

Digital Portfolio Project

Evaluation Report

1 Project Name

Digital Portfolio System

2 Project Goal & Objectives

Goal

To build an AI-powered web platform that allows professionals to **create, manage, and showcase** their digital portfolios in an intelligent, secure, and user-friendly way.

Objectives

- Provide a **centralized platform** for professionals to display skills, projects, and experiences.
 - Enable **AI-assisted portfolio creation** and improvement.
 - Ensure **secure authentication** with role-based access.
 - Offer **easy portfolio sharing** with customizable visibility settings.
 - Give **admins** monitoring & analytics tools for system oversight.
 - Provide **scalable architecture** for future expansion (mobile app, templates, job-matching).
-

3 Project Overview

The project is a **full-stack application** with:

- **Frontend (React + Tailwind CSS + Vite)** → User interface & interactions
 - **Backend (ASP.NET Core Web API + MySQL)** → Business logic, security, database operations
 - **AI Assistant (OpenAI GPT-3.5-turbo)** → Generates project descriptions, reviews portfolios, suggests skills, provides improvement tips
 - **Email & Notification System (SMTP, HTML templates)** → For confirmations, access requests, system alerts
 - **Admin Dashboard** → User & portfolio management, system analytics
-

4 Tools & Technologies

Backend

- **Framework:** ASP.NET Core 9.0
- **Database:** MySQL
- **Authentication:** Cookie-based auth + BCrypt hashing
- **AI:** OpenAI GPT-3.5-turbo API integration
- **PDF Generation:** IronPdf + DinkToPdf

Frontend

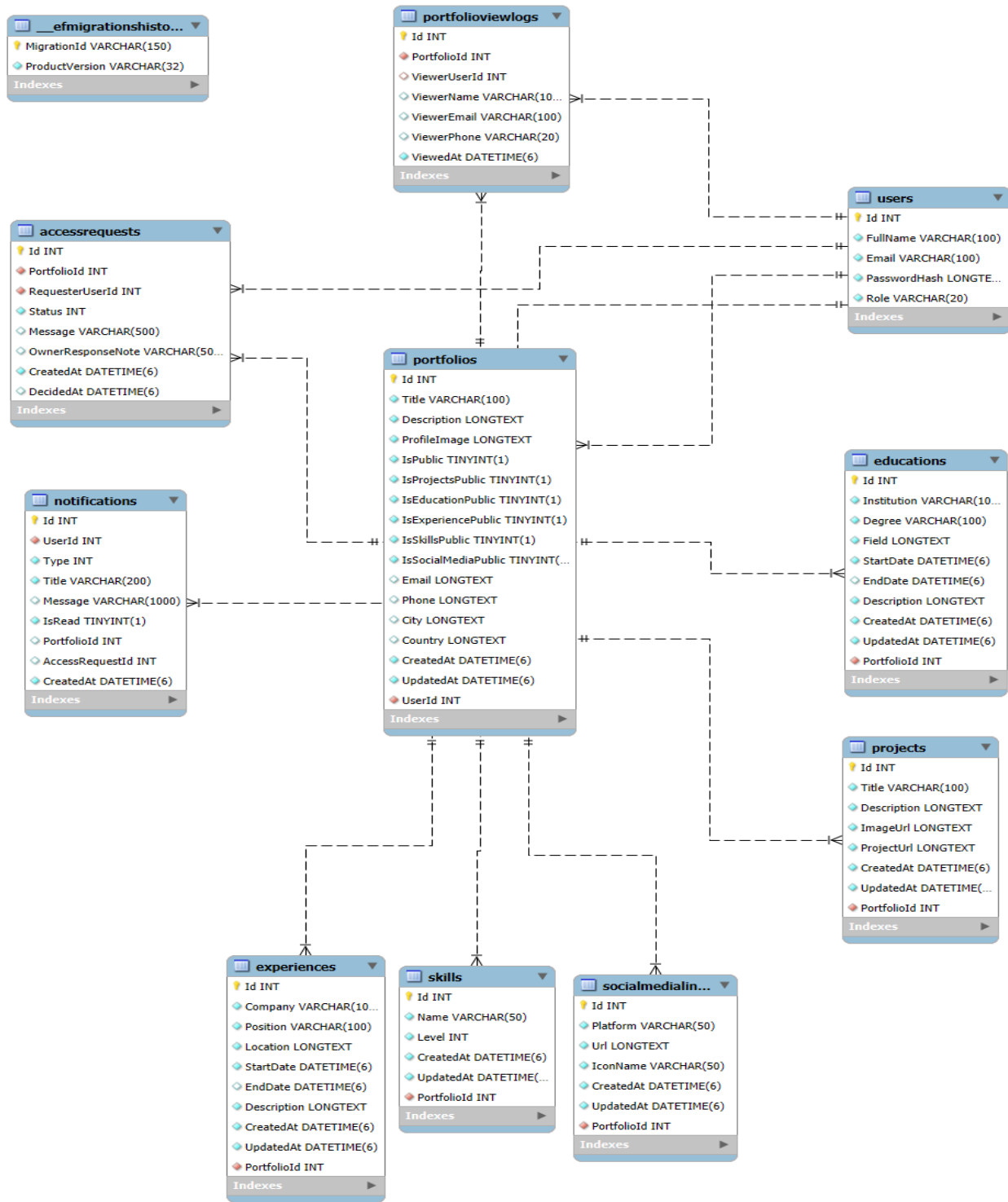
- **Framework:** React 18.2
 - **Build Tool:** Vite
 - **Styling:** Tailwind CSS + PostCSS
 - **Routing:** React Router DOM
 - **3D Effects:** Vanta.js + Three.js
 - **PDF Export:** html2pdf.js
 - **HTTP Requests:** Axios
-

