Students Handout

MVC-EF(Model View Controller)Workshop4

**Introduction**

In ASP.NET Core MVC, controllers play a pivotal role in handling user requests, orchestrating application logic, and managing interactions between models and views. This document explores two essential controllers within the "Mvc\_HireMeNow" application: PublicController and CompanyController.

**PublicController:**

**Overview:**

The PublicController is designed to manage public user interactions such as registration and login.

**Job Provider Registration:**

Endpoint: JobProviderRegistration (HTTP POST)

Description: Handles the registration of job providers.

Procedure:

Accepts user data for registration.

Calls the \_publicService.Register method.

Redirects to the login page upon successful registration.

**Login:**

Endpoints:

Login (HTTP GET)

Login (HTTP POST)

Description: Manages user login, sets a session variable (UserId), and redirects based on the user's role.

Procedure:

Accepts user credentials for login.

Calls \_publicService.LoginJobProvider for authentication.

Sets the UserId session variable.

Redirects to either the company registration or user registration page based on the user's role.

**CompanyController:**

The CompanyController focuses on company-related actions, particularly the registration process.

**Company Registration:**

Endpoints:

CompanyRegistration (HTTP GET)

CompanyRegistration (HTTP POST)

Description: Manages the registration of companies using AutoMapper for data mapping and sessions for user identification.

Procedure:

Checks if the company already exists using \_companyRepository.IsUserExist.

Maps CompanyDto to Company using AutoMapper.

Registers the company with \_companyService.Register.

Retrieves the user ID from the session.

Updates the user's profile with the new company information.

**Sessions with Microsoft.AspNetCore.Http:**

Purpose: Sessions allow us to store user-specific information across multiple requests.

Usage: The HttpContext.Session.SetString("UserId", result.Id.ToString()) statement sets the UserId session variable, ensuring it is accessible in subsequent requests.

**Conclusion:**

You have now explored essential aspects of ASP.NET Core controllers and sessions. Understanding how to manage user interactions and persist user-specific data is crucial in web development. Feel free to explore these concepts further and experiment with different scenarios.

Happy learning and coding!

Top of Form