# SoftUni Forum Workshop – Part 2

This application is created as a workshop for the Angular Couse @ Software University

# Task Requirements

So far, we have created **one** **dynamic** **page** which **lists all themes** sorted by two different criteria. The next step is to implement **few** **more** **pages** and **route** **between** **them**. You will be provided with **HTML** and **CSS** skeleton for all other additional pages that you must create. There is one catch, half of these pages require **authentication.** You must create some **properties** or **services** that **fake** this **authentication**. Because we don’t know yet how to manipulate **forms** properly or the real case with the authentication and authorization. We will fix that later when we have the knowledge to do so.

# Logged in navigation bar

When the "**user**" is "**logged in**", they should see the following navigations:

The mini nav-bar includes:

* [**Username's profile**] **a-tag** should refer to the profile page ("**localhost:4200/profile** ")
* [**Logout**] a-tag should refer to ("**localhost:4200/logout**")

The nav-bar below the logo of the SoftUni forum includes:

* [**Home**] **a-tag** should refer to the home page ("**localhost:4200/home**")
* [**Themes**] **a-tag** should refer to the "page with all themes" ("**localhost:4200/themes**")
* [**New Theme**] **a-tag** should refer to the "create theme" ("**localhost:4200/add-theme**")

Use the **isLoggedIn** signal from **AuthService** to conditionally display navigation options for logged-in users. Bind navigation visibility to the signal, so it updates reactively based on login state.



# Not logged in navigation bar

When the user is "**not logged in**" he should see the following navigations:

The mini nav-bar includes:

* [**Login**] **a-tag** should refer to the home page ("**localhost:4200/login** ")
* [**Register**] **a-tag** should refer to ("**localhost:4200/register**")



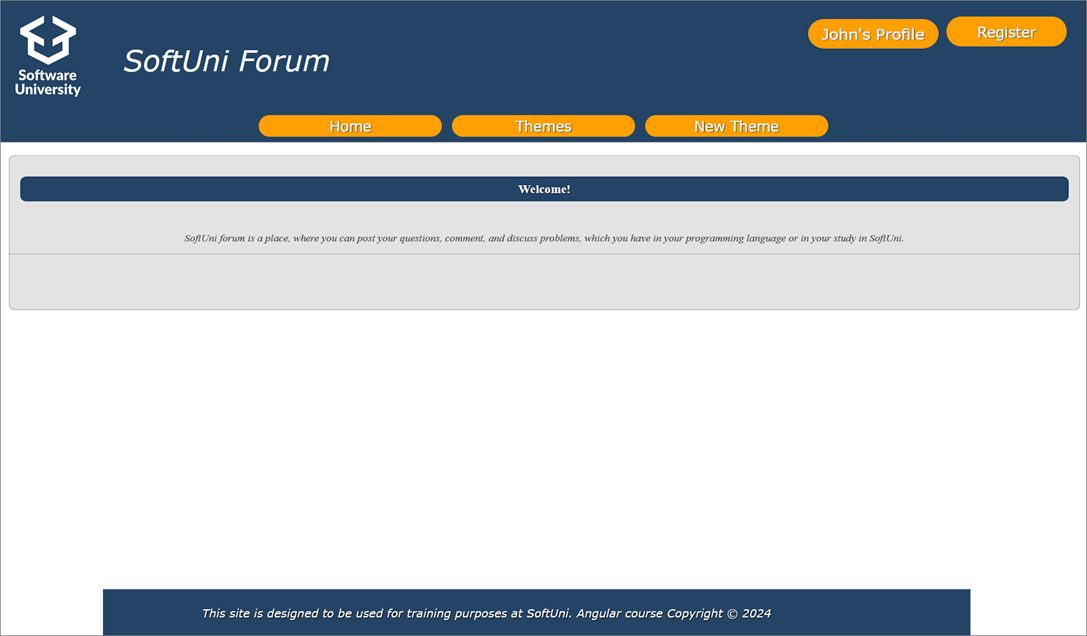
# Home page (Not Logged in)

This is the home page ("**localhost:4200/home**"). This is a welcome user page. The welcome component that we have already can be reused here. All "users" can access this page no matter of their authentication. The not logged in user view:



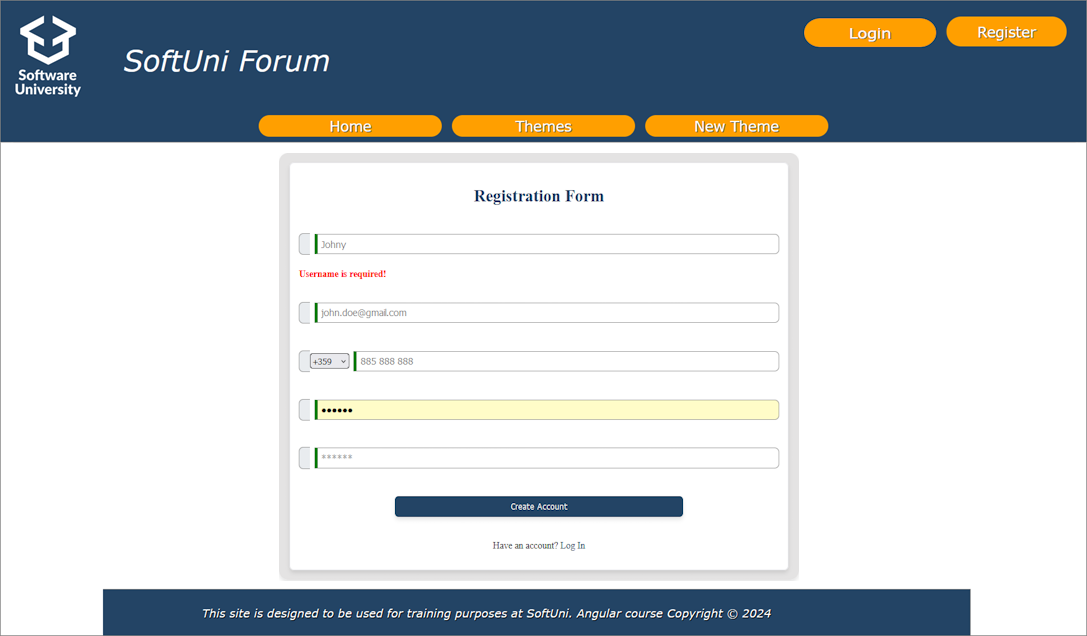
# Home page (Logged in)

The logged in user view:



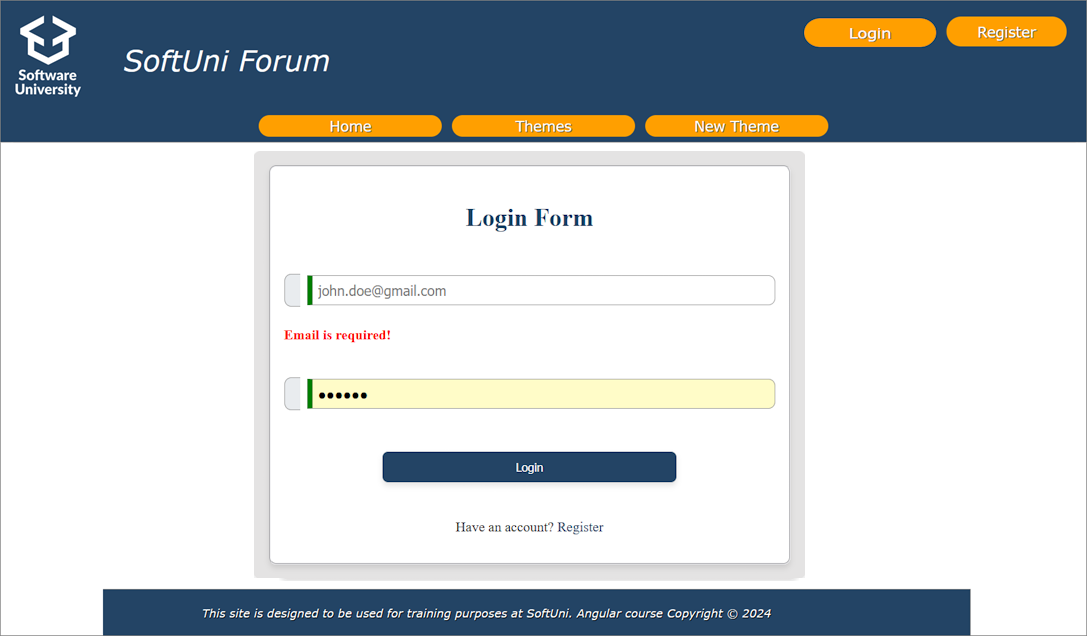
# Register (Not Logged in)

This is the register page ("**localhost:4200/register**"). Use **signals** in **AuthService** to manage the **isLoggedIn** state **reactively**. For now, we don't have a database with user registration. So, make sure **each registered** user is stored in **userService** or some other appropriate place.