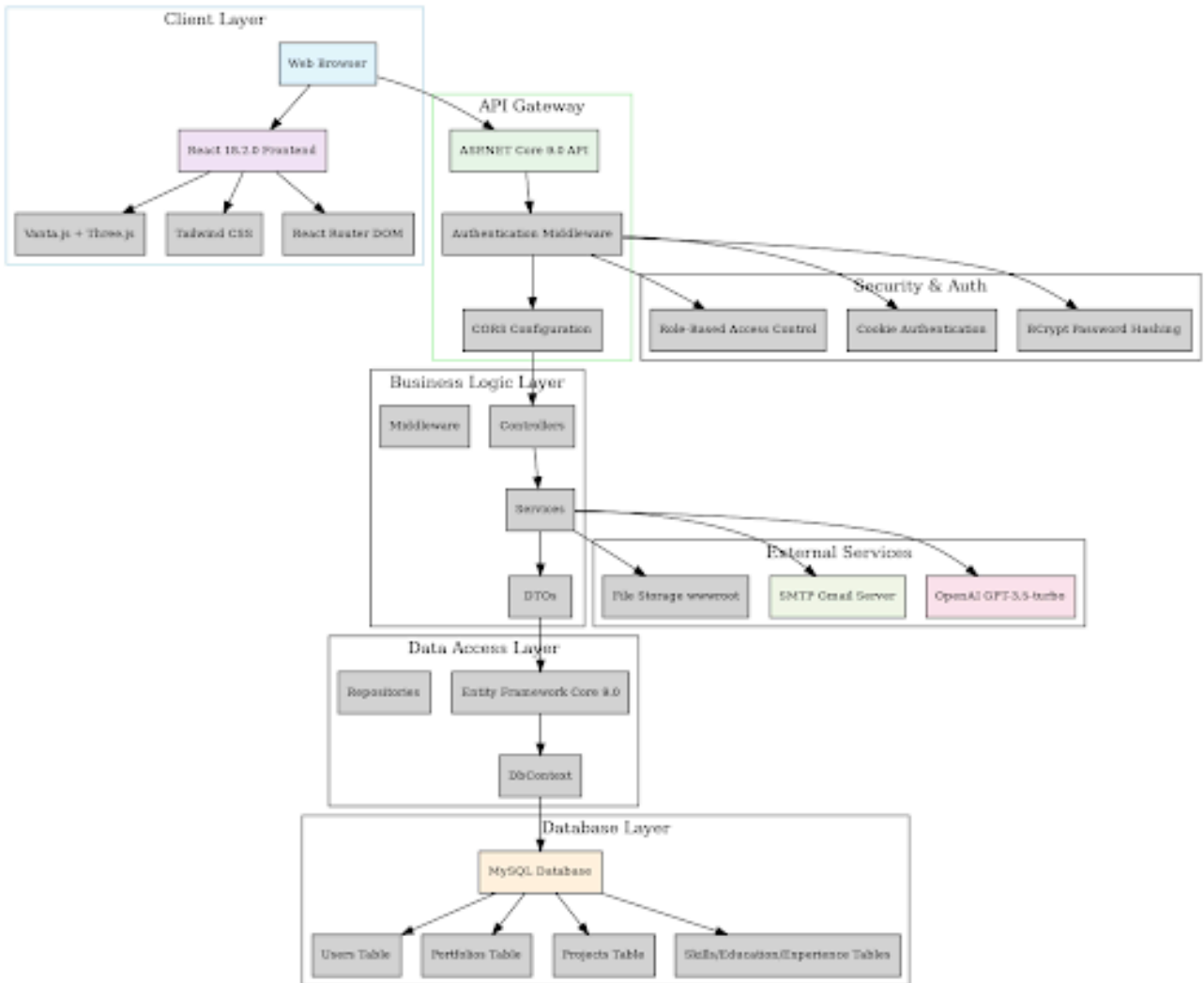
5 Database Schema & Application Architecture

