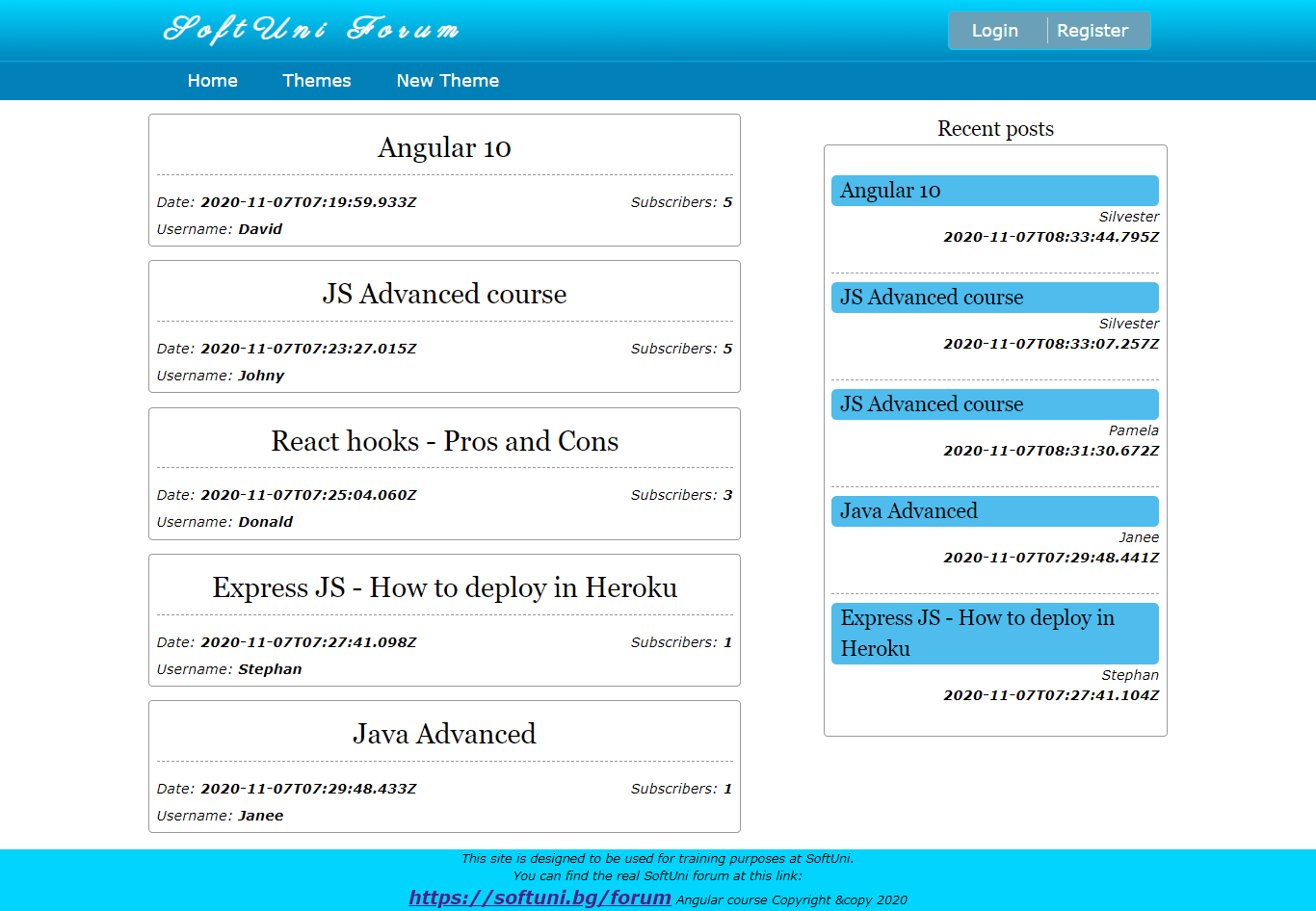
# SoftUni Forum Workshop - Part 1

This application is created as a workshop for the [Angular Corse](https://softuni.bg/trainings/3047/angular-november-2020) @ [Software University](https://softuni.bg)

# Task Requirements

You will be provided with skeleton (**HTML** and **CSS**). When starting the index.html, the following page will appear:



The application is a forum, where each registered user can create a theme or post a comment. For now, your task is to check the given skeleton (HTML and CSS files) and **split them into components**, so each part can be **reusable**. Be careful when you structure the components.

## MongoDB

Before you start with creating a new project in Angular you need to install MongoDB. You can find an installation guide in the provided document:

**MongoDB-Installation-Guide.docx**

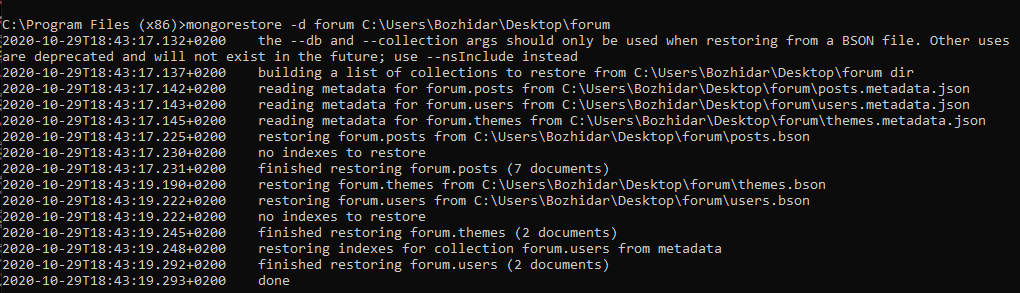
After the successful installation your database will be empty, so you can load the provided from us initial data in it as follows:

1. Download the provided folder named "forum".
2. In the console write the following command:

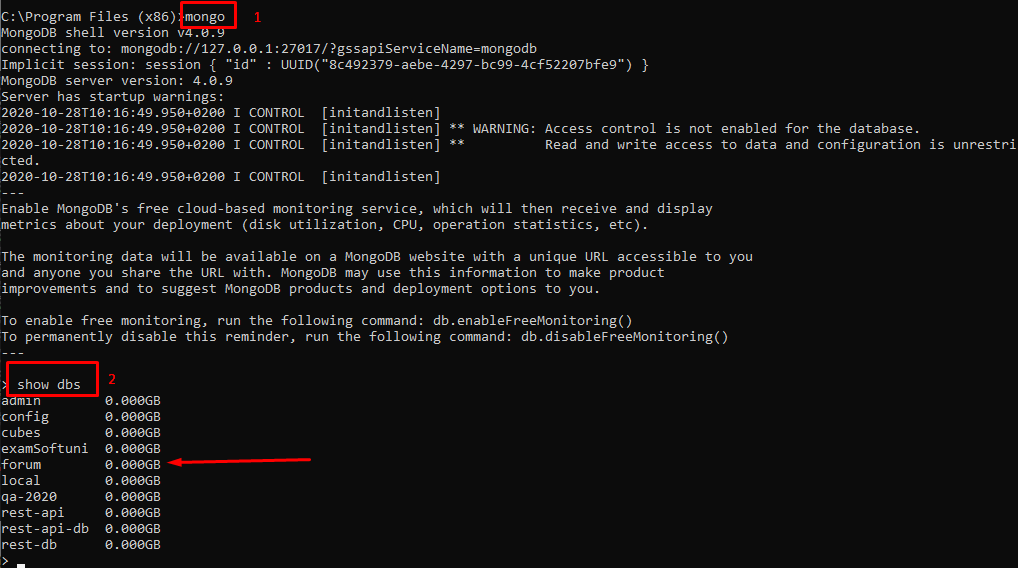
**mongorestore -d forum C:\Users\Name\Desktop\forum**

**Note\* C:\Users\Name\Desktop\forum – replace with the path of the folder "forum" in your computer!!**

On the console you will see something like this:



That's it! Now you can check if the db is in your dbs by the following commands:



## REST API

You are provided also with **REST API**. To bring life on that API you should first install the dependencies with the command "**npm install**". After that, type the command "**npm start**". In order to fetch the themes, you will have to make a **GET** **request** on "**localhost:3000/api/themes**".

## Logged/Unlogged User

**NOTE: Because we haven't covered authentication yet (how to login and register), you can hardcode it with a variable.**

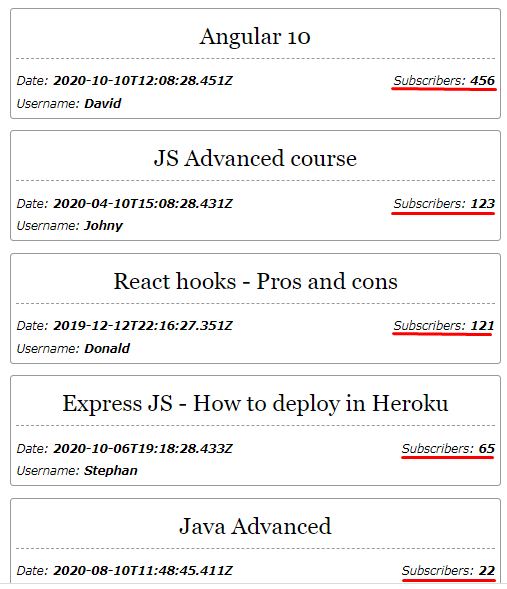
**Logged out** users see the **Login** and **Register** buttons/links.

## Implementation of the Main Section

After the backend is ready to use, you should **fetch all themes** and **list them** on the home page in the format you saw in the skeleton.

The **themes in the main** section should be **descending sorted by the subscribers**.

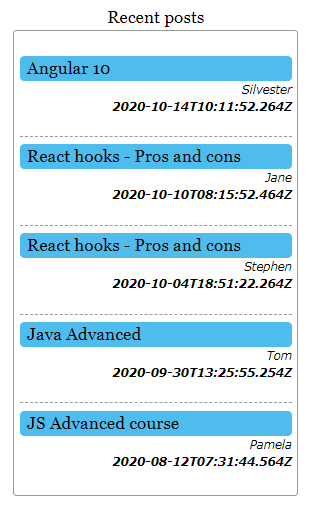
Each theme should have a **title**, **date** (when the theme is created), the **username** of the creator, **count of the subscribers** of the theme.



## Implementation of the Aside Section

In the **Recent posts section, only the 5 latest posts** must be shown! **(**[**https://localhost:3000/api/posts?limit=5**](https://localhost:3000/api/posts?limit=5)**)** The posts will be returned from the REST-api **sorted by the time of creation.**

Each component should have a **title (the title of the theme)**, a **username** of the user, whom the last comment in the theme belongs, and the **time** the post was created.



***GOOD LUCK*!**