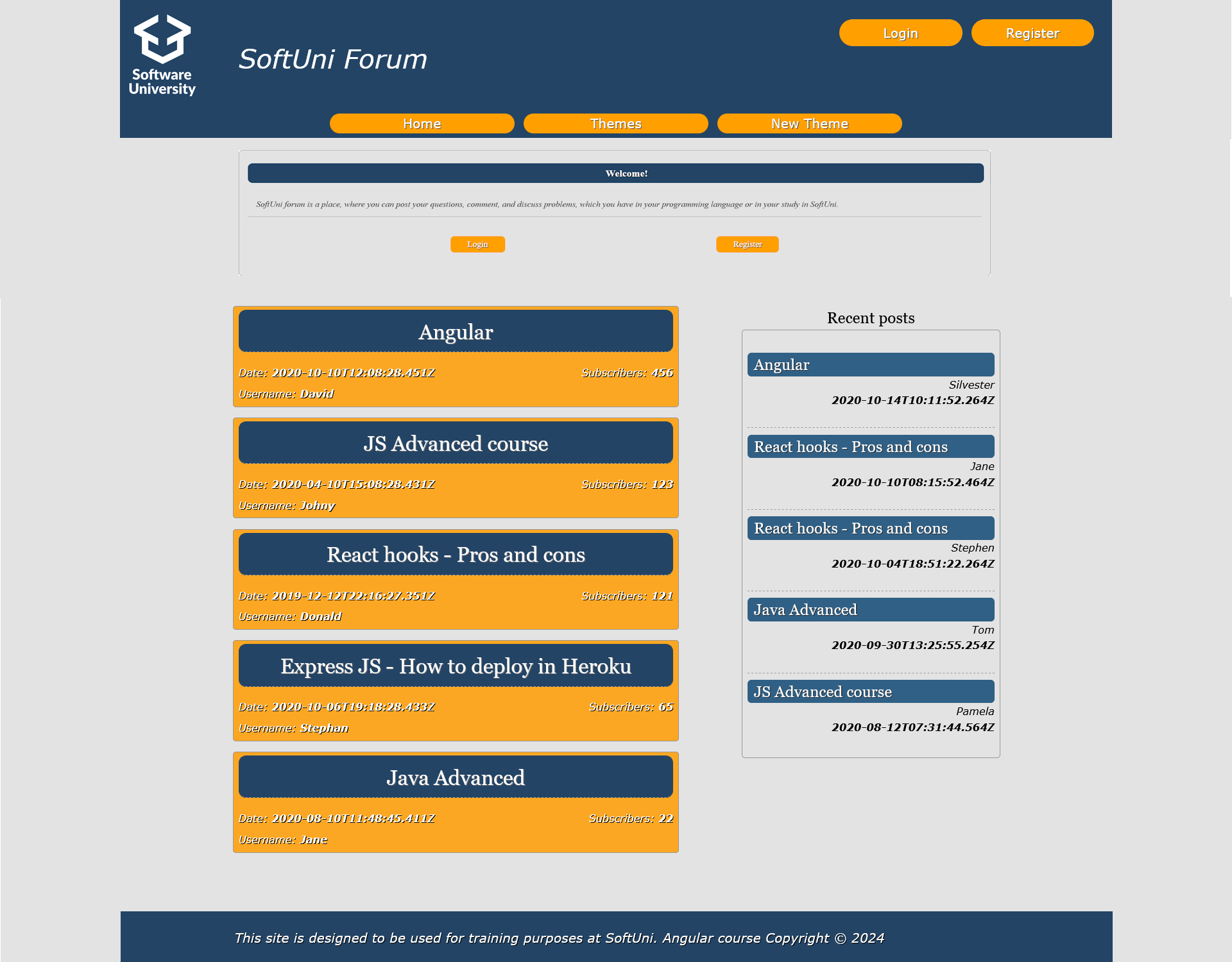
# Login (Not Logged in)

This is the login page ("**localhost:4200/login**"). This will be another **point** of your app to use **signals** in **AuthService** to manage the **isLoggedIn** state **reactively**. For now, we don't have a database with user registration. So, make sure **each registered** user is stored in **userService** or some other appropriate place. Each of them can log in successfully.

