# Mod Panel – 26 October 2017

The mod panel is a place in which users, can post about their work. Your task is to create a web application that has back-end and front-end logic using the provided resources (**Web Server**, **MVC Framework**, **Bootstrap** and views).

## Data Models

Create the required **entities**. Use the appropriate **data types**.

* The system contains information about **users**, **posts** and **admin logs**
* Users can **register** in the system. After successful registration, an admin must approve the user. The user has **email, password, list of posts** and **position**
* A **post** has **title** and **content**
* A **log** has **admin** and **activity**

## Functionality

* **Guests** (anonymous users) can register an account with their own email and password
* **Guests** can login by email and password.
* **Guests** can view the default home page including links to **login** and **register**
* **Logged-in users** can logout.
* **Logged-in users** can **create** **posts**
* **Admins** can **edit** and **delete posts**
* Each **admin activity** mustbe **logged**
* **Admins** canreviewthe **log**
* Admins can **approve** **users**
* When guest user tries to access a page that is allowed only for logged in user he should be **redirected to the login page**
* When **user** tries to access **admin** page**,** he should be **redirected to the login page**

## Design the Database

Design **entity classes** and create a **database** to hold the **users**, **posts** and **logs**

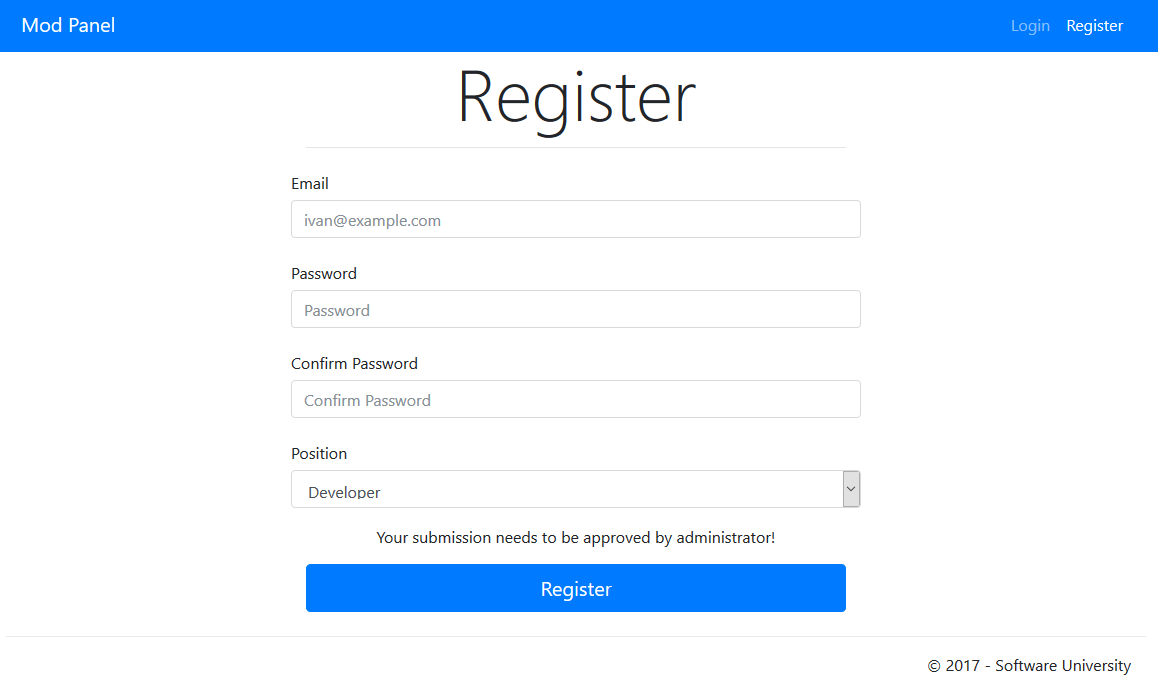
Score: **10** **points**

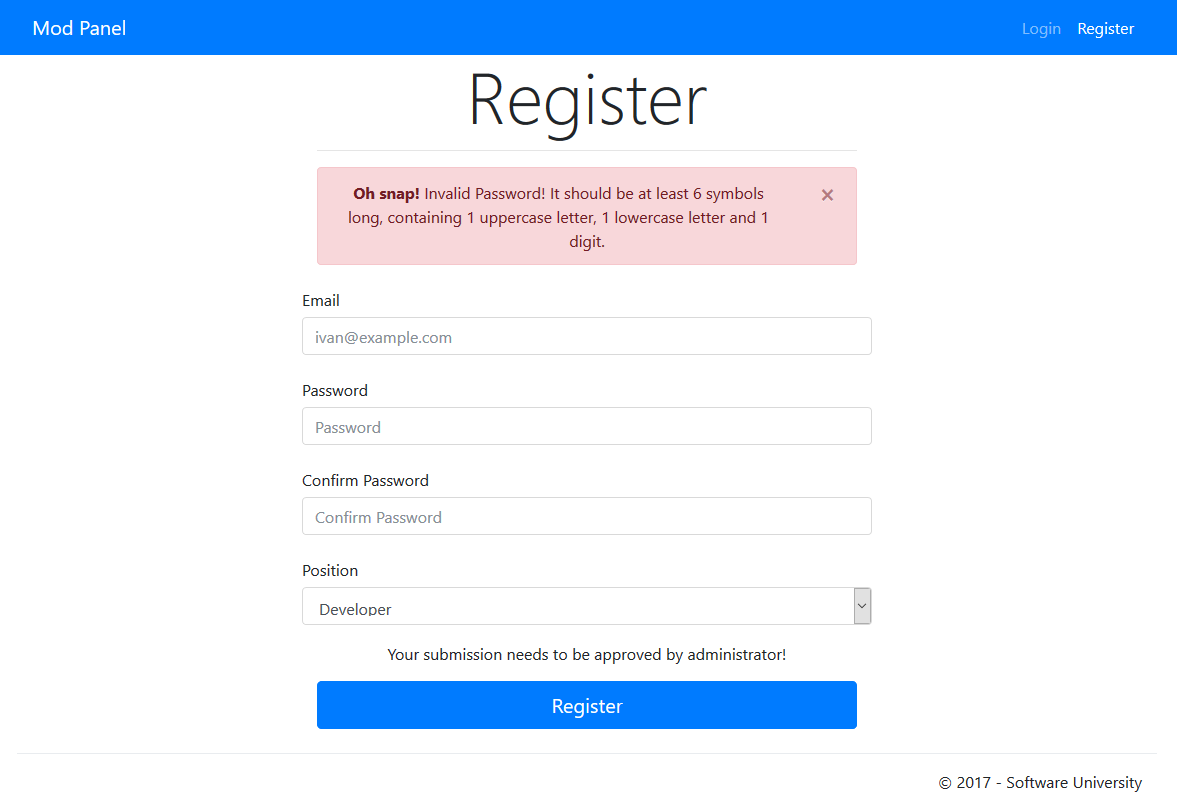
## Implement User Registration, Login and Logout

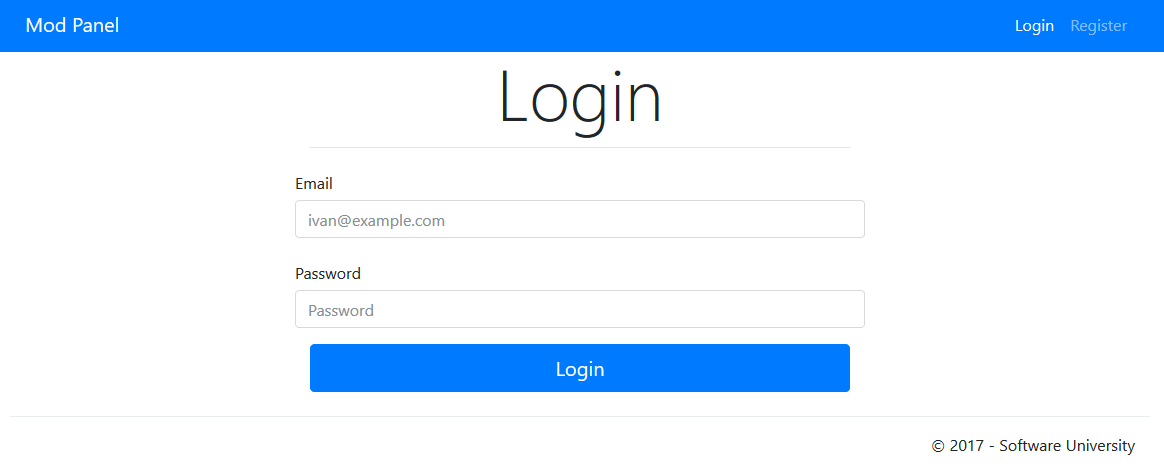
The guest users can register and login. He must enter the following:

