**Entities Layer**

در این بخش، کلاس‌های User و Job برای مدل‌سازی داده‌ها تعریف شده‌اند:

// کلاس

User برای تعریف اطلاعات کاربر

public class User

{

public int Id { get; set; } // شناسه یکتا برای کاربر

public string Username { get; set; } // نام کاربری برای ورود به سیستم

public string Password { get; set; } // رمز عبور برای ورود به سیستم

public string Resume { get; set; } // رزومه کاربر

}

**کلاس User** اطلاعات مربوط به کاربران سیستم را ذخیره می‌کند. این اطلاعات شامل شناسه یکتا (ID)، نام کاربری، رمز عبور و رزومه است.  
  
  
  
// کلاس Job برای تعریف اطلاعات شغل‌ها

public class Job

{

public int Id { get; set; } // شناسه یکتا برای شغل

public string Title { get; set; } // عنوان شغل (مثال: "مهندس نرم‌افزار")

public string Description { get; set; } // توضیحات شغل

public string Company { get; set; } // نام شرکت ارائه‌دهنده شغل

public bool IsReviewed { get; set; } // نشان‌دهنده اینکه شغل بررسی شده است یا خیر

}

**کلاس Job** اطلاعات مربوط به شغل‌ها را ذخیره می‌کند که شامل شناسه یکتا، عنوان، توضیحات، شرکت و وضعیت بررسی شغل است.

**Use Cases Layer**

در این بخش، رابط‌های IUserService و IJobService برای انجام عملیات‌های مختلف مانند ثبت‌نام، ورود، و مدیریت شغل‌ها ایجاد شده است:

// رابط برای خدمات کاربر

public interface IUserService

{

void Register(User user); // برای ثبت‌نام یک کاربر جدید

User Login(string username, string password); // برای ورود کاربر

void UpdateUserProfile(int userId, User user); // برای به‌روزرسانی پروفایل کاربر

}

**IUserService** رابطی است که عملیات‌هایی مانند ثبت‌نام، ورود و به‌روزرسانی پروفایل را برای کاربر مدیریت می‌کند.

// رابط برای خدمات شغل

public interface IJobService

{

void PostJob(Job job); // برای ایجاد شغل جدید

IEnumerable<Job> ListJobs(); // برای نمایش لیست تمام شغل‌ها

IEnumerable<Job> SearchJobs(string keyword); // برای جستجوی شغل‌ها با استفاده از کلیدواژه

void ApplyForJob(int jobId, int userId); // برای درخواست شغل توسط کاربر

void ReviewJob(int jobId); // برای بررسی و تایید شغل

}

**IJobService رابطی است که عملیات‌هایی مانند ارسال شغل جدید، مشاهده شغل‌ها، جستجو برای شغل‌ها و درخواست شغل را انجام می‌دهد.**

**Repositories Layer**

در این بخش، کلاس‌های UserRepository و JobRepository برای پیاده‌سازی منطق ذخیره‌سازی داده‌ها و انجام عملیات‌های مربوطه ایجاد شده‌اند.

// کلاس UserRepository برای مدیریت ذخیره‌سازی و عملیات کاربران

public class UserRepository : IUserService

{

private readonly List<User> \_users = new(); // لیستی از کاربران

public void Register(User user) => \_users.Add(user); // اضافه کردن کاربر جدید

public User Login(string username, string password) =>

\_users.FirstOrDefault(u => u.Username == username && u.Password == password); // جستجو برای کاربر با نام کاربری و رمز عبور

public void UpdateUserProfile(int userId, User user)

{

var existingUser = \_users.FirstOrDefault(u => u.Id == userId); // جستجو برای کاربر بر اساس ID

if (existingUser != null)

