Workshop Handouts:

Creating Job Portal APIs with AutoMapper

Introduction:

Welcome to the workshop on creating Job Portal APIs using AutoMapper, Domain-Driven Structure, Repository Pattern, and EF JWT Authentication Implementation. In this workshop, you will learn how to build a Job Portal API with features such as job seeker profiles, qualifications, sign-up, and login. We will be using AutoMapper for object mapping, a Domain-Driven Structure for organizing code, the Repository Pattern for data access, and EF JWT for authentication.

Workshop Outline:

1. Setting Up the Project:

- Use a Domain-Driven Structure.

- Implement the Repository Pattern for data access.

- Integrate EF JWT for authentication.

2. Job Seeker Profile Controller:

- Overview of the `JobSeekerProfileController` class.

- Explanation of attributes like `[ApiController]` and `[Authorize(Roles = "JOB\_SEEKER")]`.

- Methods for adding qualifications, getting profiles, and updating profiles.

- Demonstration of using AutoMapper for data mapping.

3. Profile Service and DTOs:

- Understanding the `IJobSeekerProfileService` interface.

- Explanation of DTOs such as `JobseekerQualificationDTo` and `ProfileDTO`.

- Implementation of methods for adding qualifications, getting profiles, and updating profiles.

4. Job Seeker Controller:

- Overview of the `JobSeekerController` class.

- Methods for job seeker sign-up, email verification, password setting, and login.

- Utilization of AutoMapper for mapping data between entities and DTOs.

Code Walkthrough:

1. JobSeekerProfileController:

- [Authorize(Roles = "JOB\_SEEKER")](#): Ensures that only job seekers can access the controller.

- [HttpPost("{jobseekerId}/profile/{profileId}/Qualification")](#): Endpoint for adding qualifications to a job seeker's profile.

- [HttpGet("{jobseekerId}/profiledetails")](#): Endpoint for retrieving job seeker profiles.

- [HttpGet("/profile/{profileId}/Qualification")](#): Endpoint for getting qualifications associated with a profile.

- [HttpPut("UpdateJobSeekerProfile")](#): Endpoint for updating a job seeker's profile.

- [HttpPost("AddProfile")](#): Endpoint for adding a job seeker profile.

- [HttpGet("GetJobSeekerProfile/{jobSeekerId}")](#): Endpoint for retrieving profiles by job seeker ID.

2. JobSeekerController:

- [HttpPost("job-seeker/signup")](#): Endpoint for job seekers to sign up.

- [HttpGet("job-seeker/signup/{jobSeekerSignupRequestId}/verify-email")](#): Endpoint for verifying job seeker email.

- [HttpPost("job-seeker/signup/{jobSeekerSignupRequestId}/set-password")](#): Endpoint for setting job seeker password after sign-up.

- [HttpPost("job-seeker/login")](#): Endpoint for job seekers to log in.

Conclusion:

By the end of this workshop, you will have a clear understanding of how to structure a Job Portal API, integrate AutoMapper for object mapping, implement JWT authentication with EF, and follow domain-driven principles. Good luck with your development journey!