TPSOW-PORTAL

Project Overview:

The **TPSOW (Third-Party Service & Operations Workflow) Portal** is an enterprise-level web application designed to manage vendor requests, approval workflows, operational tasks, and documentation in a single digital platform. Built using **.NET Core MVC**, **C#**, **Entity Framework**, and **SQL Server**, the portal provides a secure and scalable environment for internal staff and external vendors to collaborate.
It integrates with multiple backend systems through **WSDL-based SOAP services** and supports full workflow lifecycle—from request creation to approval and closure.

The solution features a modern UI using **Bootstrap**, **HTML5**, **CSS3**, **jQuery**, and **JavaScript**, ensuring a smooth and responsive user experience.
The system was deployed on IIS and includes modules for user management, service requests, audits, and reporting.

Role & Responsibility:

Frontend Development

- Developed responsive UI screens using **Bootstrap**, **HTML5**, **CSS3**, **jQuery**, and **JavaScript**.
- Implemented dynamic components such as request forms, validation mechanisms, and workflow action buttons.
- Improved usability with client-side form validations and interactive UI elements.

Backend Development

- Built MVC controllers, service layers, and business logic modules using **.NET Core MVC & C#**.
- Developed workflow logics, approval processes, and task routing functionalities.
- Integrated with external systems using **WSDL SOAP services** for data exchange.
- Implemented Entity Framework with code-first/migration approach for database operations.

Database & ORM

- Designed SQL Server database schemas, tables, stored procedures, and optimized queries.

- Used **Entity Framework** for CRUD operations, relationship mapping, and data validation.
- Ensured data integrity and performance tuning.

Integration Work

- Consumed enterprise SOAP services for vendor verification, authentication, and operational data retrieval.
- Built service wrappers to standardize SOAP interaction using C#.

Deployment & DevOps

- Managed full application deployment to **IIS server** across Dev, QA, and Production environments.
- Created web.config settings, connection string configurations, and environment setups.
- Performed smoke testing and post-deployment validations.

Client Communication

- Actively participated in requirement gathering and client calls.
- Provided technical suggestions, clarified requirements, and shared progress updates.
- Worked with QA, client stakeholders, and product managers for issue resolution and feature enhancement.

Key Objective:

- Digitize the entire vendor workflow management process.
- Streamline approvals, request tracking, and documentation handling.
- Integrate with external SOAP services for real-time validation and data syncing.
- Provide a scalable and secure portal accessible to internal and external users.
- Ensure smooth deployment and maintenance through IIS-based hosting.

Core Features

◇ Vendor Request Management

- Create, track, and manage vendor service requests.
- Multi-level approval workflow system.

◇ SOAP Service Integration

- Real-time data validation and status checks using WSDL SOAP services.
- Integration with backend systems for vendor and service data.

◇ User & Role Management

- Role-based authentication and authorization.
- Separate access for admin, approver, vendor, and operator.

◇ Dashboard & Reporting

- Status-wise request summary, pending approvals, and daily activity logs.
- Downloadable reports for audits and operational planning.

◇ Document Handling

- Upload and validate invoices, service records, and compliance documents.
- Secure storage with versioning support.

◇ Entity Framework ORM

- Clean and structured data access with automated migrations.
- Reduced boilerplate code for CRUD operations.

◇ Responsive UI

- Fully mobile-friendly design using Bootstrap.
- Smooth interaction with jQuery and JavaScript enhancements.

◇ IIS Deployment

- Hosted on IIS with environment-specific configuration.
- Automated deployment steps with minimal downtime.

Technology Stack

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

Results:

- Reduced operational time and manual paperwork by **70%**.
- Improved vendor transparency with real-time request status updates.
- Faster deployment cycles due to IIS automation and standardized configuration.

- Enhanced data accuracy and reduced human errors through SOAP integration and validation.

- Boosted performance and maintainability with .NET Core and Entity Framework.

- Improved overall client satisfaction due to active communication and timely delivery.

-