{

existingUser.Username = user.Username; // به‌روزرسانی نام کاربری

existingUser.Password = user.Password; // به‌روزرسانی رمز عبور

existingUser.Resume = user.Resume; // به‌روزرسانی رزومه

}

**UserRepository** مسئول مدیریت کاربران است، شامل عملیات ثبت‌نام، ورود و به‌روزرسانی اطلاعات کاربر.

// کلاس JobRepository برای مدیریت ذخیره‌سازی و عملیات شغل‌ها

public class JobRepository : IJobService

{

private readonly List<Job> \_jobs = new(); // لیستی از شغل‌ها

private readonly Dictionary<int, List<int>> \_applications = new(); // ذخیره درخواست‌های شغل‌ها

public void PostJob(Job job) => \_jobs.Add(job); // ایجاد شغل جدید

public IEnumerable<Job> ListJobs() => \_jobs; // بازگشت تمام شغل‌ها

public IEnumerable<Job> SearchJobs(string keyword) =>

\_jobs.Where(j => j.Title.Contains(keyword, StringComparison.OrdinalIgnoreCase)); // جستجو برای شغل‌ها با کلیدواژه

public void ApplyForJob(int jobId, int userId)

{

if (!\_applications.ContainsKey(jobId))

\_applications[jobId] = new List<int>();

\_applications[jobId].Add(userId); // اضافه کردن درخواست کاربر برای شغل

}

public void ReviewJob(int jobId)

{

var job = \_jobs.FirstOrDefault(j => j.Id == jobId); // پیدا کردن شغل بر اساس ID

if (job != null)

job.IsReviewed = true; // تغییر وضعیت به "بررسی شده"

* **JobRepository** مسئول مدیریت شغل‌ها است و عملیات‌هایی مانند افزودن شغل جدید، جستجو، درخواست شغل و بررسی شغل‌ها را انجام می‌دهد.
* **Controllers**
* در این بخش، JobPortalController که رابط بین لایه‌های مختلف است، عملیات‌هایی مانند ثبت‌نام، ورود، ایجاد شغل و درخواست شغل را مدیریت می‌کند.

// کلاس کنترلر برای مدیریت درخواست‌ها

public class JobPortalController

{

private readonly IUserService \_userService;

private readonly IJobService \_jobService;

public JobPortalController(IUserService userService, IJobService jobService)

{

\_userService = userService;

\_jobService = jobService;

}

// عملیات‌های مربوط به کاربران

public void RegisterUser(User user) => \_userService.Register(user);

public User LoginUser(string username, string password) => \_userService.Login(username, password);

public void UpdateUserProfile(int userId, User user) => \_userService.UpdateUserProfile(userId, user);

// عملیات‌های مربوط به شغل‌ها

public void PostNewJob(Job job) => \_jobService.PostJob(job);

public IEnumerable<Job> GetJobList() => \_jobService.ListJobs();

public IEnumerable<Job> SearchJobs(string keyword) => \_jobService.SearchJobs(keyword);

public void ApplyToJob(int jobId, int userId) => \_jobService.ApplyForJob(jobId, userId);

public void MarkJobAsReviewed(int jobId) => \_jobService.ReviewJob(jobId);

}

**JobPortalController** مسئول انجام عملیات‌ها بین کاربران و شغل‌ها است. این کلاس درخواست‌های مربوط به ثبت‌نام، ورود، ایجاد شغل، جستجو و درخواست شغل را مدیریت می‌کند.

var userService = new UserRepository();

var jobService = new JobRepository();

var controller = new JobPortalController(userService, jobService);

// ثبت‌نام کاربر جدید

var newUser = new User { Id = 1, Username = "Seyed\_Sediq", Password = "Seyed4444", Resume = "Experienced in C# Development" };

controller.RegisterUser(newUser);

// ارسال شغل جدید

var newJob = new Job { Id = 1, Title = "Software Engineer", Description = "Develop high-quality software solutions", Company = "Tech Co." };

controller.PostNewJob(newJob);

// ورود کاربر

var loggedInUser = controller.LoginUser("Seyed-Sediq", "Seyed4444");

// درخواست شغل

if (loggedInUser != null)

controller.ApplyToJob(1, loggedInUser.Id);

// جستجو برای شغل

var searchResults = controller.SearchJobs("Engineer");

// بررسی شغل

controller.MarkJobAsReviewed(1);

**\*سید میکاییل صدیق\***

**-درس آزمایشگاه مهندسی نرم افزار-**

**+پروژه سایت کاریابی+**