# Pizza Forum – 27.02.2017

Since we all love pizza so much it will be cool to have some place to share our thoughts on pizzas. That would be the Pizza Forum. 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 Model

* The system contains information about **users**, **categories**, **topics** and **replies**
* Users can **register** in the system. After successful registration, the user has **username, email**, passwordand information whether he **is an administrator or not**. The username and email must be unique – there cannot be two users with the same username or same email. Users also have a **list of topics** that he created.
* The **first registered user** becomes also an **administrator**.
* **Categories** have **name** and **list of topics**.
* **Topics** have **title**, **author** (user), **content**, **publish date** and **list of replies**
* **Replies** have **content**, **author**, **publish date**, **image URL**

## Functionality

* **Guests** (anonymous users) can register an account with their own username, email and password.
* **Guests** can login by username or email and password.
* **Logged-in users** can logout.
* **Guests** can view the home page. It holds latest 10 created topics
* **Logged-in users** can create new topics in provided categories
* **Logged-in users** can edit or delete their own topics
* **Logged-in users** can post replies to all topics. If they want optionally can attach an image to the reply via URL. If image URL is present to the reply it must be visualized under the content of the reply
* **Administrators** can add, edit or delete categories
* When guest user tries to access a page that is allowed only for logged in user he should be redirected to the login page

## Design the Database

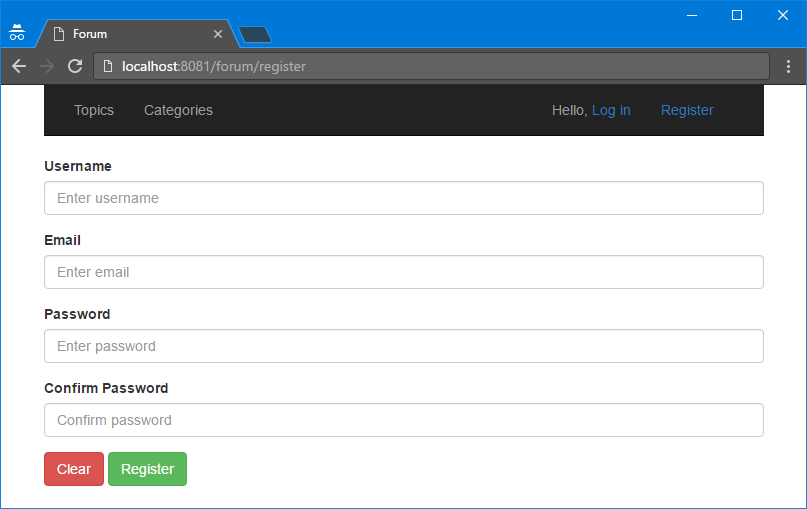
Design **entity classes** and create a **database** to hold the **users**, **categories**, **topics** and **replies**. Use Entity Framework code-first.

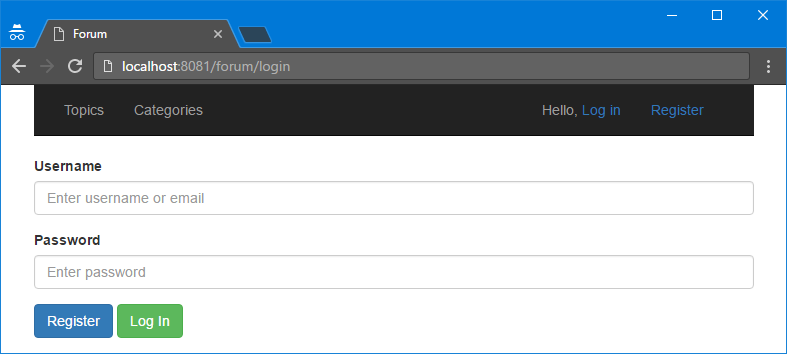
## Implement User Registration, Login and Logout

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

* **Register**
  + **Username** – must be **at least 3 symbols long** and can contain **only lowercase letters and numbers**
  + **Email** – must contain **@** sign
  + **Password** – must be at **exactly 4 symbols** long and can contain **only digits**
  + **Confirm Password** – must **match** the provided password
* **Login**
  + User can log in with **username or email** **and password**
* **Logout**
  + When logged in the user should have option to **log out**

[image of register form]