****

# Themes page (Not Logged in)

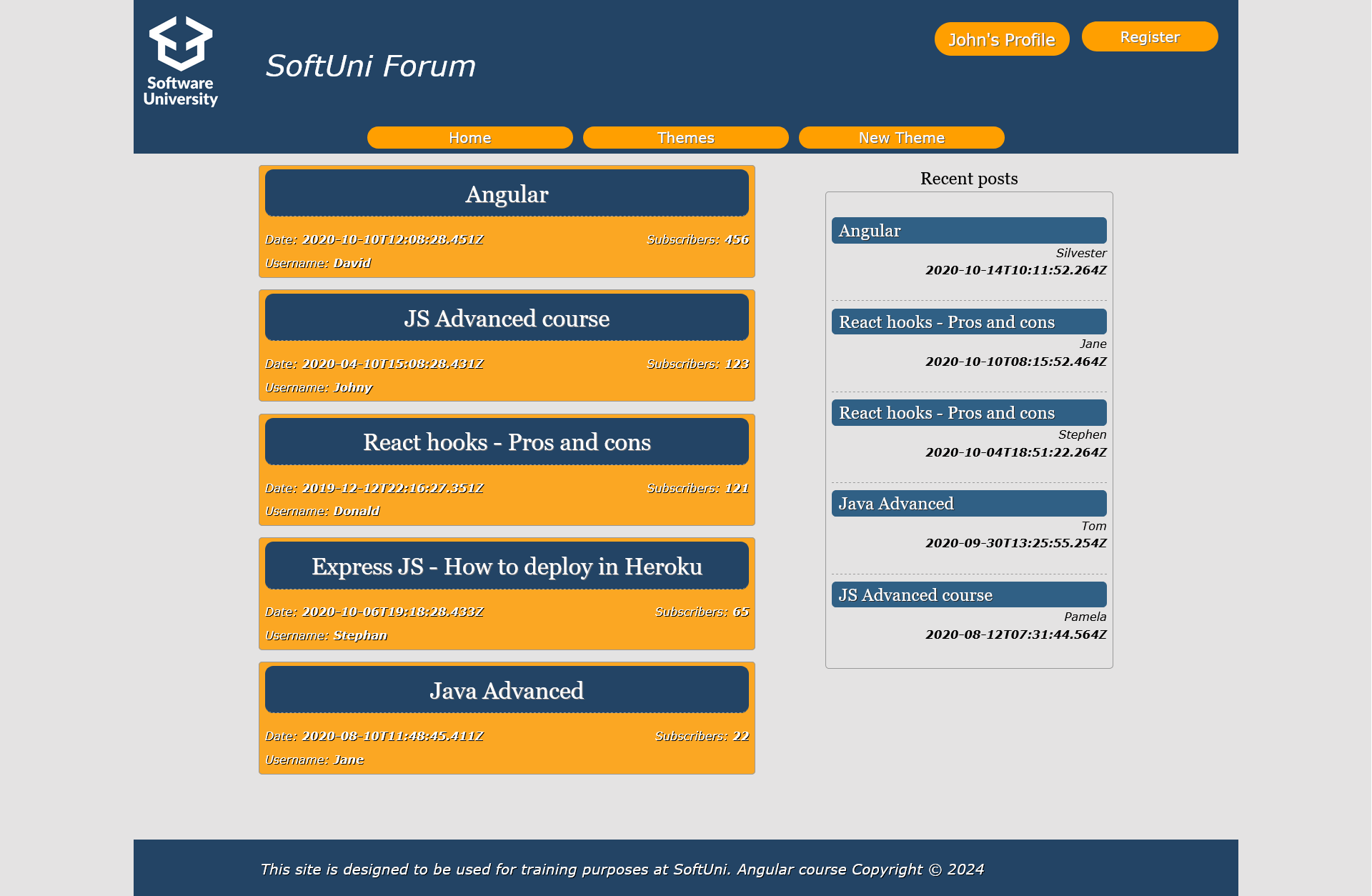
This is the themes page ("**localhost:4200/themes** "). This is the page we already have. All "users" can access this page no matter of their authentication. By clicking the title of a theme, you will be redirect to the theme content view. It is described below. There are several differences between the logged and anonymous user. The anonymous user should see the following:



# Themes page (Logged in)

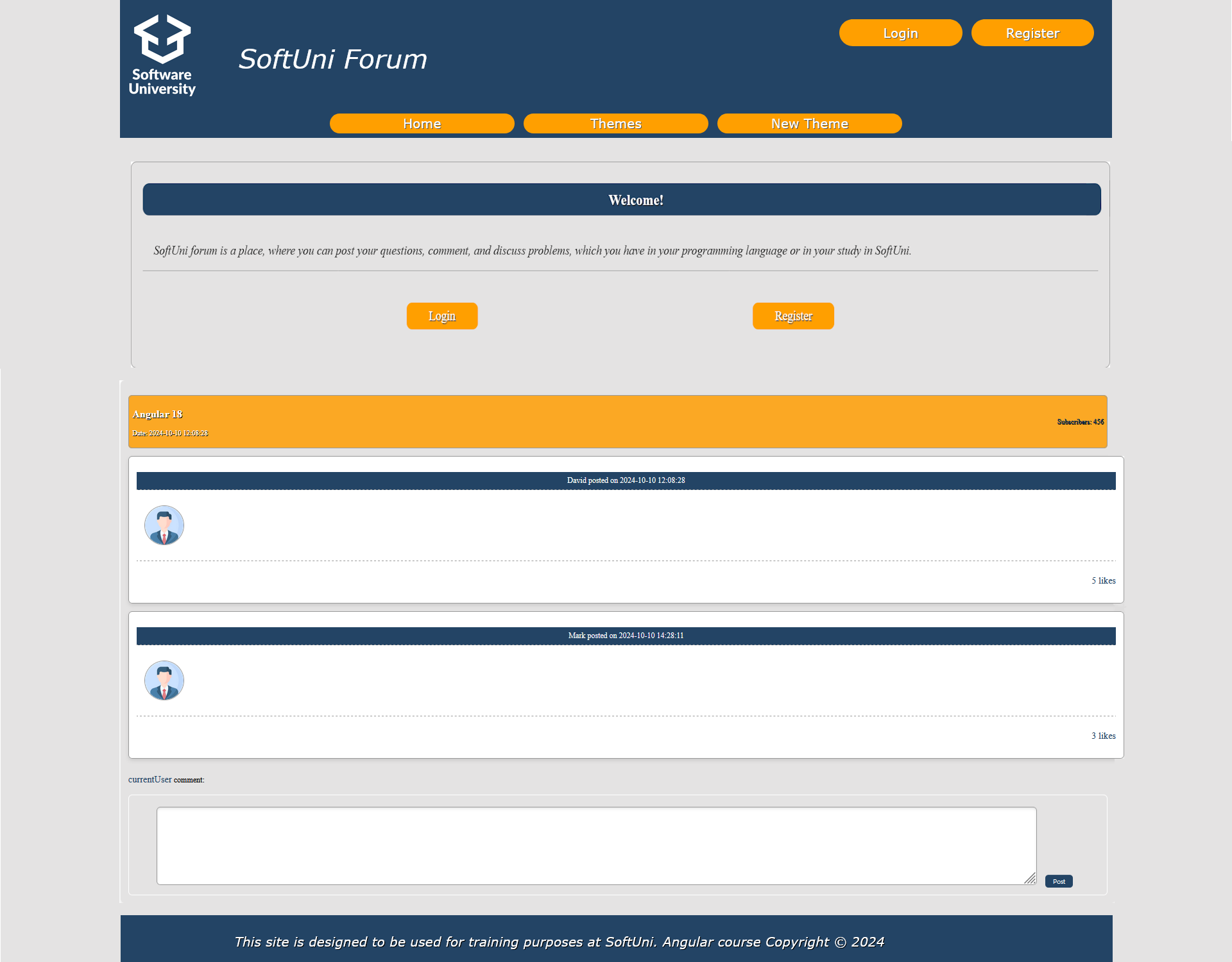
This is the themes page ("**localhost:4200/themes** ") for logged in users:

**NOTE**: Because you are still not really logged in, you can hardcode this **userId ('5fa64b162183ce1728ff371d')** in your service, to write your logic for subscribed users (**Red** and **Green** buttons).



