Students Handout

MVC-EF(Model View Controller)Workshop1

**Overview:**

In the provided code snippets, you can observe the implementation of two controllers - AdminController and PublicController - in a C# ASP.NET MVC application. These controllers are part of a larger application designed for job management and user registration.

**AdminController:**

**Dependencies:**

The controller relies on three services: IAdminService, IJobService, and IPublicService.

It has a constructor that injects these services to enable dependency injection.

**Actions:**

**Login:**

The Login action handles both GET and POST requests for user login.

It utilizes the injected IAdminService to validate and authenticate an admin user.

If successful, it sets a session variable and redirects to the "joblist" action; otherwise, it displays an error message.

**UpdateProfile:**

The controller has two actions for updating admin profiles.

GET: Retrieves the logged-in user's profile for editing.

POST: Accepts form data to update the admin's profile using the UpdateProfile method from IAdminService.

**JobList:**

Retrieves a list of jobs using the injected IJobService and returns the "joblist" view.

**JobSeekerListing:**

Uses the JobSeekerListing method from IAdminService to get a list of job seekers.

**Registration:**

Handles both GET and POST requests for admin registration.

POST: Calls the RegisterAdmin method from IPublicService to register a new admin.

**PublicController:**

**Dependency:**

The PublicController has a dependency on IPublicService.

**Actions:**

Registration:

Handles both GET and POST requests for user registration.

POST: Calls the Register method from IPublicService to register a new user.

**Login:**

Handles both GET and POST requests for user login.

POST: Uses the LoginJobSeeker method from IPublicService for authentication.

If successful, sets session variables and redirects to the "Registration" view; otherwise, displays an error message.

**Important Notes for Students:**

**Dependency Injection:**

Discuss the significance of dependency injection in making the controllers more modular and testable.

**Authentication and Authorization:**

Emphasize the usage of roles and sessions for authentication in the AdminController and PublicController.

**Service Layer:**

Explain the separation of concerns by delegating business logic to service classes (IAdminService, IJobService, and IPublicService).

**Error Handling:**

Highlight the importance of error handling in web applications, as demonstrated by the use of try-catch blocks.

**MVC Architecture:**

Reinforce the Model-View-Controller (MVC) architecture and how these controllers handle user interactions and data processing