**MVC Workshop: Job-Seeker Profile Application**

Welcome to the MVC Workshop where you'll delve into building web applications using ASP.NET Core with a focus on the Model-View-Controller (MVC) architecture. In this workshop, we will explore two essential controllers, PublicController and UserController, and understand how they interact with various models to manage user registration, login, and profile information.

**PublicController**

**Registration**

View: Registration.cshtml

Description: Displays the registration form for job seekers.

Action: Registration(User user)

Accepts user registration data.

Calls \_publicService.Register(user) to register the user.

Redirects to the login page upon successful registration.

**Login**

View: Login.cshtml

Description: Allows users to log in.

Action: Login(string email, string password)

Accepts user credentials.

Calls \_publicService.LoginJobSeeker(email, password) to validate and authenticate the user.

Redirects to the user's profile (Index action in UserController) upon successful login.

**UserController**

**Profile**

View: Index.cshtml

Description: Displays the user's profile information.

Action: Index()

Retrieves user data based on the stored user ID in the session.

Renders the profile view (\_DetailView, \_EducationView, \_ExperienceView, etc.).

**Education**

View: \_EducationView.cshtml, \_AddEducationView.cshtml

Description: Shows user's education details and allows adding new entries.

Actions:

\_EducationView(): Displays existing education details.

\_AddEducationView(): Displays the form to add a new education entry.

\_AddEducationView(Education education): Adds a new education entry and updates the user's profile.

**Experience**

View: \_ExperienceView.cshtml, \_AddExperienceView.cshtml

Description: Displays user's work experience and allows adding new entries.

Actions:

\_ExperienceView(): Displays existing work experiences.

\_AddExperienceView(): Displays the form to add a new work experience entry.

\_AddExperienceView(Experience experience): Adds a new work experience entry and updates the user's profile.

**Skills**

View: SkillsView.cshtml, AddSkillsView.cshtml

Description: Shows user's skills and allows adding new skills.

Actions:

SkillsView(): Displays existing skills.

AddSkillsView(): Displays the form to add new skills.

AddSkill(string skill): Adds a new skill and updates the user's profile.

**Edit Details**

View: EditDetailsView.cshtml

Description: Allows users to edit their profile details.

Action: UpdateDetails(User updatedUser)

Accepts updated user details.

Updates the user's profile information.

**File Upload**

Action: UploadFileAsync(IFormFile fileData)

Handles asynchronous file uploads.

Saves the uploaded file in the "Uploads" folder.

Updates the user's profile image.

**Conclusion**

By engaging in this workshop, you'll gain hands-on experience in implementing a robust MVC architecture with the Repository Pattern in ASP.NET Core. Best of luck with your coding journey!