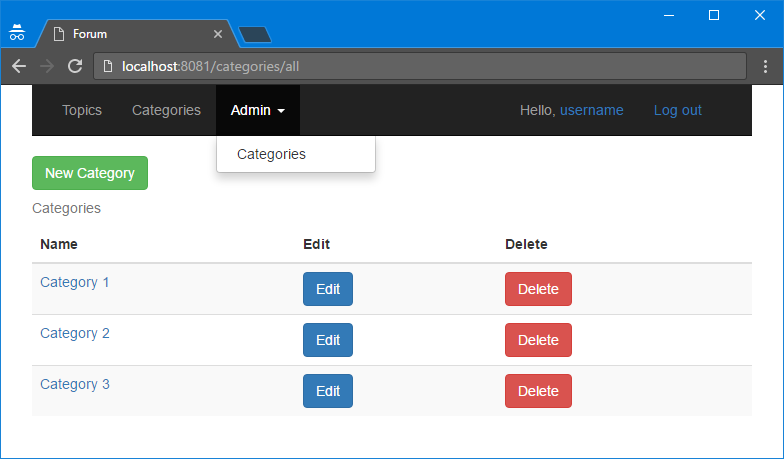


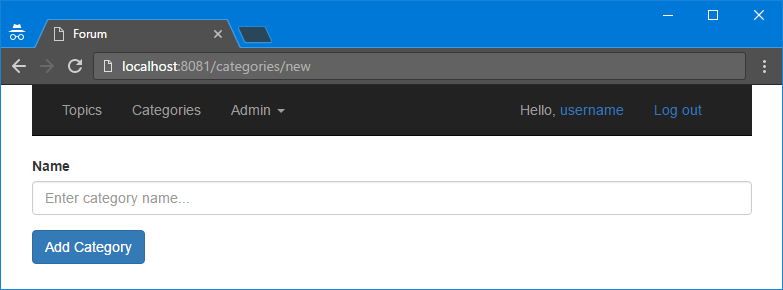


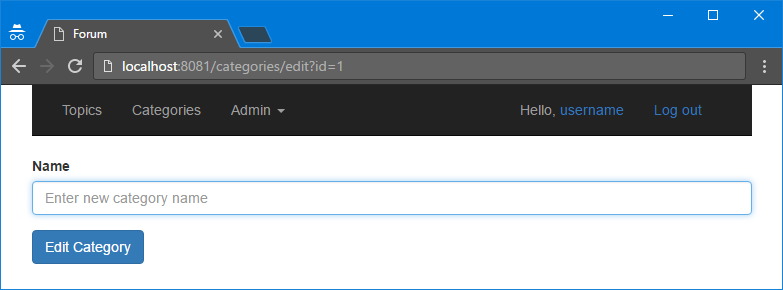


## Implement the Administration and Categories

The **first user who register** in the system also should become an **administrator**. As an administrator, he should have ability to **create, edit and delete categories**. The administrator options should appear in the **navigational bar**. If the logged in user is not administrator he should be able to **see the Admin section** but when he tries to access it he should be **redirected to home page**. The administrator has **all other privileges** as logged in user.





