TALENT SUBMISSION

Project Overview:

The Talent Submission Form System is a centralized platform created to streamline the process of collecting, validating, and managing talent applications for hiring or internal evaluation.
The system offers a modern and user-friendly interface where candidates can submit their profiles, upload resumes, provide skill details, and track submission status.
The backend is built with **.NET Core 9 MVC**, offering high performance and modular architecture, while the application is containerized using **Docker** for simplified deployment and scalability.

Role & Responsibility:

Frontend & UI Development

- Built responsive and clean UI pages for talent submission, form validations, resume upload, and dashboard views using **.NET Core 9 MVC Views**, **HTML5**, and **Bootstrap**.
- Implemented client-side validation and enhancements using JavaScript/jQuery.

Backend Development

- Developed controllers, model binding, business logic, and service layers using **C#** following clean architecture principles.
- Implemented resume parsing logic and talent profile processing workflows.
- Added modular components to support dynamic skill sets, experience details, and multi-step submission forms.

Database Development

- Designed SQL Server tables, stored procedures, triggers, and optimized queries to store candidate information, resume metadata, and submission logs.
- Implemented transactional operations for secure and consistent data storage.

API & Integration Work

- Developed and exposed internal APIs using **.NET Core 9** for handling talent submissions, resume processing, and admin module requests.
- Implemented secure communication between UI ↔ API with proper token validation.

Security & Access Control

- Added role-based access for Admin, Recruiter, and Candidate.
- Implemented file validation and security checks to prevent malicious file uploads.

Docker & Deployment

- Containerized the full application using **Docker**, including MVC app + SQL scripts.
- Wrote Dockerfiles and docker-compose configurations for multi-container setup.
- Managed deployments and environment configuration in local and cloud/on-prem environments.

Testing & Maintenance

- Performed API testing, UI testing, and debugging to ensure stability.
- Added centralized logging and exception handling for easier maintenance.

Key Objective:

- Create a simple and secure platform for candidates to submit their profiles digitally.

- Automate the recruitment data collection process and eliminate manual paperwork.

- Ensure fast, scalable, and containerized deployment using Docker.

- Provide recruiters with real-time access to submitted profiles and resume data.

- Improve accuracy and speed of talent shortlisting.

Core Features

◇ Talent Submission Form

- Multi-step talent submission form capturing personal details, skills, experience, and resume.
- File upload with validation to prevent harmful or oversized files.

◇ Resume Upload & Metadata Extraction

- Store resume metadata (file name, type, size).
- Backend process to extract key information (if required).

◇ Candidate Dashboard

- View submission status, update profile, and re-upload resume.

◇ Admin & Recruiter Panel

- View all submissions, filter by skills, experience, or status.
- Update candidate status such as *Shortlisted*, *Under Review*, *Rejected*, etc.

◇ Role-Based Access

- Secure role-based authentication for Candidate, Admin, and Recruiter.

◇ Database Layer

- Robust storage for talent records, resume metadata, and audit logs.

◇ Docker Containerization

- Application packaged into containers for easy deployment.
- Support for multi-container environment (.NET MVC app + SQL Server container).

Technology Stack

- **Frontend:** html5, css3, Bootstrap, jQuery/Javascript
- **Backend:** .NET CORE MVC, C#, Entity Framework
- **Database:** SQL SERVER
- **SERVICE:** DOCKER

Results:

- Reduced talent data processing time by **50–60%** through automated submission and tracking.

- Faster deployments and environment parity achieved using **Docker containers**.

- Improved accuracy and consistency of talent submissions via enforced validation rules.

- Enhanced recruiter productivity with centralized candidate data and filtering tools.

- Scalable architecture suitable for handling thousands of submissions.

-