MPB PORTAL WEB APP - Documentation

Table of Contents

---

1. Project Overview

The **MPB PORTAL WEB APP** is a web-based system developed using **ASP.NET Core MVC** for managing user activities, reports, statistics, and case management. It allows different user roles (Admin, Organization, User) to perform specific operations, ensuring secure and efficient handling of data.

This project is done using ASP.NET Core MVC to improve scalability, security, and performance.

---

2. Features

- **User Authentication & Authorization**
  - Email verification before login

    - Secure password reset and change password functionality
- **User Management**
    - Role-based access control (Admin, Organization, User)
    - User registration and login with email verification
- **Case Management**
    - Create, edit, delete, and view cases
    - Admin controls case status updates
- **Reports Management**
    - Upload, edit, delete, and view reports
    - Reports are approved or declined by Admin
    - Status updates with dynamic color coding
- **Activity Management**
    - Organizations can create and manage activities
    - Admin has the power to approve or reject activities
- **Statistics & Data Visualizations**
    - Graphical representation of reports and cases using **Chart.js**
    - Downloadable reports in **CSV, Excel, and PDF**
- **Error Handling & Logging**
    - User-friendly error messages
    - Logging using built-in ASP.NET Core logging services

---

3. Technologies Used

Backend:

- **ASP.NET Core 6.0+**
- **Entity Framework Core**
- **ASP.NET Identity** (for authentication & authorization)
- **Microsoft SQL Server** (Database)

- **FluentValidation** (for model validation)

Frontend:

- **Bootstrap 5**
- **Chart.js** (for statistics visualization)
- **jQuery & AJAX** (for smooth UI interactions)

Other Libraries:

- **OfficeOpenXml (EPPlus)** – Excel exports
- **CsvHelper** – CSV report downloads
- **iTextSharp** – PDF report generation

---

4. Software Requirements

To run the **MPB PORTAL WEB APP**, ensure you have the following installed:

Development Environment

1. **.NET SDK 6.0+** (Download from dotnet.microsoft.com)
2. **Visual Studio 2022+** (Recommended for debugging and development)
3. **SQL Server 2019+** (Download from Microsoft SQL Server)
4. **Microsoft SQL Server Management Studio (SSMS)** – for database management
5. **Node.js** (optional, for frontend package management)

Required Packages

Run the following commands in **NuGet Package Manager Console** to install the necessary packages:

powershell

```
# ASP.NET Core Identity
dotnet add package Microsoft.AspNetCore.Identity.EntityFrameworkCore


# Entity Framework Core and SQL Server Provider
dotnet add package Microsoft.EntityFrameworkCore.SqlServer
dotnet add package Microsoft.EntityFrameworkCore.Tools


# EPPlus (Excel Reports)
dotnet add package EPPlus


# CSV Helper
dotnet add package CsvHelper


# iTextSharp (PDF Export)
dotnet add package itext7


# Email Sending (SMTP)
dotnet add package MailKit


# Data Validation
dotnet add package FluentValidation.AspNetCore
```

---

5. Installation Guide

Follow these steps to set up the project:

Step 1: Clone the Repository

bash

```
cd mpb-portal
```

Step 2: Set Up Database

- Open **SQL Server Management Studio (SSMS)** and create a new database: **Database Name: MPBPortalDB**
- Open **appsettings.json** and update the database connection string:

json

```
"ConnectionStrings": {
  "DefaultConnection": "Server=YOUR_SERVER;Database=MPBPortalDB;Trusted_Connection=True;MultipleActiveResultSets=true"
}
```

Step 3: Run Database Migrations

powershell

```
dotnet ef migrations add InitialCreate
dotnet ef database update
```

Step 4: Run the Project

powershell

```
dotnet run
```

- Open **https://localhost:7195/** in the browser.

---

6. Configuration

- **Email Settings (SMTP)**
  - Open appsettings.json and configure your SMTP settings:

json

```
"EmailSettings": {
  "SmtpServer": "smtp.gmail.com",
  "Port": 587,
  "SenderEmail": "your-email@gmail.com",
  "SenderPassword": "your-password",
  "EnableSSL": true
}
```

- **Set up the email sender in Program.cs:**

csharp

```
builder.Services.AddScoped<IEmailSender, EmailSender>();
```

---

7. Database Schema

Tables

- Users
- Roles
- UserRoles
- Cases
- Reports
- Activities

---

8. System Modules

1. **Authentication Module**
2. **User Management Module**
3. **Case Management**
4. **Report Management**
5. **Activity Management**
6. **Statistics & Reports**

---

9. User Roles and Permissions

Role	Permissions
Admin	Manage Users, Approve/Deny Reports, Update Cases
Organization	Create Activities, Submit Reports
User	Submit Reports, View Activities

---

10. API Endpoints

Authentication

- POST /Account/Login
- POST /Account/Register
- POST /Account/VerifyEmail
- POST /Account/ChangePassword

Case Management

- GET /Cases
- POST /Cases/Create
- PUT /Cases/Edit/{id}
- DELETE /Cases/Delete/{id}

Report Management

- GET /Reports
- POST /Reports/Create
- PUT /Reports/Edit/{id}
- DELETE /Reports/Delete/{id}
- GET /Reports/Download?type=pdf

---

11. Error Handling and Logging

- **Logging Mechanism:**
  - Logs errors in logs/error.log
  - Uses **ASP.NET Core built-in logging**
- **User-Friendly Messages**
  - Success & Error messages are stored in TempData
  - Bootstrap alerts display errors to the user

---

12. Security Features

- **Authentication & Authorization using ASP.NET Identity**
- **Data Validation using FluentValidation**

- **Hashed Passwords**
- **CSRF Protection**
- **SQL Injection Prevention**
- **Secure File Uploads**

---

13. Future Enhancements

- Add **real-time notifications** for reports and cases
- Implement **Multi-Factor Authentication (MFA)**
- Improve **audit logging** for tracking user actions
- Enhance **mobile responsiveness** of the dashboard

Conclusion

The **MPB PORTAL WEB APP** is a fully functional system that facilitates user authentication, case and report management, and real-time statistical insights. By following this documentation, you should be able to set up, run, and maintain the system efficiently.