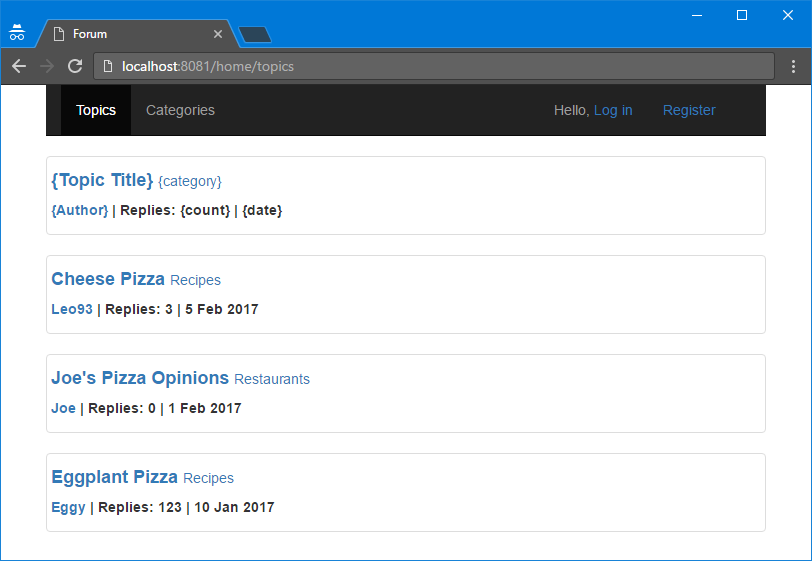


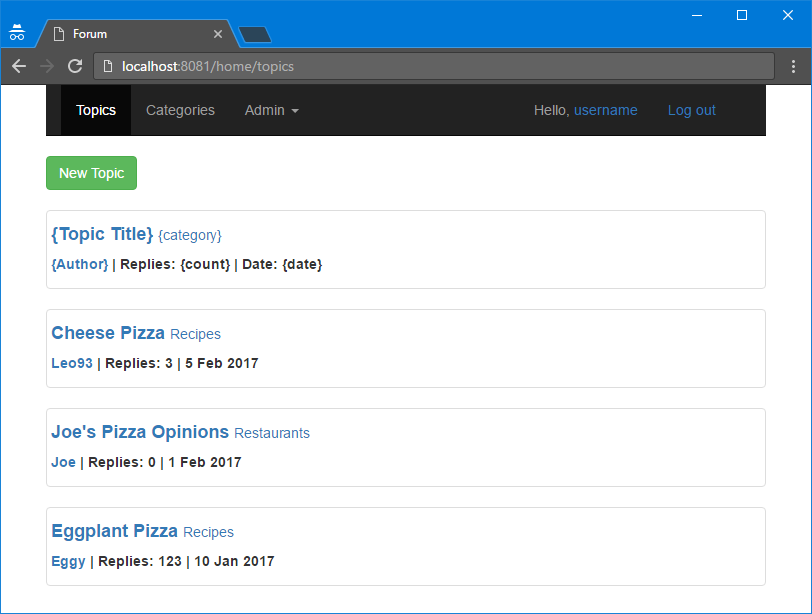
## Implement Home Page

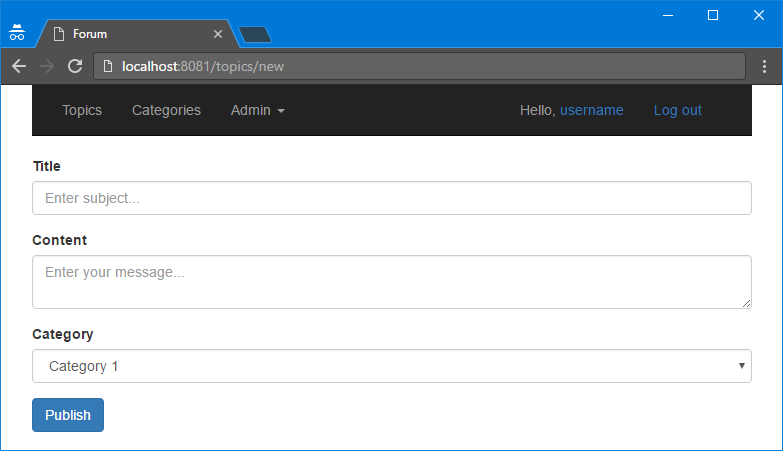
Home page should display **latest 10 topics** to the all users (guest and logged in). If the user is logged in, he should be able to see a **button** that redirects to a page with a form where the user can **create a new topic**.

The **form** for the new topic should contain:

* **Title** – no longer than 30 symbols
* **Content** – plain text no longer than 5000 symbols
* **Category** – dropdown control with all created categories. Notice if an admin adds a new category it should appear as option when new topic is created



