Yehyun Kim – Lina

000848388

Lab1

Sep. 18. 2023

* Lab status (what’s working and not working)

Everything is working as expected. See the test plan below;

* Test Plan
  + Statement of authorship is present in EmployersController.
  + On the very first run, create 2 roles and 2 users (one for manager and one for Employees)
  + Unauthenticated users are prevented from accessing the EmployersController – redirects to Login page.
  + Managers have full access .
  + Employees have read only access.
  + Get action of the EmployersController in shared layout is working as expected.
  + Manager: creating employee with all the fields.
  + Manager: creating employee without Incorporation Date is working.
  + Manager: edit employee is working.
  + Manager: delete employee in working.
  + Employees: edit, delete, create Employee will not work.
* Three things you learned from the lab?
* Database Seeding was interesting to learn since it really helps creating initial data(roles, users) upon running. Learned how to preload the essential data to the database.
* MVC Concept: Dividing into model-view-controller helped me understand the structure of the project and its benefit(easy to scale the project.. etc.).
* Authentication & Authorization using Identity Framework. I learned how to create user accounts, setting their roles and restrict access based on user’s role. This process is important for building secure web applications.

* Three areas that you need to work on based on the lab?
* Explore more IdentityUser page. In this lab I only customized user register page and profile page. I saw few pages when I scaffold. Therefore I’d like to explore more.
* Complex authentication & authorization. In this lab there were only 2 user roles, however, in the real world there should be a lot more. I’d need to know how to handle those complex role-based access control.
* Front-end integration. If the application is using other front-end languages, I need to integrate these user management features with it. Therefore, it will be helpful to work on registration and profile management system using backend API instead of MVC.

* At least five key screenshots with Use Case Names

**In General – for both Manager and Employee:**

*Home Page*

**A screenshot of a computer

Description automatically generated**

*Profile Page*

A screenshot of a computer

Description automatically generated

*Register Page:*

**A screenshot of a computer

Description automatically generated**

*When anonymous user tries to access Employers page:*

**A screenshot of a computer

Description automatically generated**

**For Manager:**

*Employer page – Manager*

A screenshot of a computer

Description automatically generated

**‘**Create’ Clicked *– Manager*

A screenshot of a computer

Description automatically generated

‘Edit’ Clicked *– Manager*

A screenshot of a computer

Description automatically generated

*‘Details’ clicked – Manager*

A screenshot of a computer

Description automatically generated

‘*Delete’ Clicked – Manager*

A screenshot of a computer

Description automatically generated

**For Employee:**

*Employers Page – Employee*

**A screenshot of a computer

Description automatically generated**

*When click ‘Create’*

**A screenshot of a computer

Description automatically generated**

*When click ‘Edit’ – Employee*

A screenshot of a computer

Description automatically generated

*‘Delete’ Clicked – Employee*

A screenshot of a computer

Description automatically generated

‘*Detail’ Clicked – Employee*

A screenshot of a computer

Description automatically generated