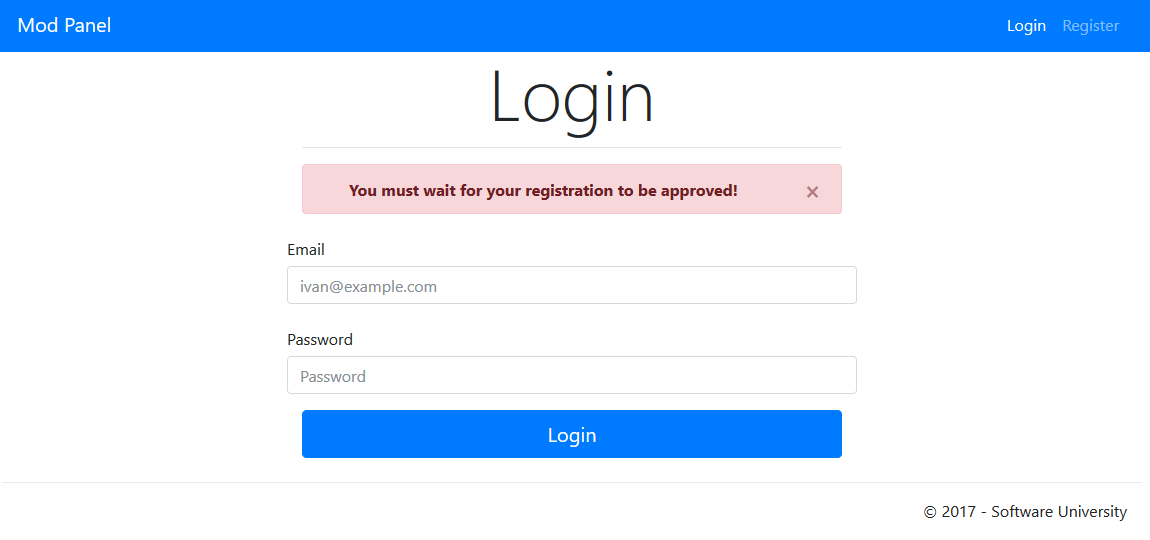
* **Register**
  + **Email** – must contain **@** **sign** and a **period**. It must be unique.
  + **Password** – length must be **at least 6 symbols** and must contain at least **1 uppercase**, **1 lowercase** letter and **1 digit**
  + **Confirm Password** – must **match** the provided password
  + **Position** – one of the following
    - Developer
    - Designer
    - Technical Support
    - Technical Trainer
    - HR
    - Marketing Specialist
* **Login**
  + User can log in with **email** and **password**
* **Logout**
  + When logged in the user should have option to **log out**

After register, the user should be **approved**. The **first user** to **register** becomes an **admin**. If there are **any errors** with the registration, they **should be** **displayed** in the register form. All validations should be performed both on the back-end and front-end.









Each type of user must have **unique navigation bar**.

Guest:

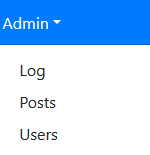


User:



Admin:

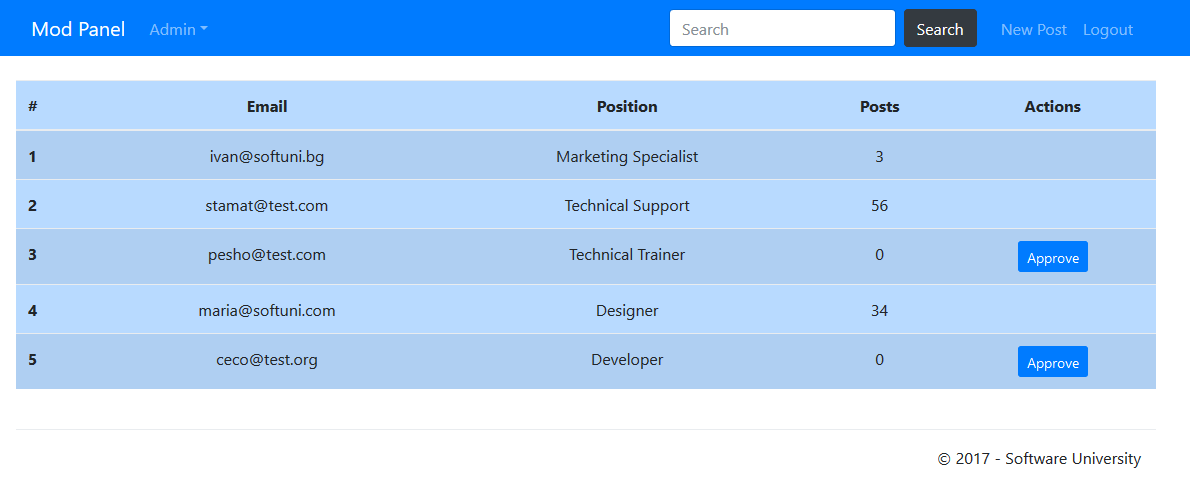




Score: **15** **points**

## Implement User Approval

Every **user** should be approved. The admins can **approve users** via the dropdown:



Score: **5** **points**

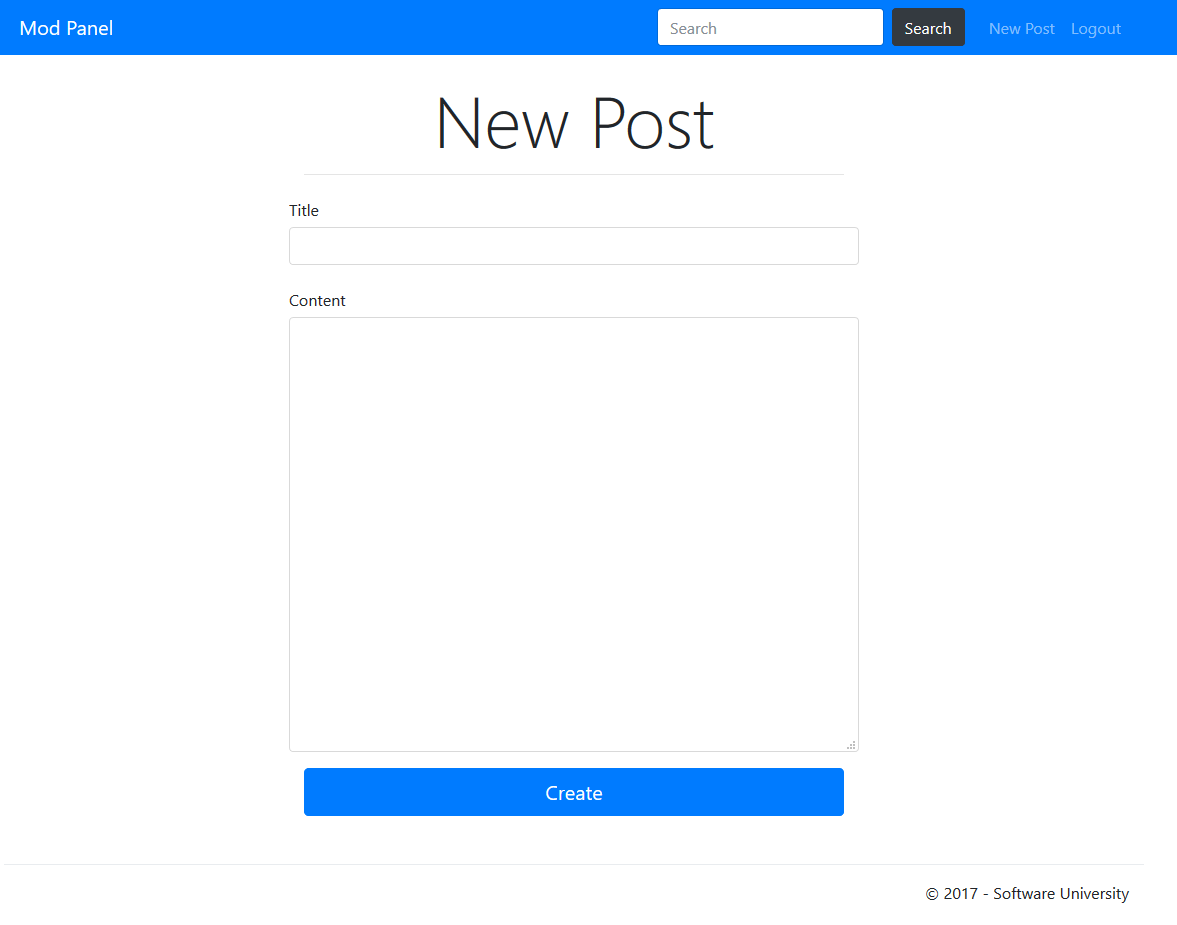
## Implement Posts

As a user, you have the option to **create posts**. A post should be added only if it matches those criteria:

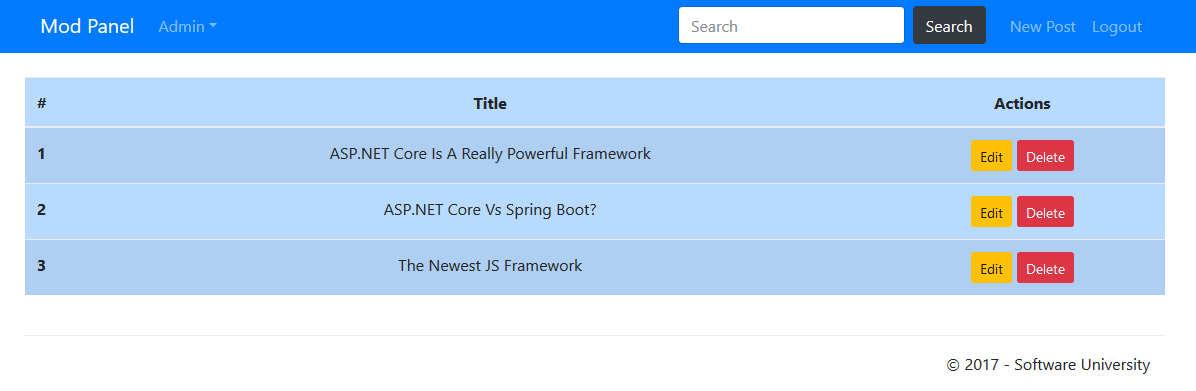
* **Title** – must begin with **uppercase letter** and has length between **3 and 100 symbols** (inclusive)

All validations must be performed on the back-end.