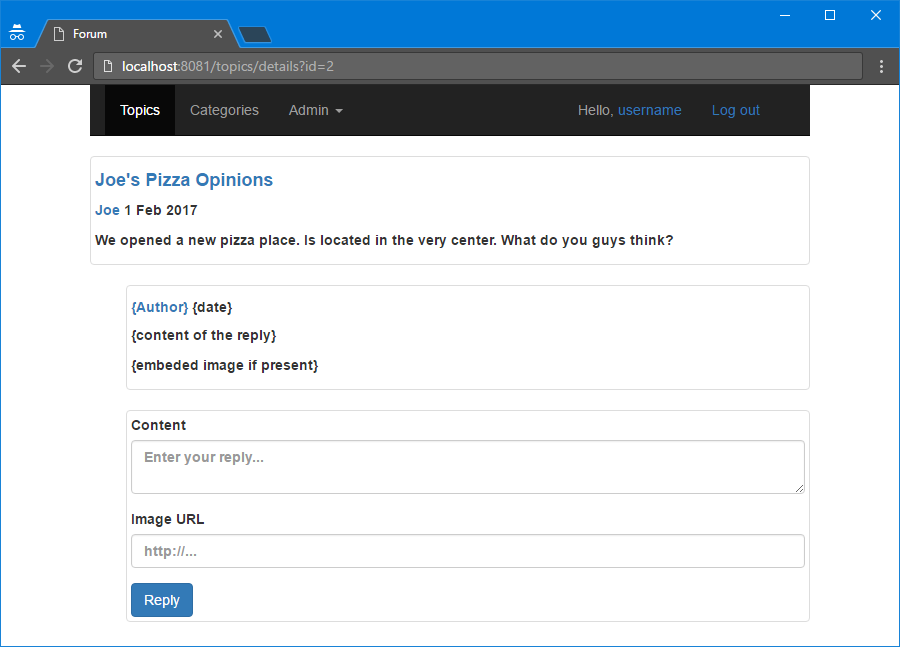


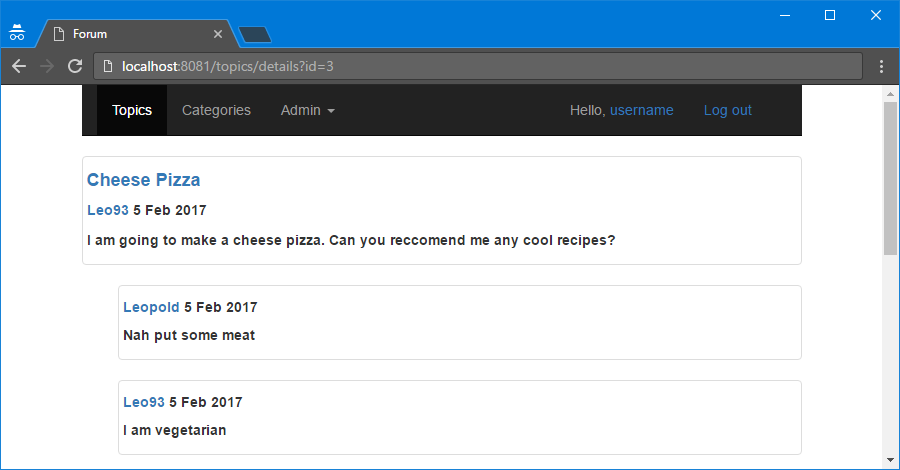


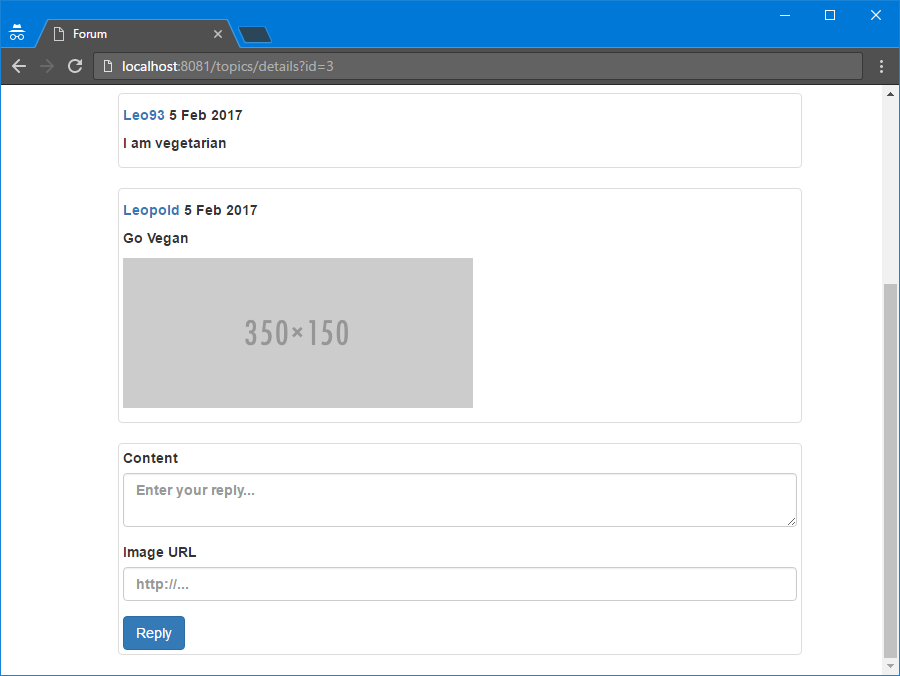
## Implement Topic Details Page

Each **topic** should be able to be seen in a **detailed view** where **all replies** should be shown under the content of the topic. Also, **each user** should be able to **add new reply** to that topic.

The **reply** can contain only **plain text** and **image URL** (optional).

