Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

Web Technologies

Laboratory work #4

# Restrict the access

*Author:*
Dumitru Chetrusca

*Supervisor:*
Tudor Plugaru

Chișinău 2018

**Laboratory work #4**

## 1 Purpose of the laboratory

Understand how authentification and authorization works. Get more familiar with MVC pattern.

## 2 Laboratory Work Requirements

- **Main Requirement**
  - Basic authentification system (basically, the autentificated user can do anything);
- **Bonus Points:**
  - `(2pt)` 2 or more roles are defined and they have different actions defined and each role has well defined permissions;

## 3 Laboratory work implementation

### 3.1 Implemented Tasks and Points

- Basic authentification system (basically, the autentificated user can do anything);
- `(2pt)` 2 or more roles are defined and they have different actions defined and each role has well defined permissions;

### 3.2 Laboratory work analysis

Repository of the project :
https://github.com/DEMENCI/MoviesPlatform

In my application for authentification I've used some templates including models, views and controllers which are responsible for registration and log in.

To shape the data of and account I have a model named **UserProfile.cs** with some properties of an account and in the file **AccountModels.cs** I have multiple models defined which corespond to different scenarios which may appear at registration and log in.

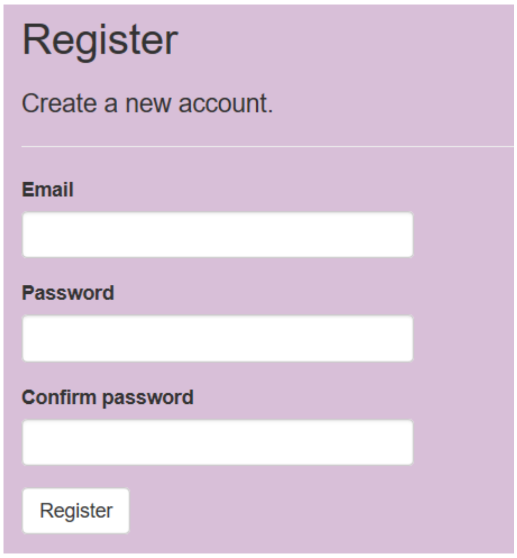
The **AccountController** has ActionMethods for all the actions possible at authentification. Also, in the views folder there are **Account** views.

```
public DbSet<UserProfile> UserProfiles { get; set; }
```

This way a new table is created where are stored account properties. If we would try to register with the same name we will get a message as in Figure 3.2.

To restrict the access I've used [Authorize] attribute at the reviews so that only a logged user could create, delete or edit.

## 3.3 Proof of the work with screens


Register

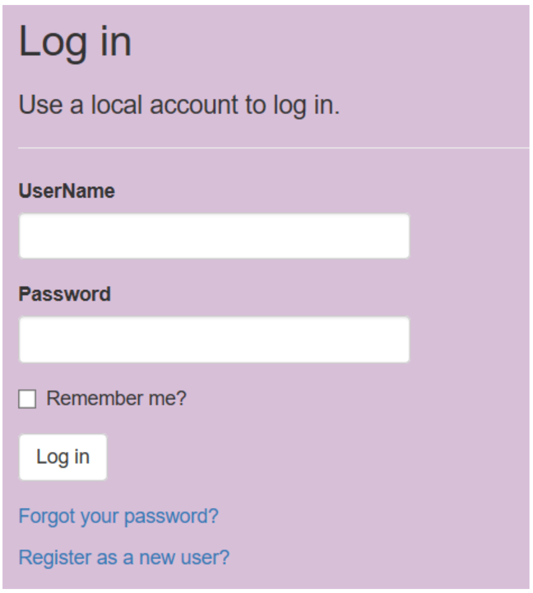
Create a new account.