Key Tables

- **Users:** Stores user details & roles (User/Admin)
- **Portfolios:** One portfolio per user, contains visibility options
- **Projects, Education, Experience, Skills, SocialMediaLinks** → Linked to portfolio (1:N)
- **Notifications:** System/user alerts
- **AccessRequests:** Request/approval for private portfolios
- **PortfolioViewLogs:** Records views for analytics

ER Model:







6 Key Features

Basic Features

- User authentication (register/login/logout)
- Portfolio creation with sections: Projects, Education, Experience, Skills, Social Links
- Customizable visibility (public/private)
- Email confirmation & notifications

AI-Powered Features

- Portfolio review & rating
- Intelligent improvement suggestions
- Automatic project description generator
- Personalized skill recommendations

Admin Features

- User & portfolio management
- Analytics & reports (views, engagement)
- Role-based access

Other Features

- PDF portfolio export
- Real-time password strength check



7 Security & Authentication

- **Password Security:**
 - Min length = 8, must include upper/lowercase, number, special char
 - Frontend strength meter + backend BCrypt hashing
 - **Session Security:**
 - Cookie authentication with HttpOnly, Secure, SameSite policies
 - **Role-based Access Control:**
 - Users → Manage their own portfolios
 - Admins → Manage all users + system analytics
 - **Email Security:**
 - SMTP with Gmail app-specific password
 - HTML templates for branded emails
-