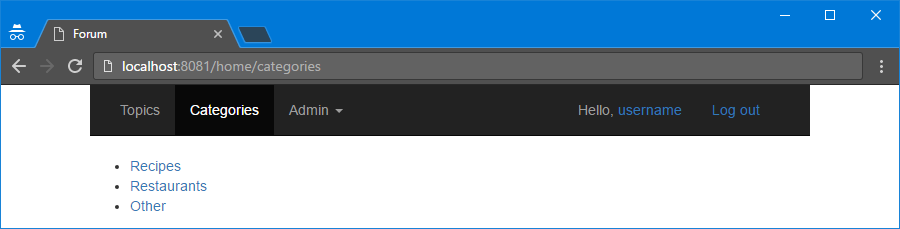


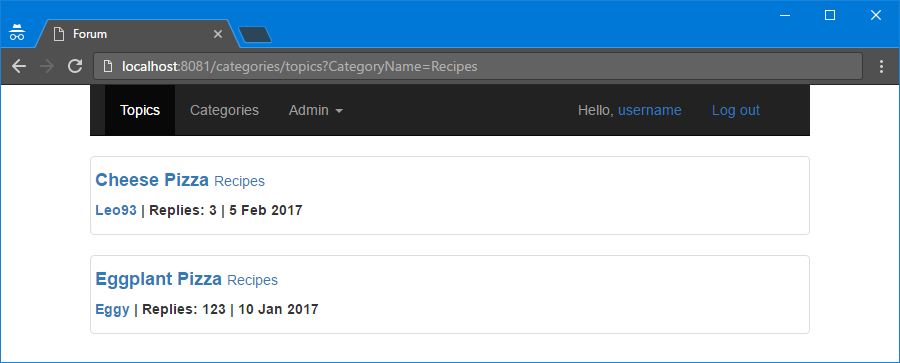




## Implement Categories Page

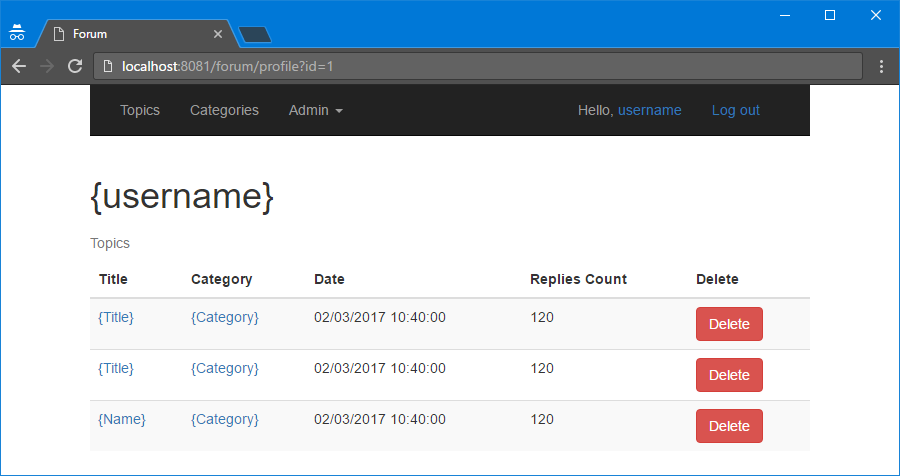
On that page, all logged in users can see a **list of links to categories** when select some category it should be redirected to page where he can see **all topics in that category**. Topics should be listed and **formatted** like in the **home page.**

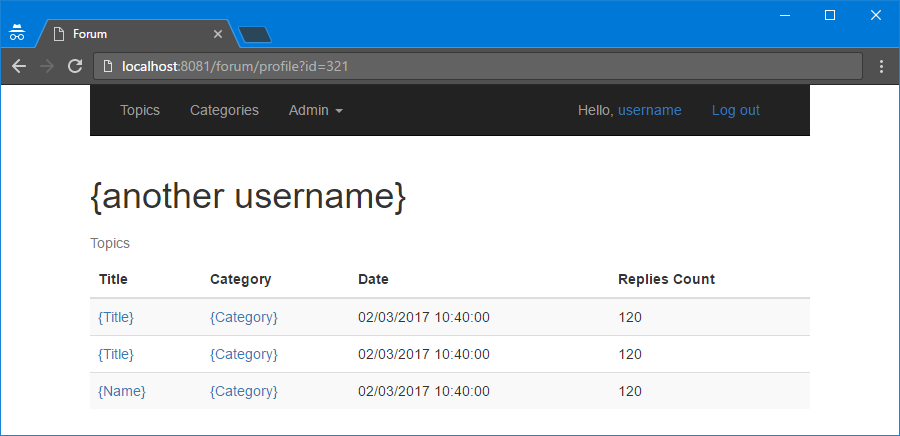




## Implement Profile Page

Each user should have a **profile page** where he can see **a list of all of his posts** with option to **delete** any of them.





## 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