# Issue Tacker – 28 February 2017

Issue Tracker is a system that reports bugs. Your task is to build one. You have to use Your task is to create a web application that has back-end and front-end logic using the provided resources (HTTP Server, MVC Framework, HTML and CSS files). You are free to edit those provided resources to fit your needs.

## Data Models

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

* The system contains information about **users** and **issues**
* Users can **register** in the system. After successful registration, the user has **username, full name**, passwordand role (regular user or administrator).
* The **first registered user** becomes also an **administrator**.
* An issue has **name**, **priority**, **status**, **creation date** and **author**
* Issue **priority** can be **low, medium** or **high**
* Issue **status** can be **new** or **solved**

## Functionality

* **Guests** (anonymous users) can register an account with their own username, full name and password.
* **Guests** can login by username and password.
* **Logged-in users** can logout.
* **Guests** can view the home page.
* **Logged-in users** can create new issues.
* **Logged-in users** can edit or delete their own issues.
* **Administrators** can add, edit or delete all issues
* When guest user tries to access a page that is allowed only for logged in user he should be redirected to the login page
* All pages should be accessible **only** for logged in users **except home, register and login pages**.
* If a user tries to access any other page and it is **not logged in**, he should be **redirected** to the login page

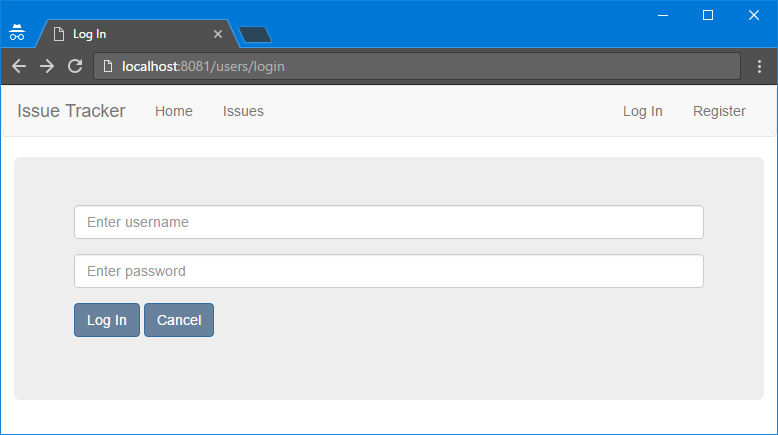
## Design the Database

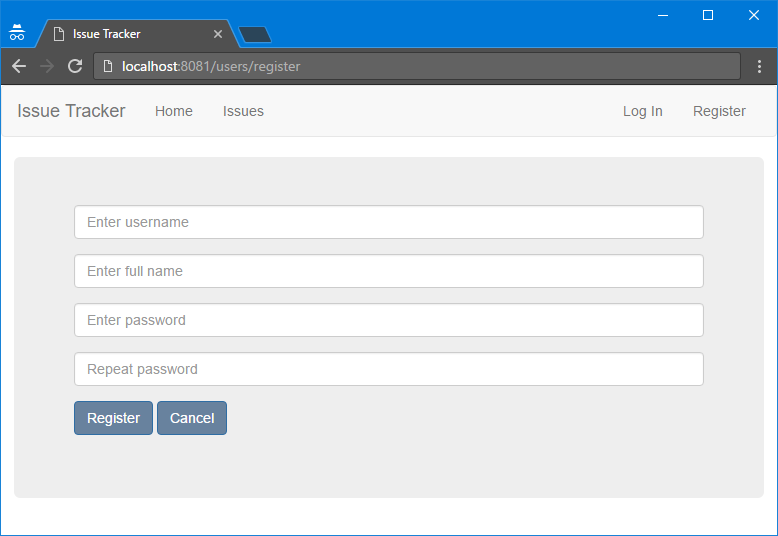
Design **entity classes** and create a **database** to hold the **users** and the **issues**.

## Implement User Registration, Login and Logout

The guest users can register and log in the forum. He can provide to:

* **Register**
  + **Username** – must be between **5** and **30** symbols
  + **Full Name** - must be at least **5** symbols
  + **Password** – at least **8** symbols. It should contain a **capital letter**, a **number** and one of the following **signs: [!@#$%^&\*,.]**
  + **Confirm Password** – must **match** the provided password
* **Login**
  + User can log in with **username and password**
* **Logout**
  + When logged in the user should have option to **log out**