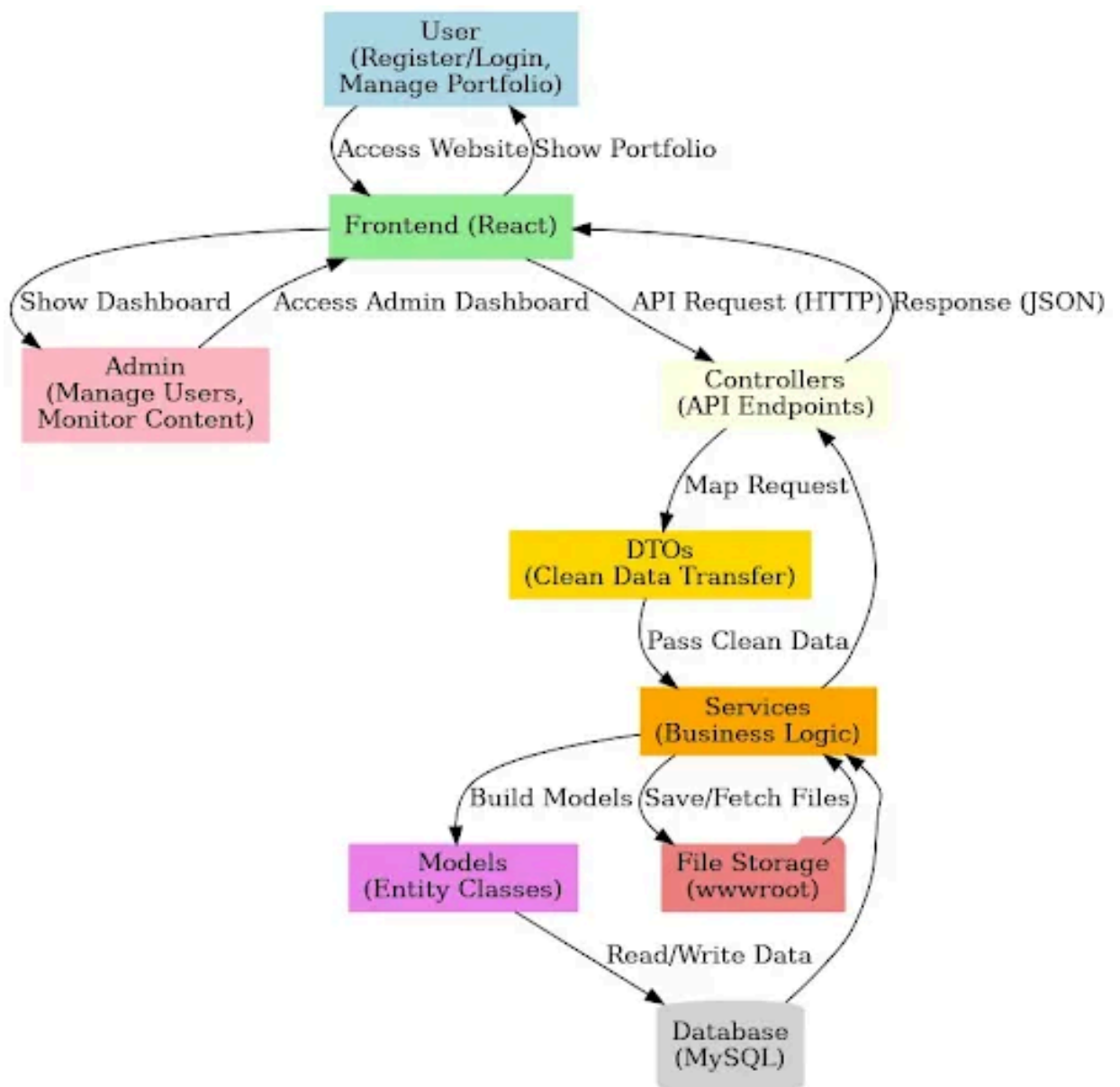
8 AI Assistant Integration

- **Model:** OpenAI GPT-3.5-turbo
 - **Capabilities:**
 - Analyze portfolio & give improvement suggestions
 - Generate professional project descriptions
 - Recommend career-based skills
 - Review each portfolio section with ratings
 - **Conversation Flow:**
 - Context-aware, maintains chat history
 - Provides actionable options (improve, add skills, etc.)
-

9 Workflows

User Registration

1. Frontend validation (email, password strength, confirm password)
2. Backend validation (duplicate email check, password hash, DB save)
3. Confirmation email sent via SMTP



Portfolio Creation

1. Authenticated user fills details
2. Backend saves portfolio + linked sections
3. Portfolio available for AI analysis

AI Interaction

1. User asks AI assistant (improve/review/generate description)
2. AI model generates structured response
3. Frontend chatbot shows suggestions + options

Admin Dashboard

- Monitor all users
- Manage portfolios
- Analyze engagement & view logs

10 Configuration

- **Configuration Files:**
 - `appsettings.json` → Database connection, SMTP, OpenAI API key, JWT secrets
-



11 Learning Outcomes

From this project, you demonstrate knowledge in:

- Full-stack development (React + ASP.NET Core)
 - Secure authentication & role-based access
 - AI integration with OpenAI
 - Database modeling with EF Core
 - Advanced UI/UX design with animations & responsive layouts
 - Email systems (SMTP, HTML templates)
 - Deployment-ready architecture
-

12 Future Enhancements

- Real-time collaboration with WebSockets
- Advanced analytics (portfolio engagement insights)
- Multiple portfolio templates
- LinkedIn/GitHub/Stack Overflow imports
- Mobile application (React Native/Flutter)
- Job matching with AI