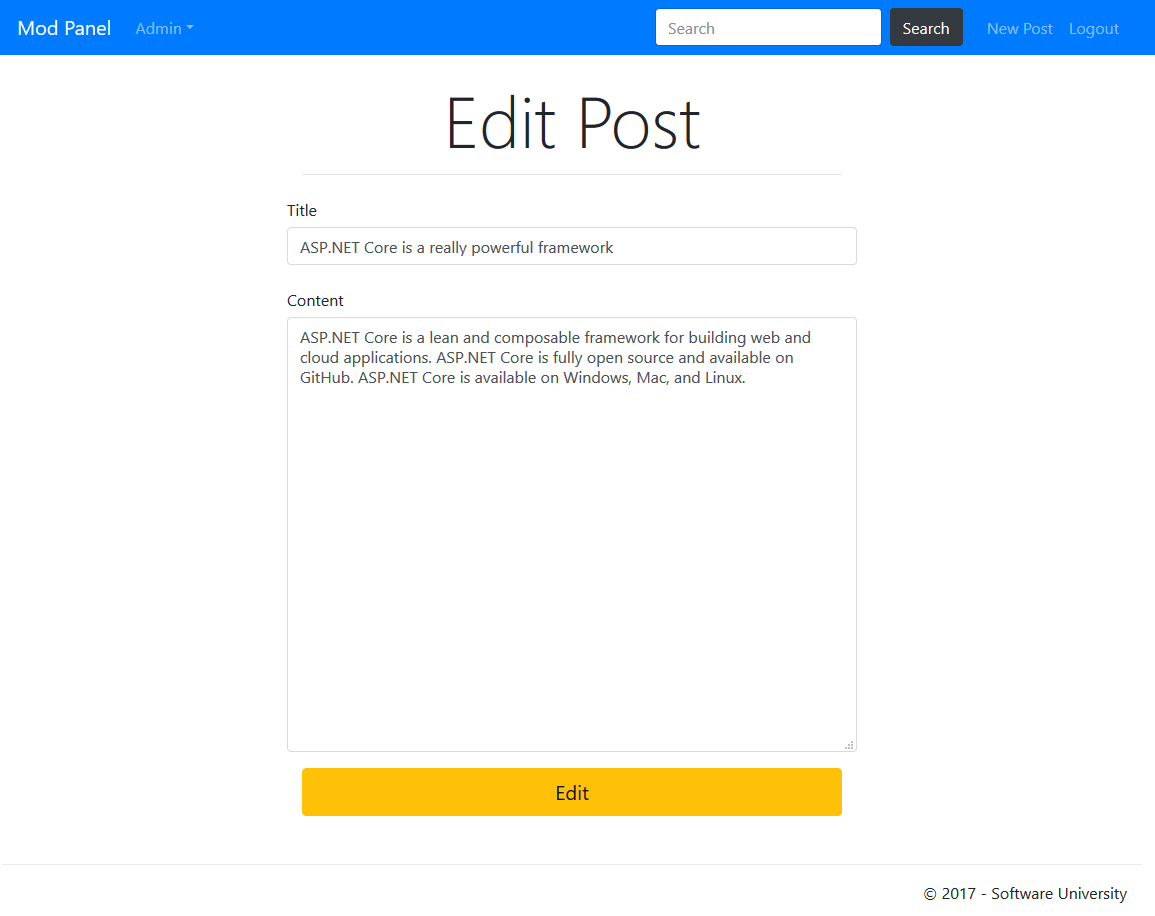
Adding a post:



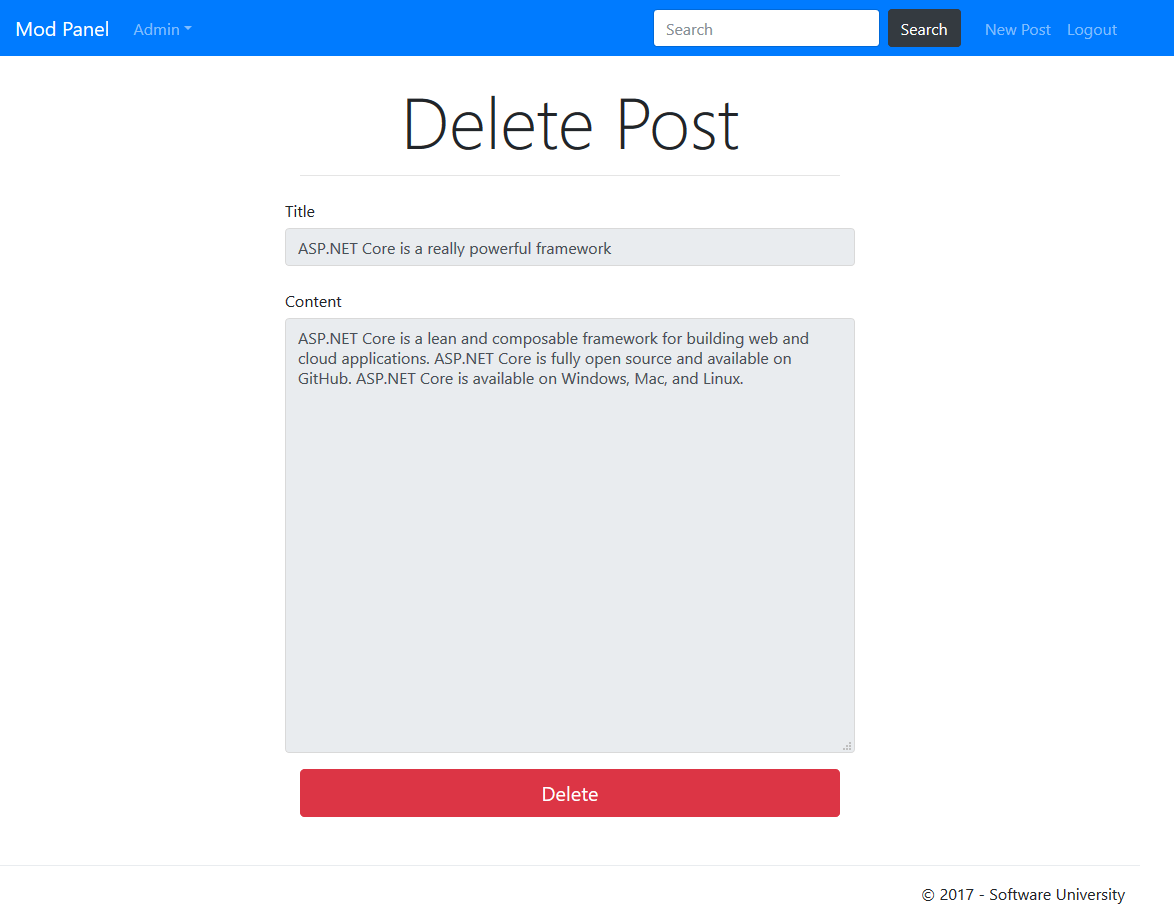
**Admins** should be able to see a table with all posts, using the admin drop-down:



**Editing** **post**:



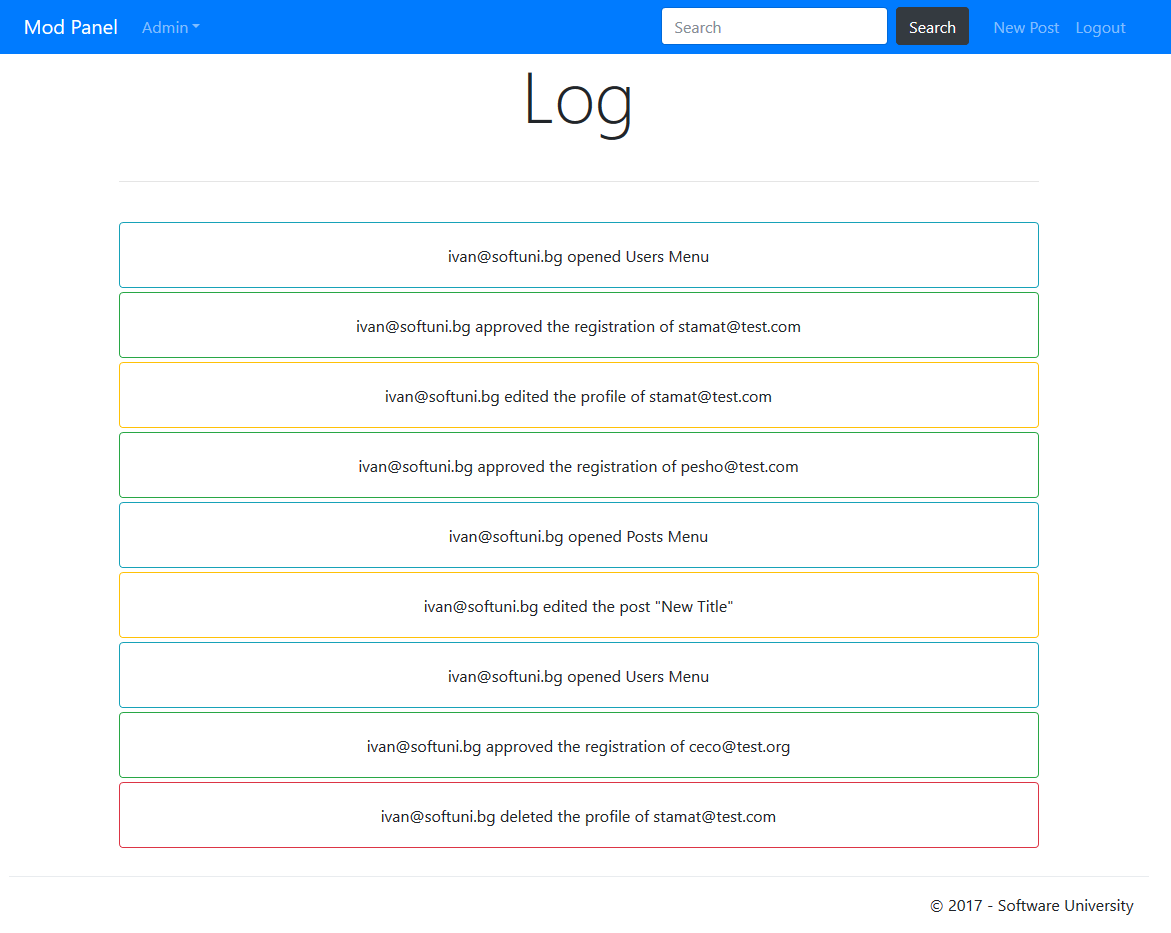
**Deleting** **post**:



Score: **15** **points**

## Implement Admin Log

Every **admin activity** should be automatically logged. The admins can **view the log** via the dropdown:



The messages for different activities are as following:

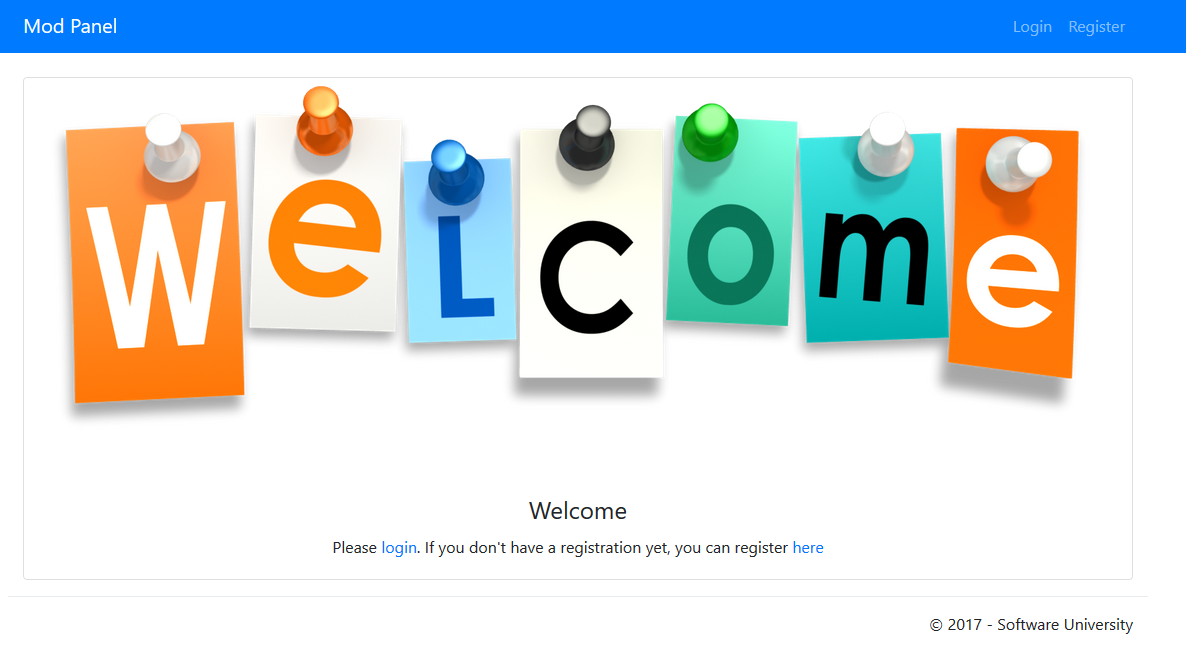
* {admin} created/edited/deleted the post "{post title}"
* {admin} approved the registration of {user}
* {admin} edited/deleted the profile of {user}
* {admin} opened **Posts**/**Users**/**Log** Menu

Score: **15** **points**

## Implement Guest Home Page

It should show a message telling the user to login or register.

