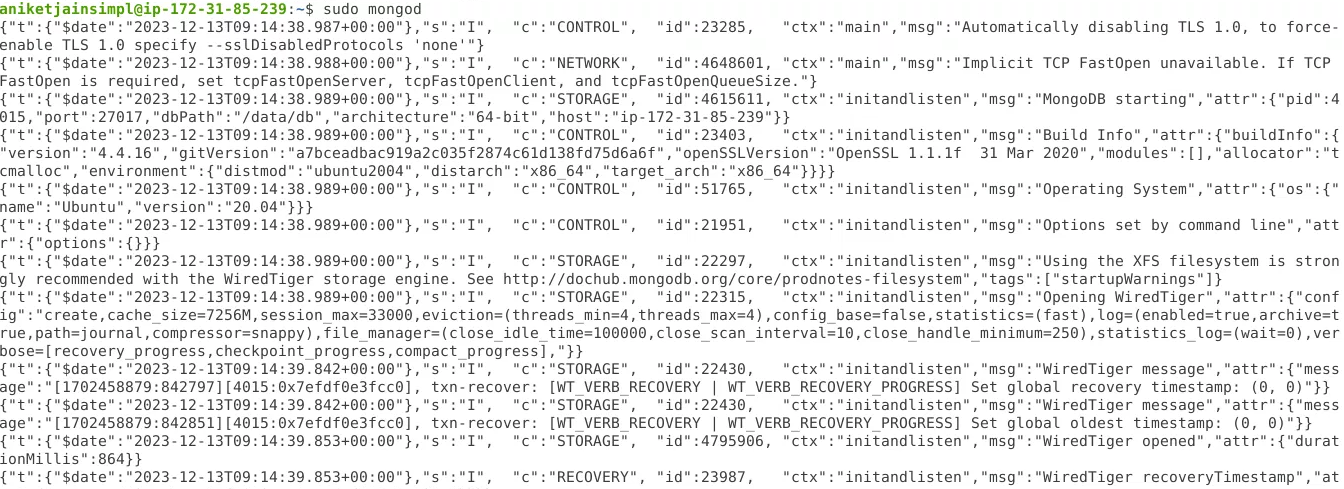
**Step 2: Start the MongoDB server and create the Rentfurlax schema in MongoDB**

* 1. Open the terminal, and run the following command to check the MongoDB server’s status:

**mongod --version**

****

* 1. Run the following command to start the MongoDB server from the terminal  
     **sudo mongod**

****

****

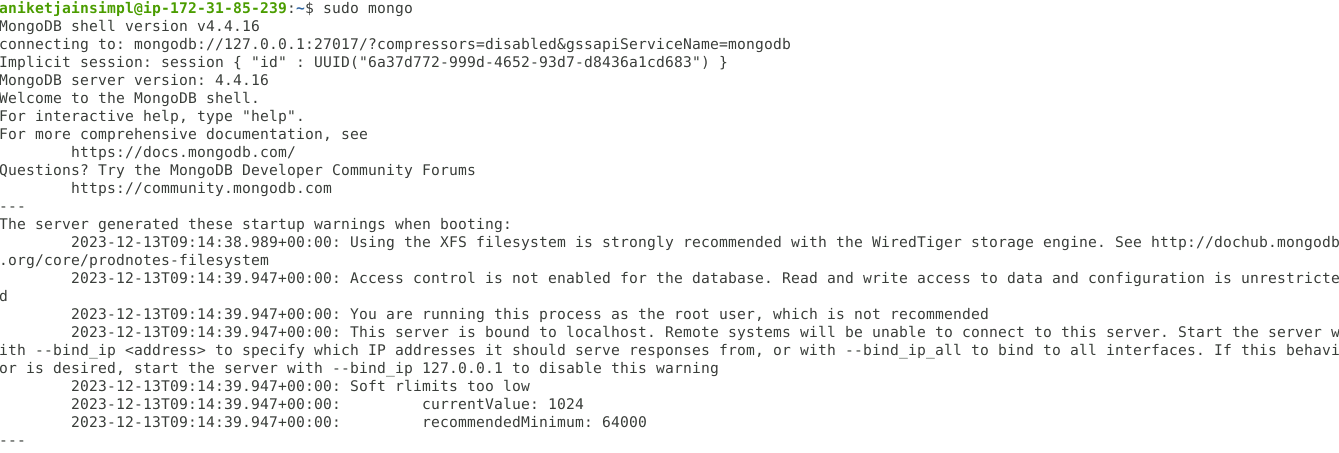
NOTE: Do not close this terminal as this is the mongo server running.

In case getting /data/db error please execute the below command:

**sudo mkdir -p /data/db**

* 1. Open another terminal and use the following command to start the mongo shell:

**sudo mongo**

******

* 1. Once the mongo shell is up and running, run the below command to see existing databases:

**show dbs**



* 1. To create a database, execute the below command:

**use rentfurlax**



* 1. To see existing collections if any, execute the below command. It will be empty.