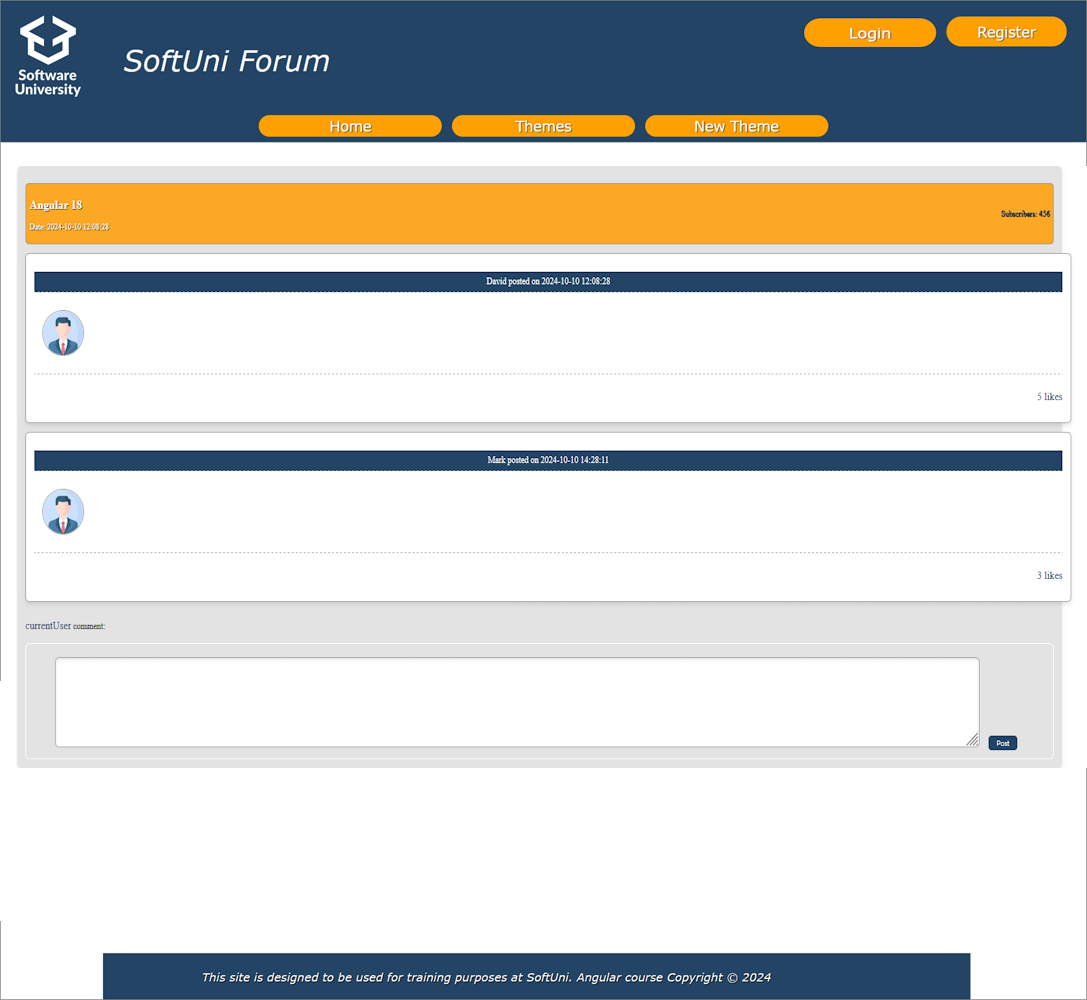
# Theme comments (Not Logged in)

This is the theme comments page ("**localhost:4200/themes/:themeId** "), where the user can see all the posts for the selected theme.



# Theme comments (Logged in)

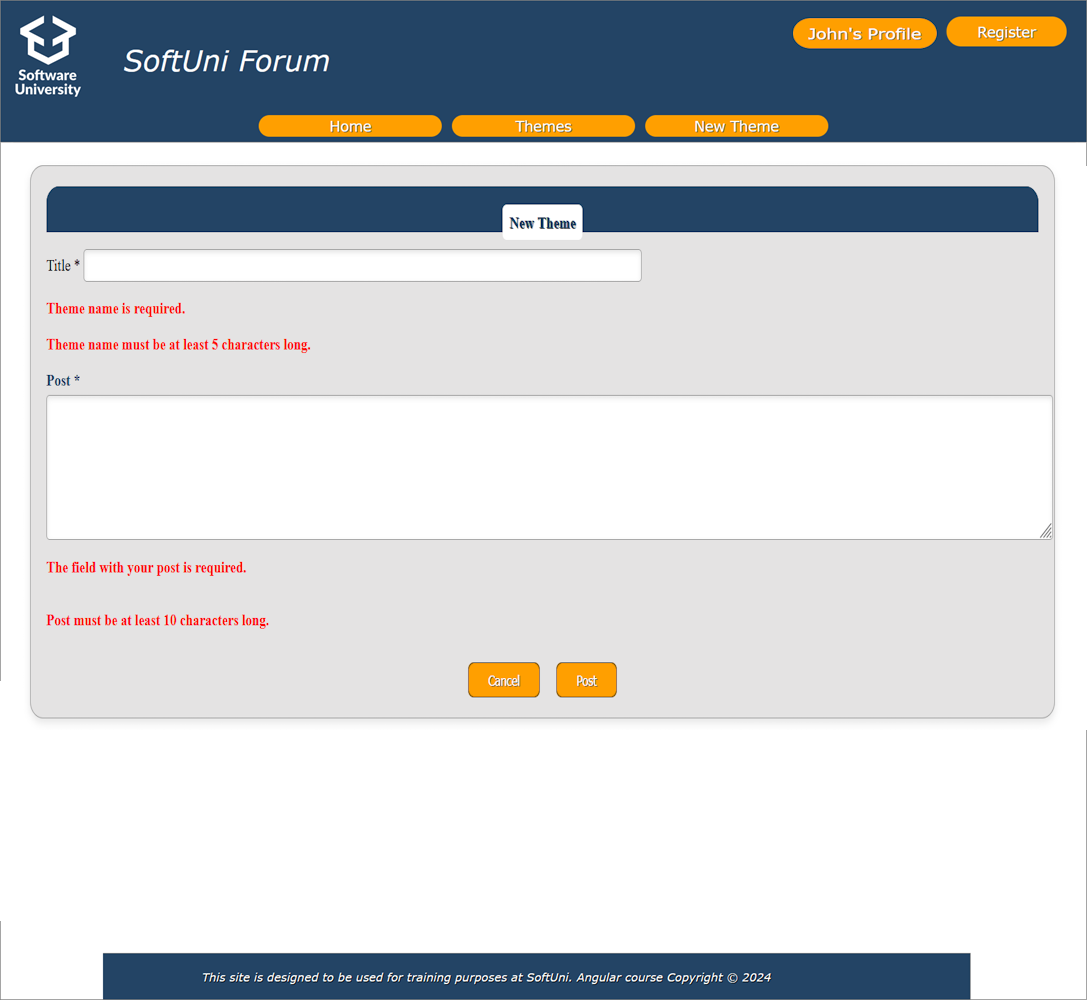
The logged in user can **write new comment** or **like** the other user's posts.