**Guest** home page:

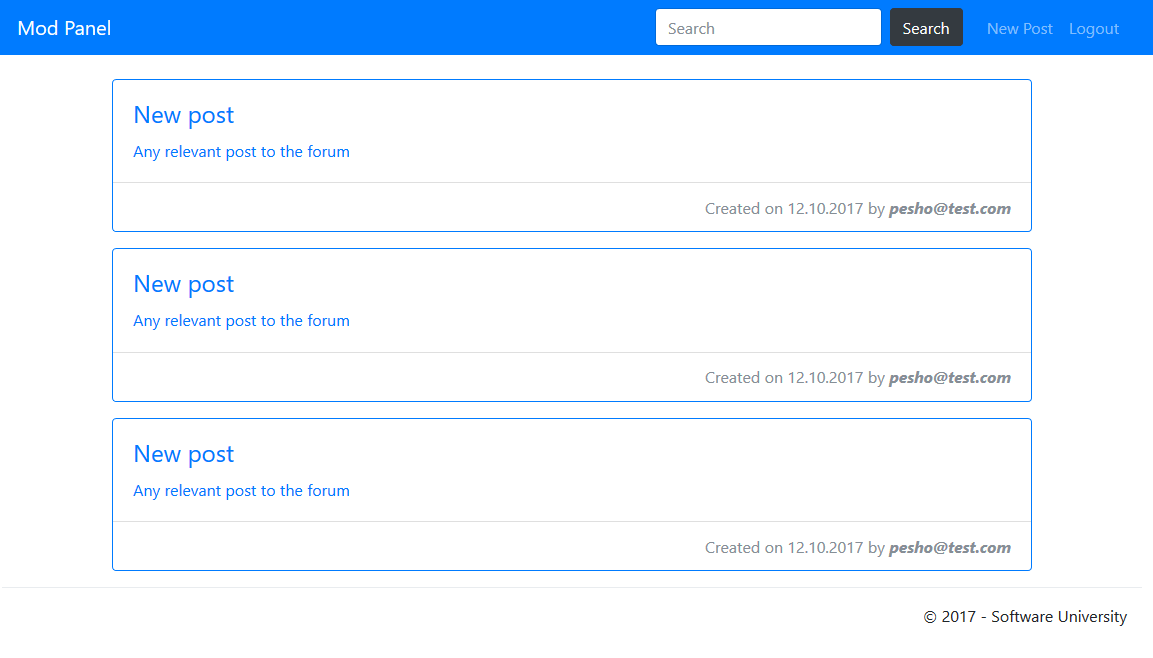


Score: **5** **points**

## Implement User Home Page

It should show all posts sorted by date. The newest post should be on top.

**User** home page:

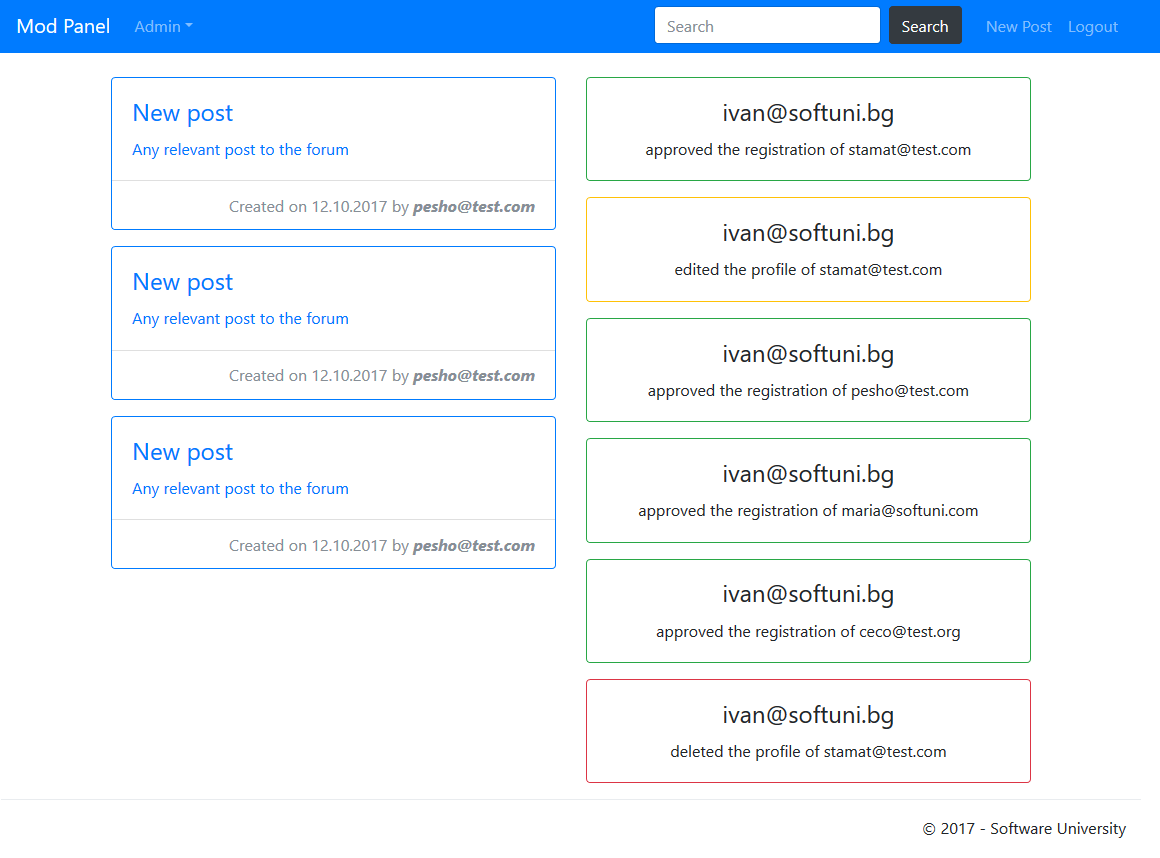


Score: **5** **points**

## Implement Admin Home Page

It should show all posts sorted by date and the log activity side by side. The newest posts and logs should be on top.

**Admin** home page:



Score: **10** **points**

## Implement Search Engine

As you've probably noticed there is a **search** in the navigation bar. It should search **posts** by **title**. Use the **user home** page as a **template** for the result set.

Score: **10** **points**

## Project Infrastructure

Points for code quality / good application structure / additional effort.

* Points for implementing **separate data layer**.
* Points for using **dependency inversion**.
* Points for using **AutoMapper**.
* Points for using **services** for different jobs.

Score: **10 points**