**show collections**



**Step 3: Configure Python for CORS**

1. Execute the Django project (Capstone 1)
2. Do install cors headers by using following command:

**sudo python -m pip install django-cors-headers==3.10.1**

**A screen shot of a computer program

Description automatically generated**

1. Add **coreheaders** in **settings.py**

**INSTALLED\_APPS = [**

**'django.contrib.admin',**

**'django.contrib.auth',**

**'django.contrib.contenttypes',**

**'django.contrib.sessions',**

**'django.contrib.messages',**

**'django.contrib.staticfiles',**

**'rest\_framework',**

**'rentfurlax',**

**'corsheaders',**

**]**

**A screenshot of a computer program

Description automatically generated**

1. Update Middleware for **cors**

**MIDDLEWARE = [**

**'django.middleware.security.SecurityMiddleware',**

**'django.contrib.sessions.middleware.SessionMiddleware',**

**'django.middleware.common.CommonMiddleware',**

**'django.middleware.csrf.CsrfViewMiddleware',**

**'django.contrib.auth.middleware.AuthenticationMiddleware',**

**'django.contrib.messages.middleware.MessageMiddleware',**

**'django.middleware.clickjacking.XFrameOptionsMiddleware',**

**'corsheaders.middleware.CorsMiddleware',**

**'django.middleware.common.BrokenLinkEmailsMiddleware',**

**]**

A screenshot of a computer program

Description automatically generated

1. Add the below lines in the settings.py file to enable CORS so that the React project can request Python REST API:

**CORS\_ORIGIN\_ALLOW\_ALL = True**

**CORS\_ORIGIN\_WHITELIST = [**

**"http://127.0.0.1:3000"**

**]**

A screen shot of a computer program

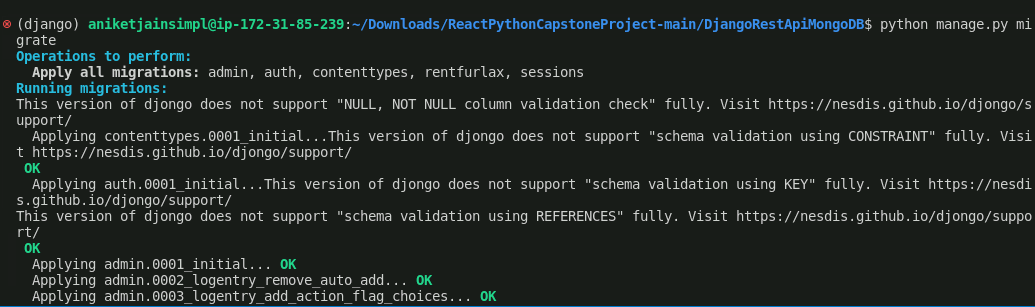
Description automatically generated

**Step 4: Start the Python Project**

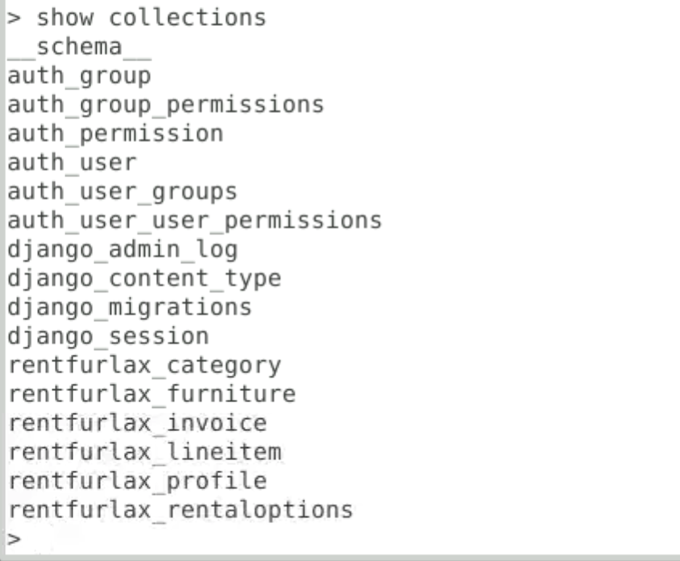
1. Run the below commands to migrate models to the database:

**python manage.py makemigrations rentfurlax  
python manage.py migrate**

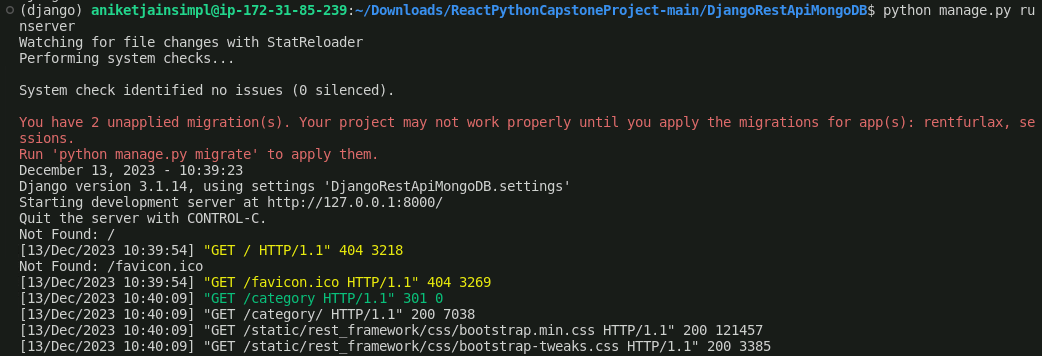
A screen shot of a computer

Description automatically generated  


1. Check Mongo shell for the collections created mapping with the models



1. Run the project by executing the command   
   **python manage.py runserver**



See the output on the browser as follows:

A screenshot of a computer

Description automatically generated

1. Refer to <https://github.com/github-simplilearn-net/PythonFSDCapstone2Mongo/tree/main/mongodata> for inserting data into the database.

**Step 5:** **Install NodeJS and create a React app**

* 1. Run the below commands to install the node:  
     **sudo apt update**  
     **curl -sL https://deb.nodesource.com/setup\_16.x | sudo bash -  
     sudo apt-get install -y nodejs**

A screen shot of a computer

Description automatically generated



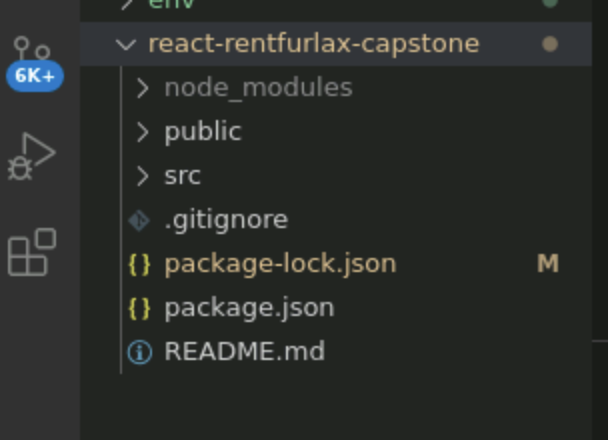
* 1. Running the below command should display the node version  
     **node -v**



* 1. Run the below command to create a project in VS Code  
     **npm create-react-app react-rentfurlax-capstone**

Refer to <https://github.com/Simplilearn-Edu/ReactPythonCapstoneProject/tree/main/react-rentfurlax-capstone> for React codes.

* 1. It will create a project folder as follows



* 1. Open the terminal from within the **react-rentfurlax-capstone** folder and execute the command  
     **npm start**

A screen shot of a computer

Description automatically generated

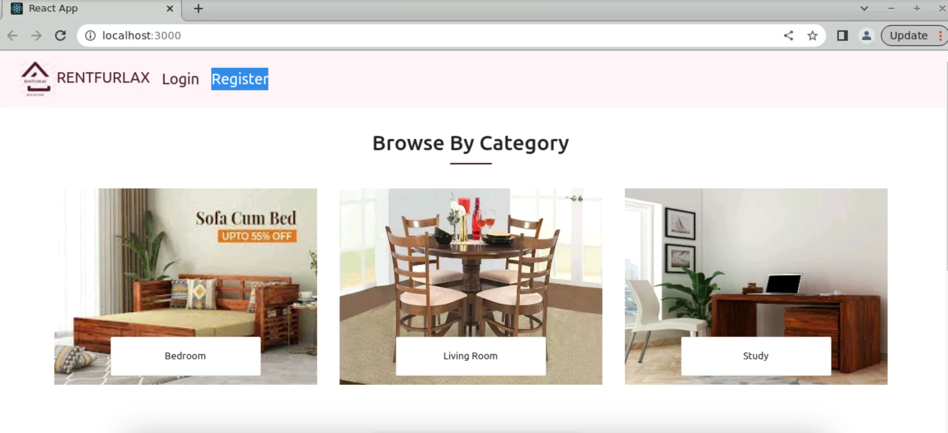
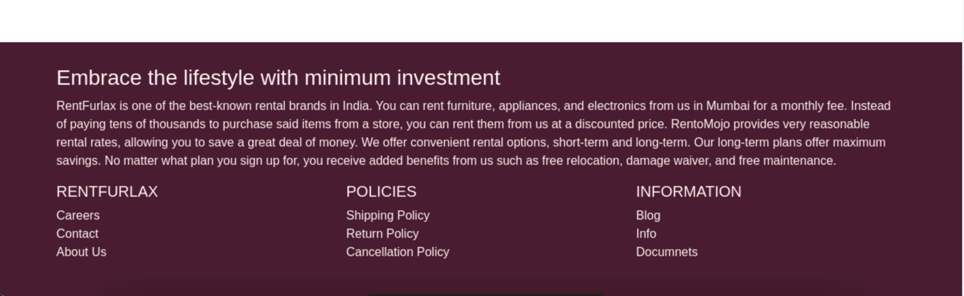
* 1. Output should be seen as below:

A screenshot of a computer

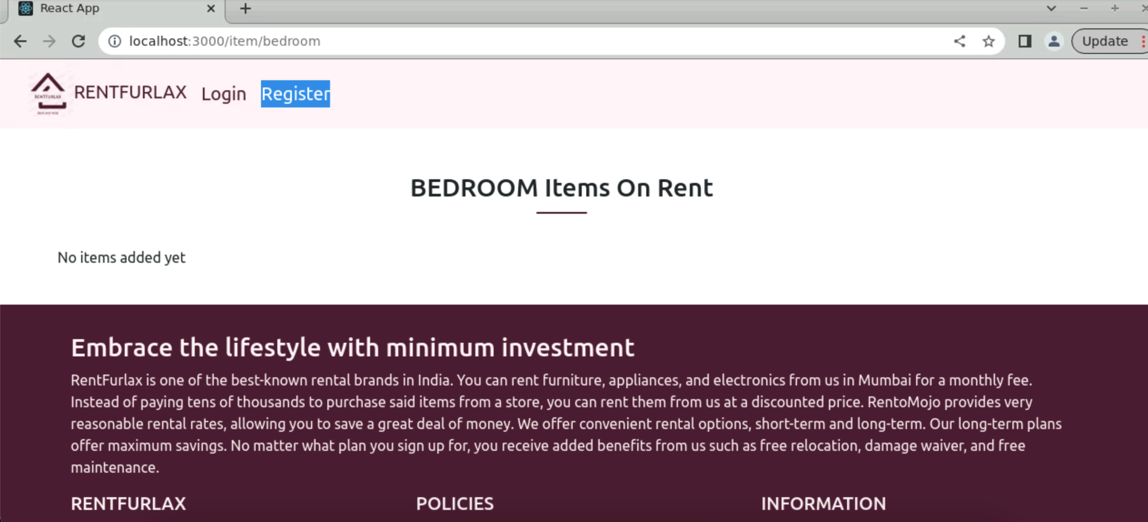
Description automatically generated

**Step 6: Test the application**

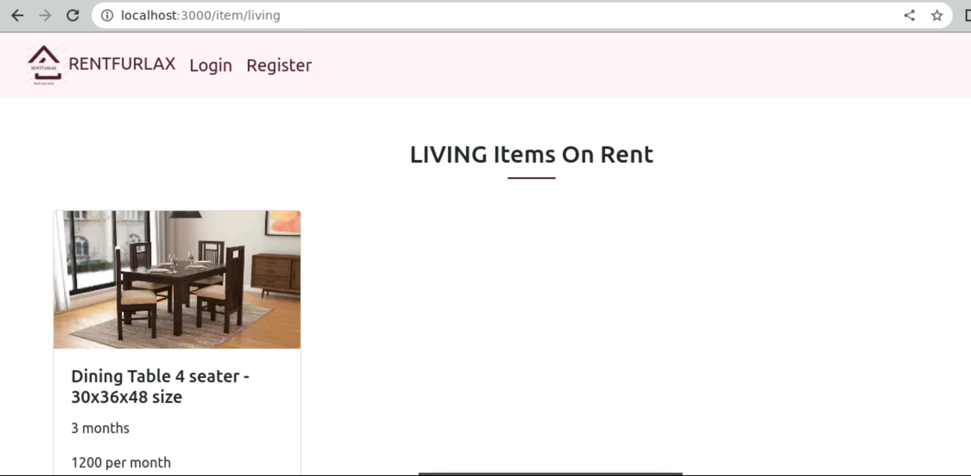
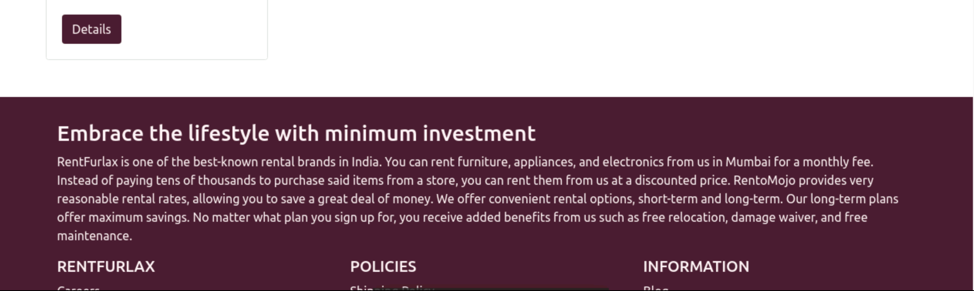
1. Home pages look as follows:

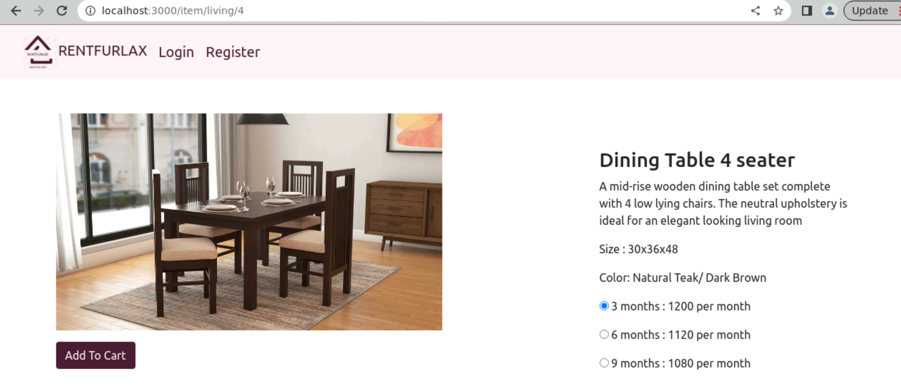
1. If there are no items for a particular category, clicking on a category should look as follows:

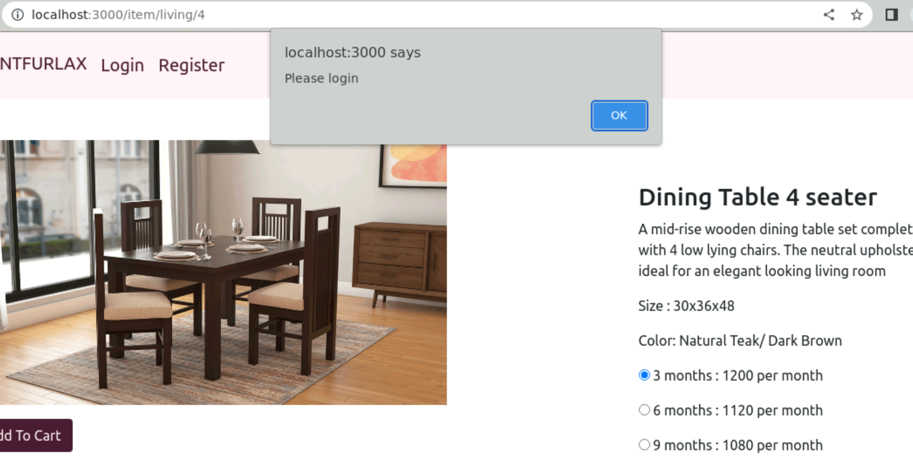


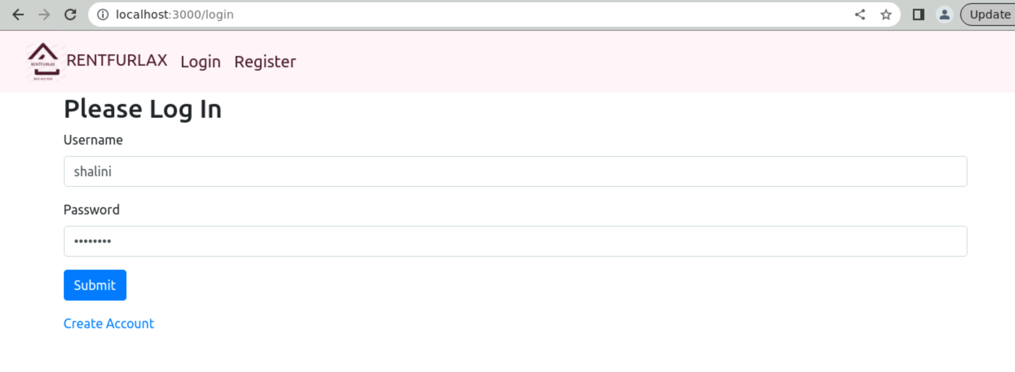
1. If items exist, the page should look as follows:

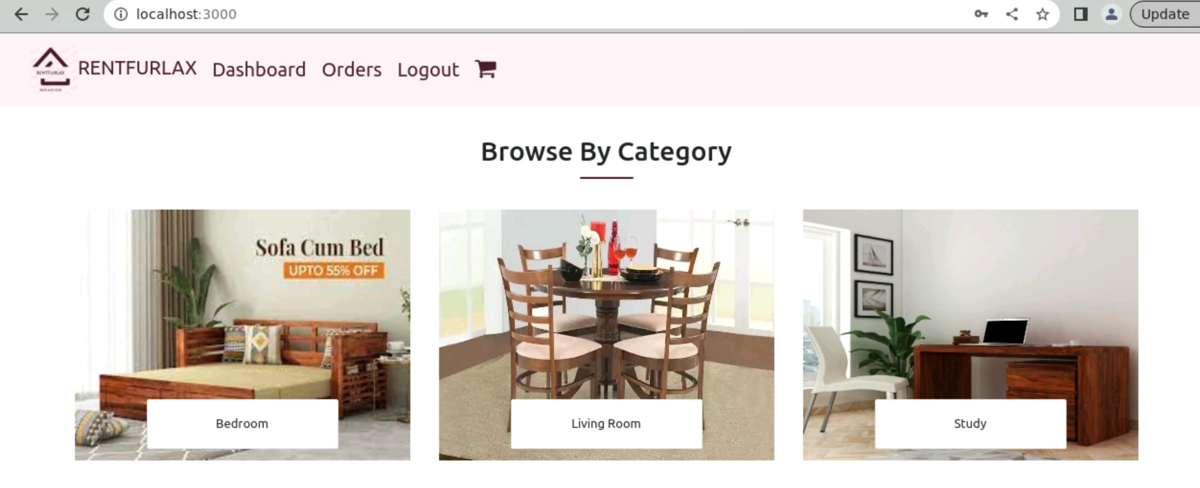
1. Clicking on details should display the page as follows:

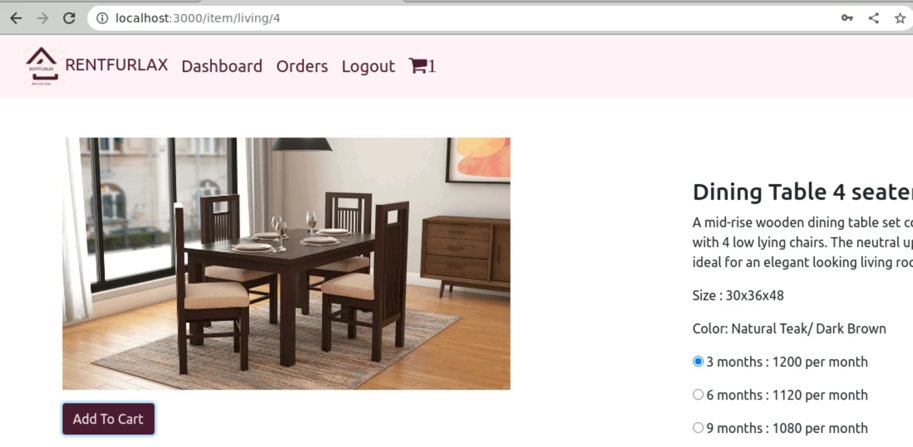

1. Clicking on Add To Cart should display the below alert if a user is not logged in:  
     
   
2. The user should be navigated to the login page as shown below:



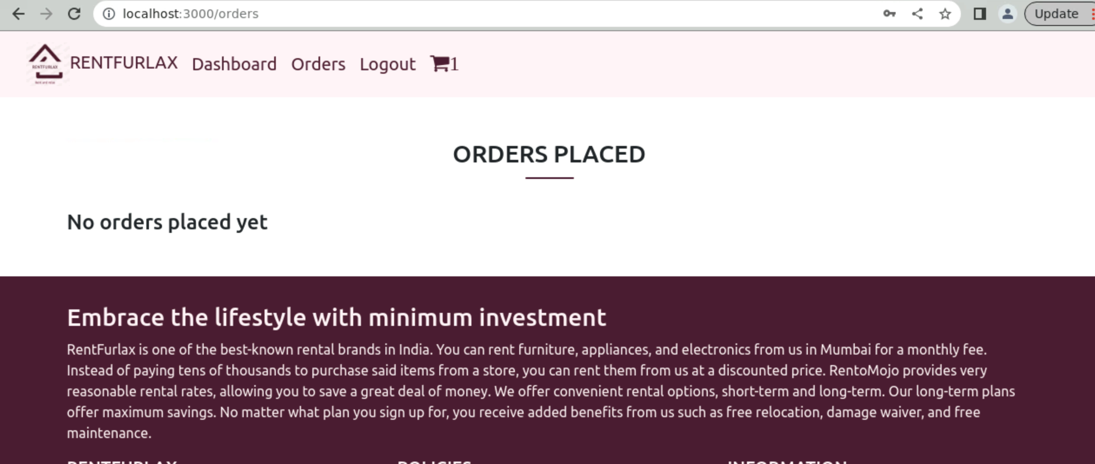
1. Once the login is successful, store the username in the session storage of the browser. It redirects to the home page with the links updated as shown below:



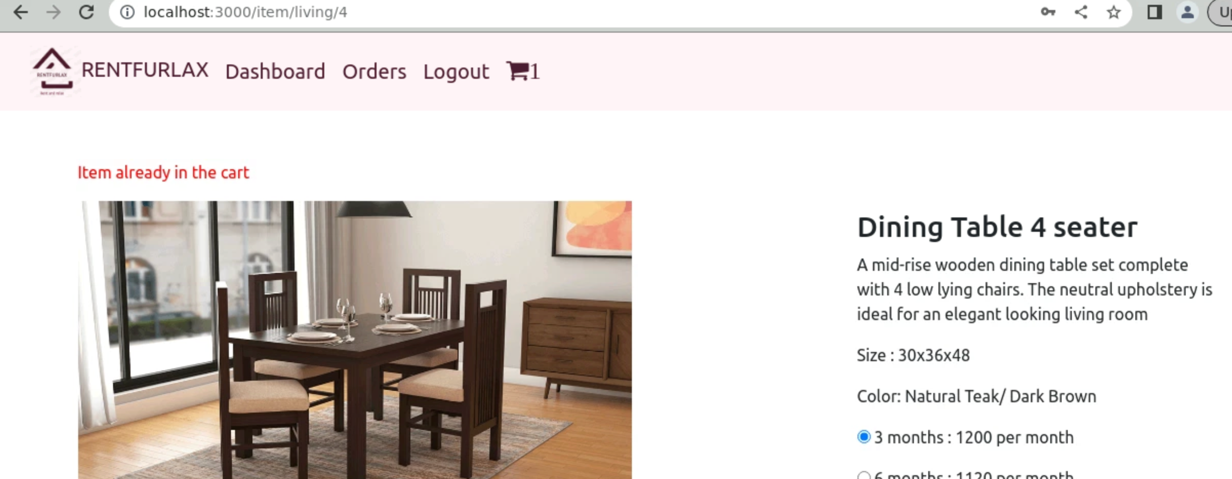
1. Now when the user clicks **Add To Cart**, it should add the item to the cart and the cart icon should have 1 as a count displayed as seen below:

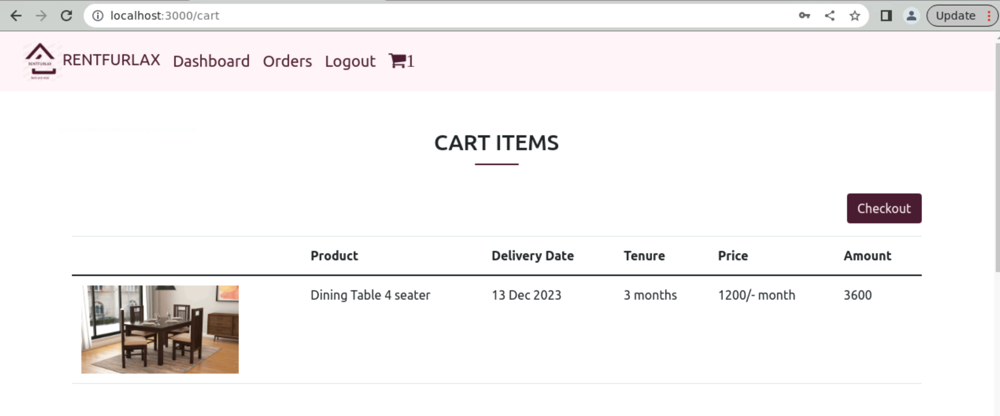


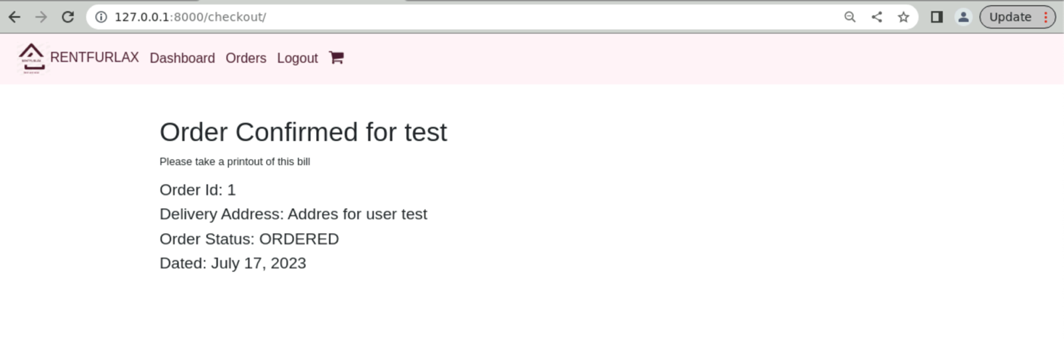
1. Clicking on orders if no items yet should display as follows:

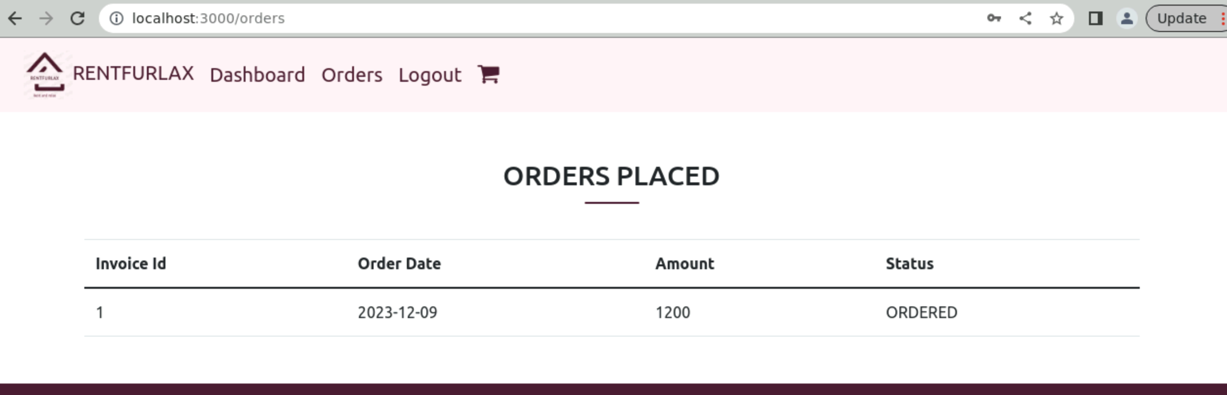


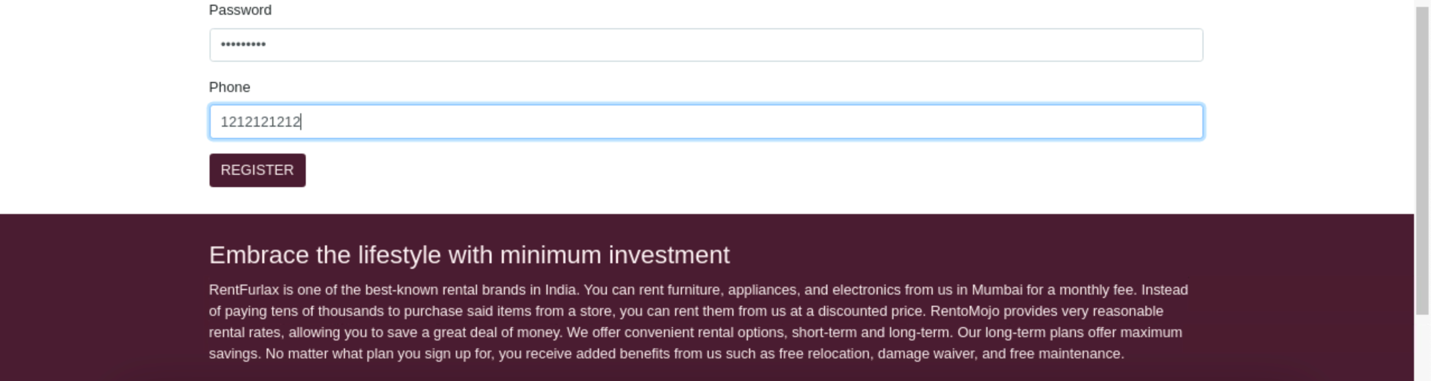
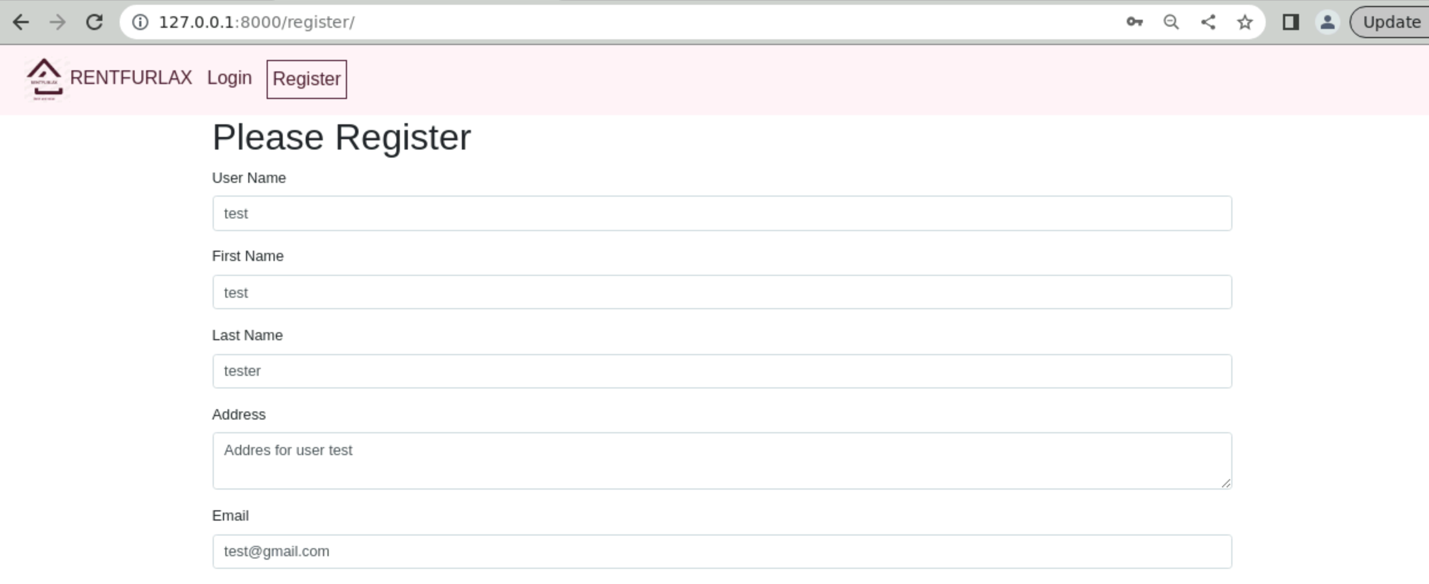
1. If the user adds the item again should display the error message as follows:



1. Clicking on the cart item should display the page as shown below  
     
   
2. Clicking on checkout should add an item to the MongoDB database, take it to the page as seen below and the cart should be empty

****

1. Now clicking on orders should display as follows:  
     
   
2. Click on Register should allow you to add a new user and redirect to the login

****