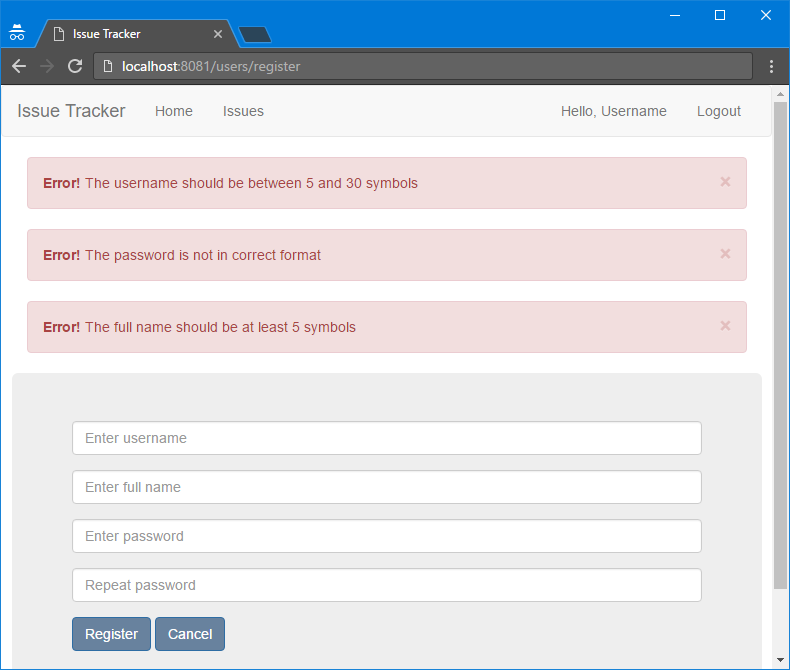


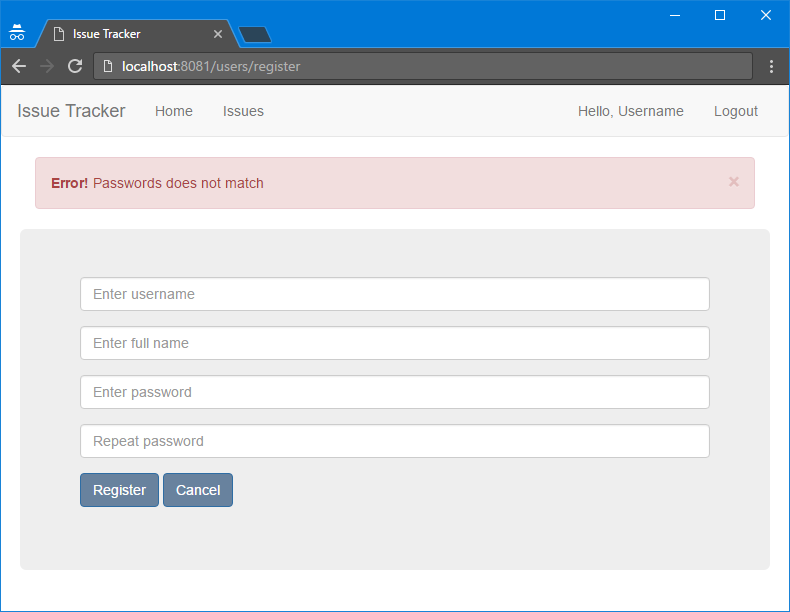




## Implement Validations

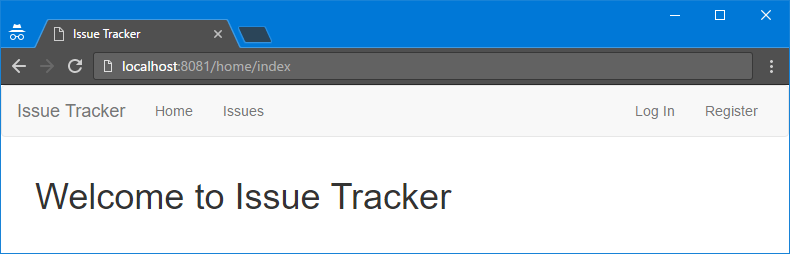
If any of the constraints fails **alert** the user with appropriate message. Implement validations for both register and login forms.

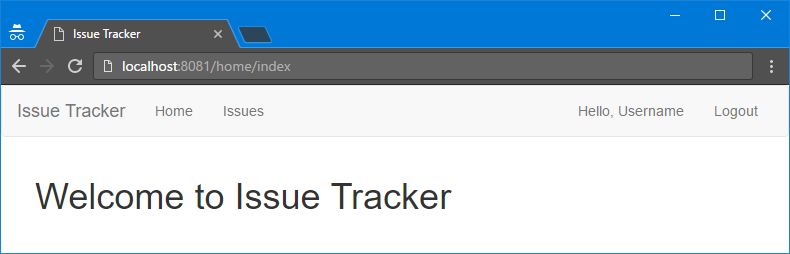




## Implement Home Page

The home page is available to **guests** and should have a small **menu**. If the user is logged in a **logout** option should appear.