# Create New Theme (Logged in only)

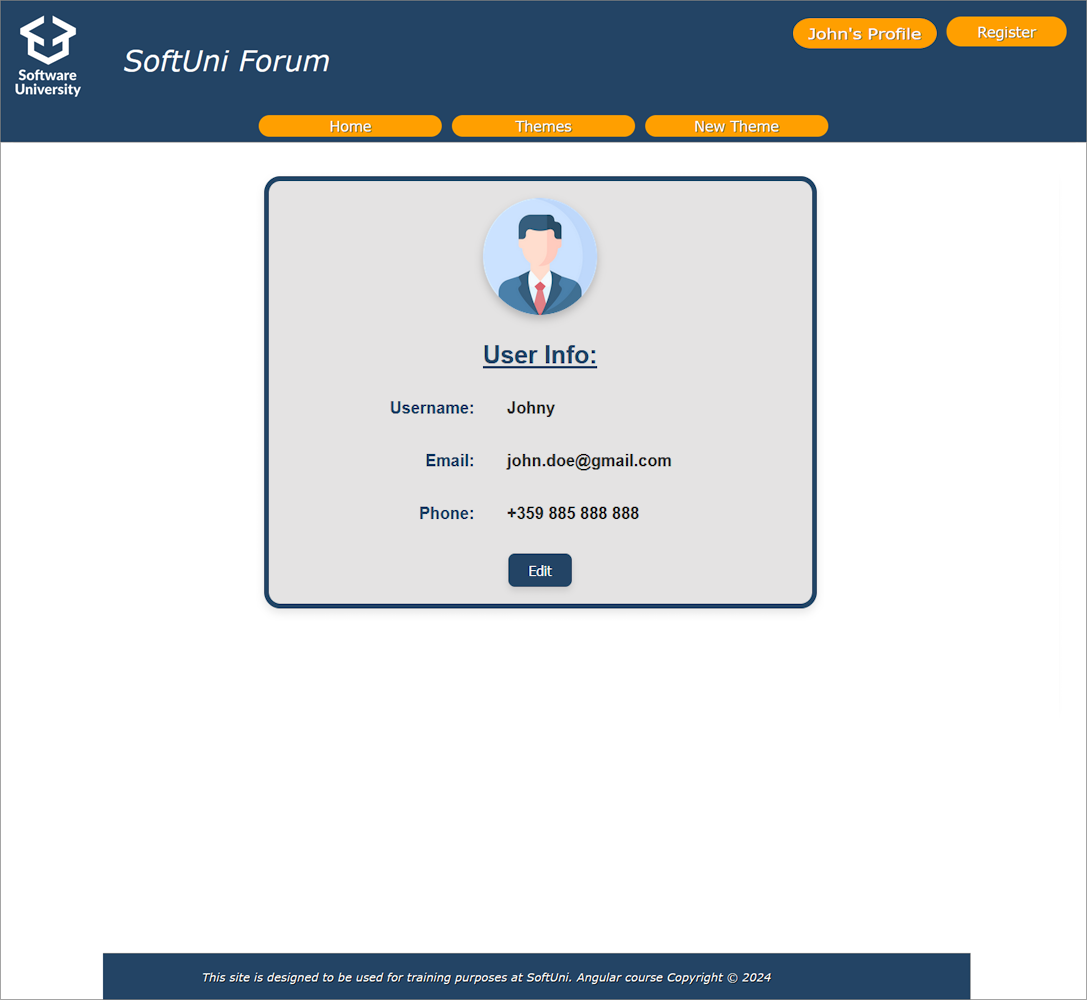
This is the create new theme page ("**localhost:4200/themes** ") where each **user** (logged in) can create his own theme.

