

# Job Portal System - Project Specification

## Project Description

The Job Portal System is a comprehensive web application built using ASP.NET Core MVC that bridges the gap between employers and job seekers. It allows companies to publish job opportunities and candidates to search, apply, and manage their applications efficiently. The system integrates role-based authentication, server monitoring, data analytics, and communication features to simulate a real-world employment platform.

## Main Features

- Authentication & User Roles
- Employer Module
- Job Seeker Module
- Job Listing & Search
- Admin Module
- Server Monitoring & Logging
- Communication & Notifications
- Reporting & Analytics
- Security & Validation
- Optional Add-ons (Advanced Features)

## Detailed Features

### 1. Authentication & User Roles

- ASP.NET Core Identity authentication.
- Roles: Admin, Employer, Job Seeker.
- Password reset via email.
- Profile management for each user type (personal info, photo, CV).

### 2. Employer Module

- Create, edit, and delete job postings.
- Set job details: title, description, requirements, experience, salary, location, category.
- Manage job applications and update status.
- View analytics per job post.
- Export applicant list to Excel or PDF.

### 3. Job Seeker Module

- Browse and filter jobs by category, location, or type.
- Upload and update CV.
- Apply for multiple jobs with cover letters.
- Save favorite jobs.

- Track application history and status.
- Receive notifications for job updates.

#### 4. Job Listing & Search

- Dynamic job listing with filters, sorting, and pagination.
- Search by keyword, company name, or title.
- View detailed job descriptions.
- Recently Posted and Trending jobs sections.

#### 5. Admin Module

- Manage all users and roles.
- Approve or reject company registrations.
- Review and delete inappropriate jobs.
- Access analytics dashboard (users, jobs, applications).
- Configure system settings and maintenance mode.

#### 6. Server Monitoring & Logging

- Middleware for request and error logging.
- Track performance metrics (response time, failed requests).
- Health check endpoint for diagnostics.
- Real-time admin dashboard for server metrics.
- Use ILogger or Serilog for structured logging.

#### 7. Communication & Notifications

- In-app messaging between employer and job seeker.
- Email notifications for application updates.
- Real-time notifications using SignalR.

#### 8. Reporting & Analytics

- Downloadable reports (PDF, Excel).
- Charts for applicants per job, job trends, conversion rates.
- Employer analytics by experience or education.

#### 9. Security & Validation

- Role-based access control.
- Validation for input and file uploads.
- HTTPS enforcement.
- File type restrictions for CV uploads.

#### 10. Optional Add-ons

- AI-based job recommendation.
- Resume parsing.
- Job expiry system.
- Two-factor authentication.
- REST API for mobile integration.

- Dark/Light mode toggle.
- Chatbot for FAQs.

### Technical Requirements

- Framework: ASP.NET Core MVC (.NET 8 or 9)
- Database: SQL Server (Entity Framework Core)
- Authentication: ASP.NET Core Identity
- Logging: Serilog
- Real-time Communication: SignalR
- Reporting: iTextSharp or Syncfusion for PDF generation

### Bonus Ideas

- Integrate Google Maps API for job location.
- Add multi-language support (English/Arabic).
- Include job categories from database.
- Generate auto CV previews.
- Add referral system for sharing jobs.

### Evaluation Criteria

Students will be evaluated based on the following criteria:

1. Correct implementation of MVC architecture.
2. Database design and entity relationships.
3. Code quality, comments, and naming conventions.
4. User interface and usability.
5. Proper use of authentication and role-based access.
6. Implementation of logging and monitoring.
7. Bonus points for extra features (AI, charts, APIs, etc.).