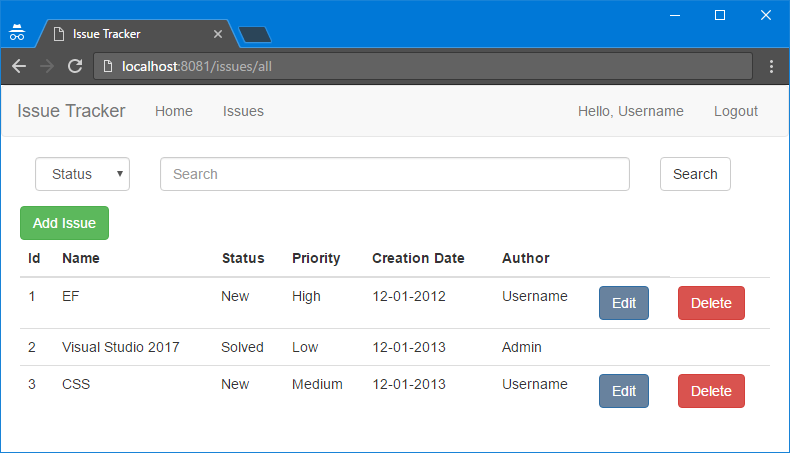




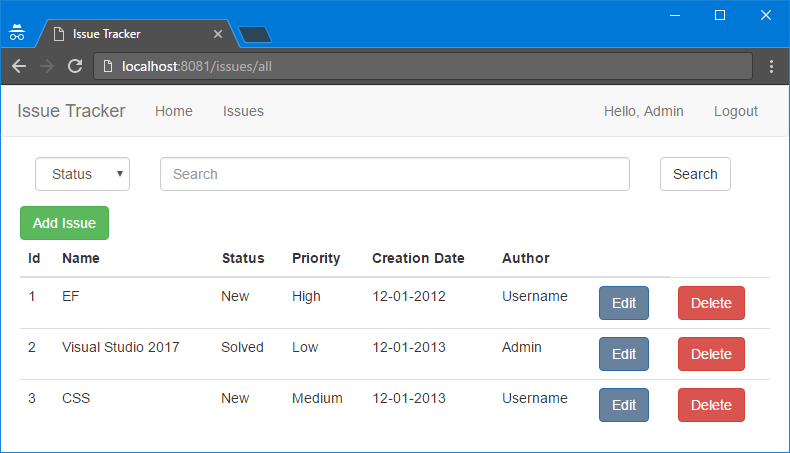
## Implement Issues Page

Here all **the issues** are listed. If you are the **owner of the issue** you can edit and delete your own issues. If you are **admin** you can edit and delete any issue. Logged in users can create issues. Implement the **search functionality**. You can look up issues by name and their status.

**User View**

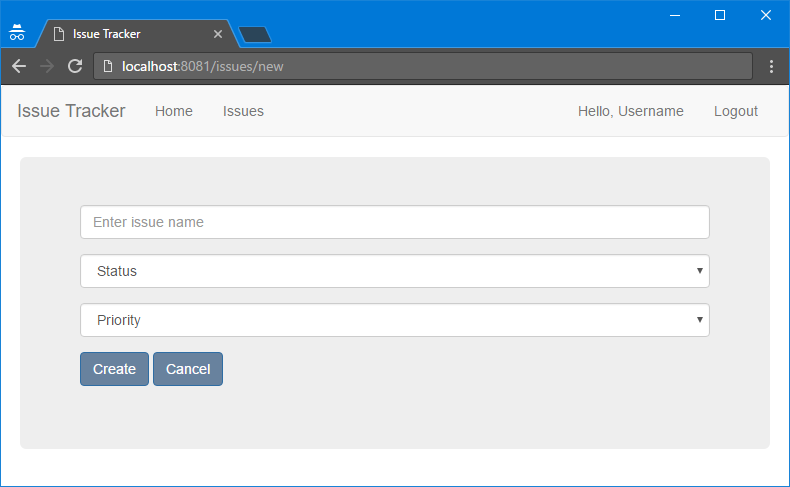


**Admin View**



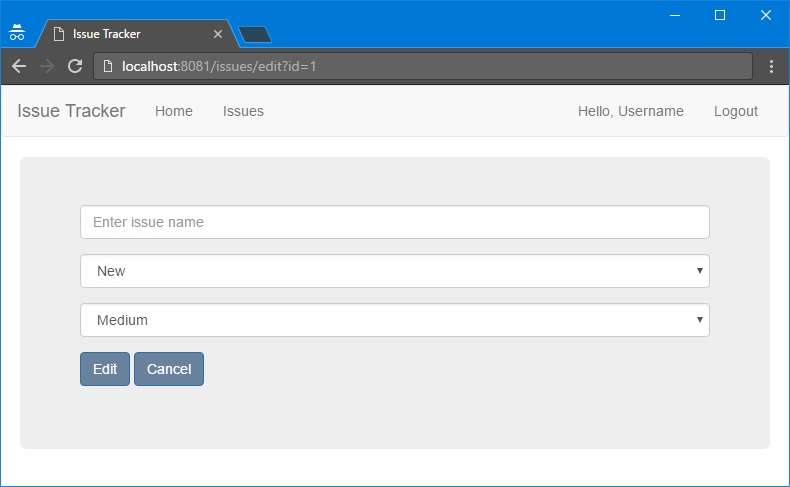
## Implement Add Issue Page

On that page **a new issue** should be added.



## Implement Edit Issue Page

On that page, **any current issue** should be edited.



## Implement Delete Issue Functionality

You should be able to **delete** an issue if that is **permitted**.

## Project Infrastructure Bonus

Bonus points code quality / good application structure / additional effort.

* Bonus points for implementing **separate data layer**.
* Bonus points for using **dependency inversion**.
* Bonus points for using **AutoMapper**.
* Bonus points for using **services** for different jobs.
* Bonus points for implementing **Repository** or **Unit of Work** pattern