* When "**Post**" button is clicked, you can try to make **"POST" request** to ("**localhost:4200/themes**") with the given theme information. After successful creation, **redirect** the current "user" to the **theme comments page** of his new created theme.
* When "**Cancel**" button is clicked, **redirect** the user to **home** page.



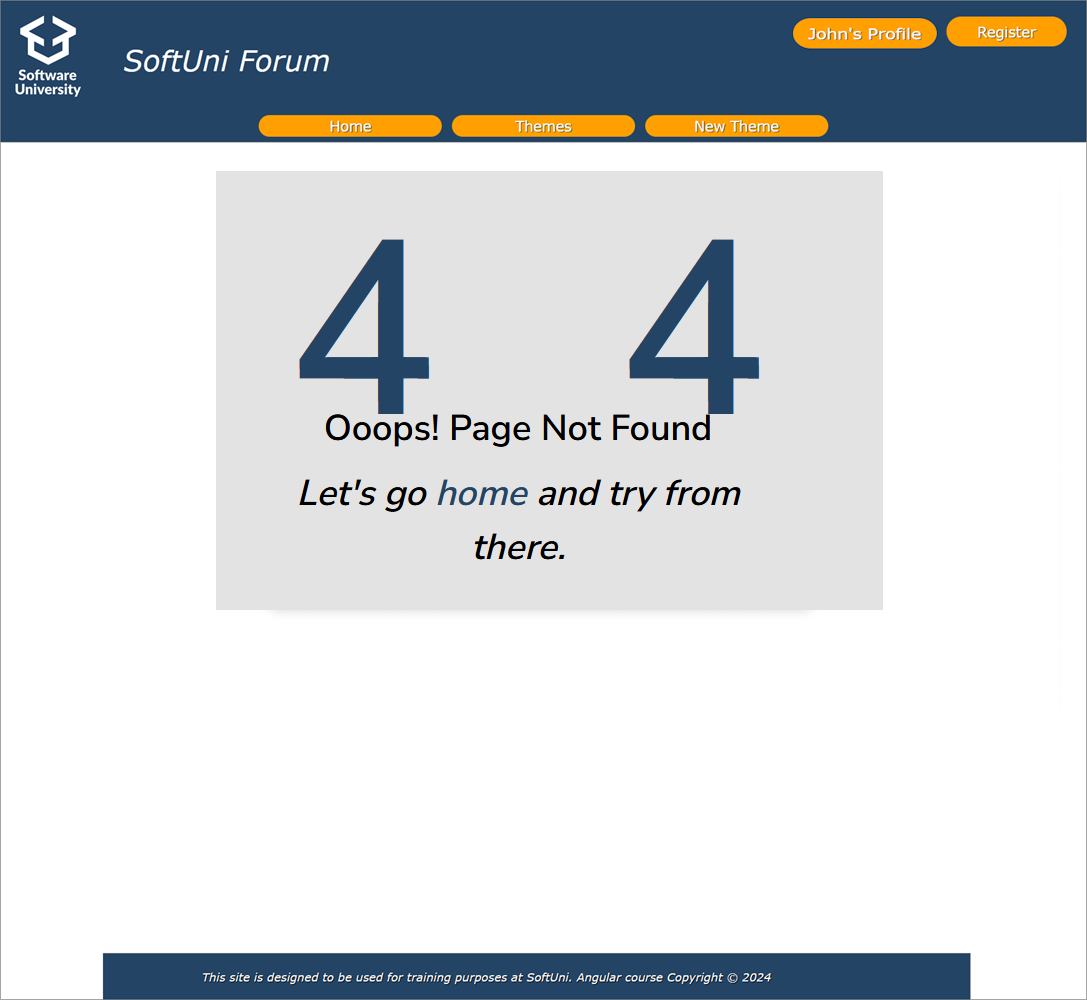
# Profile (Logged in)

This is the profile page ("**localhost:4200/profile**"). This page will show the information about the currently logged in user. For now, the data in this page will be **static**, except if you create more than the **fake** **isLoggedIn** **property**. The "Edit" button will replace the information fields with input fields, but this will be made in the next workshop, when you learn more about handling forms.



# Invalid routes

This is the page for all **invalid** **routes** ("**localhost:4200/??????** "). Use it if invalid path or wrong one is accessed.



# Protected routes

Set up lazy loading for feature modules (e.g., Themes, Auth, and Home) using routing. Directly reference standalone components in routes without declaring them in modules, where applicable. Protect logged-in routes with guards.

***GOOD LUCK*! 😊**