Email

Password

Confirm password

Figure 3.1 – Registration


Log in

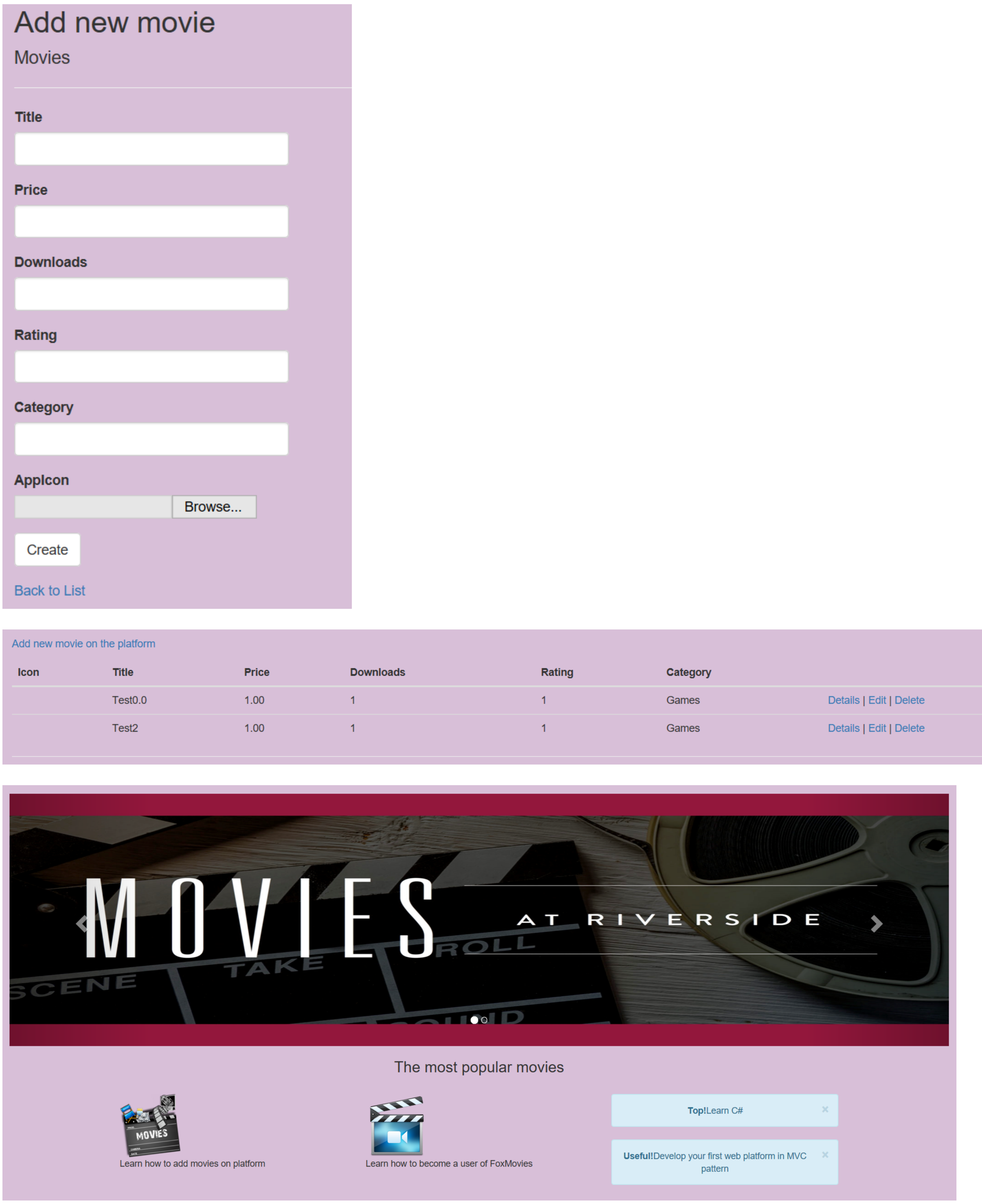
Use a local account to log in.

UserName

Password

Figure 3.2 – Login in

Figure 3.3 – Searching bar



Add new movie

Movies

Title

Price

Downloads

Rating

Category

AppIcon

Browse...

Create

Back to List

Add new movie on the platform

| Icon | Title | Price | Downloads | Rating | Category | |
|---|---|---|---|---|---|---|
| | Test0.0 | 1.00 | 1 | 1 | Games | Details \| Edit \| Delete |
| | Test2 | 1.00 | 1 | 1 | Games | Details \| Edit \| Delete |

The most popular movies

Learn how to add movies on platform

Learn how to become a user of FoxMovies

**Top!**Learn C#

**Useful!**Develop your first web platform in MVC pattern

Figure 3.6 – Main NavBar

## Conclusion

In this laboratory work I've learned about how to restrict the access and to create some roles on a web page.

The account controller with the help of coresponding models will use the table from database validate a registration and a login request. Also they predict scenarios of already registered users and similar functionalities.

If we want that our source code of the application to imply some logic and have some orthogonallity in it we should separate it into MVC entities based on what objects interact in our scenario.