# BCIT

**Comp 4952 HCI for Application Development**

**Technical Programming Option**

# Option Head Mirela Gutica

**Fall 2022**

Mark: \_\_\_\_\_\_\_\_ /100

Lab 6 ASP.NET Core MVC

This is an individual assignment. **No late assignments will be accepted**.

**This assignments has 3 parts:**

1. Read the PP presentations and the documentation content at the links provided in the PP presentations **Module 5 ASP.NET Core MVC Authentication and Authorization (1) 2022** and **Module 6 ASP.NET Core MVC Authentication and Authorization (2) Identity 2022.**
2. **Read the textbook Chapter 16 (How to authenticate and authorize users)**
3. **(25p) Part 1:** In your MovieStore application, do the exercises included in the document: **Implement Authentication and Authorization 2022.** Take screenshots of your output and submit the code.
4. **(55p) Part 2:** In your MovieStore application implement Login, Logout and Register using the Identity framework. Do the exercises included in the document **Implement Authentication and Authorization (2) - Identify 2022.** Take screenshots of your output and submit the code.
5. **(20p) Part 3. Answer the questions:**
6. What are the role and the properties of the IdentityUser class?
   1. The IdentityUser class’ role is to have functionality for authentication of a user. Its properties include: Username, email, password, confirm password, phone number, and phone number confirmed.
7. What steps do you need to do in order to add the Identity tables to the database?
   1. Make your MovieDbContext inherit from IdentityDbContext of type IdentityUser
   2. Add migration AddIdentityTable through Package Manager Console
   3. Then update database through Package Manager Console
8. Why there is no need for a View for Logout in the class example? In what cases would you include a View?
   1. Because in the implementation we just want to use a previous action. A case where we would want to return a view is to return a view with logout message feedback for the user.
9. Explain this code:
   1. There are actions for GET and POST requests for the login page. The GET request only cares about getting the previous page url that the user came from so that after the login, the user will be redirected back to the page the user was previously on. The POST request checks to see if the model state exists and then checks for the user credentials in the database tables. If there is an error in that process, an error message is rendered and notifies the user that their username/password is invalid. If it succeeded, it would check if there is a return url in the model retrieved. If there was a string URL, it would redirect to that url, if not, the user is redirected to the home page.

