.NET Core Azure

Sprint-1

Faculty Information System

Document Revision History

Date	Revision No.	Author	Summary of Changes

Table of Contents

INTRODUCTION

This document outlines a project for the .NET Line of Technology (LOT). The project is to develop Faculty Information System. This document contains the requirements, work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the .NET LOT.

SETUP CHECKLIST

Minimum System Requirements

- Intel Pentium 4 and Windows 2010
- Memory 4 GB
- Internet Explorer 11.0 or higher / Chrome
- SQL Server 2014 or 2016 client and access to SQL Server 2014 or 2016 server
- Visual Studio 2019
- Visual Studio Code
- Git

INSTRUCTIONS

- The code modules in the mini project should follow all the coding standards.

PROBLEM STATEMENT

OBJECTIVE

Need to develop Faculty Information System

Abstract of the project:

Faculty Information System allows to store and retrieve all information regarding faculty.

Faculty Information System will hold the following information:

- Faculty Personal Information
- Work History for each faculty member.
- Publication archiving and retrieval for each faculty.
- Degrees earned by the faculty members.
- Grants/Awards given to faculty members.
- Courses taught by the Faculty members.

Users of the System will be : Faculties, Students, Administrator

Role of Faculties :

- Login to System
- Add Personal Information, Work History, Publications and Grant Information, Courses taught by Faculty
- Update Information

- Delete Information
- View Information

Role of Administrator

- Login to System
- Create new users
- View the faculties publications yearwise, monthwise or recent
- Update the current Work information of Faculty
- Add Courses and Subjects
- Update/Delete Courses and Subjects
- Add Department

Role of Student

- Search the faculty publication and view the information

Phase 1: Create Client Application in Angular and Services using ASP.NET Core Web API.

MODULE LIST and MODULE DETAILS

Modules:

i. Faculty Class :- This class will contain methods which will allow to manage all Faculty

ii. Administrator class :- This class will contain methods to manage Administrator

iii. Department Class :- This class will contain methods to manage Department

iv. Publication class :- This class will contain methods to manage Publications

v. WorkHistory Class :- This class will contain methods to manage Work History

vi. Course Class : This class will contain methods to manage courses

b. Develop the prototypes for following functionalities:

i. **Login / Sign In**: Login screen would display asking user to enter 'User Id' & 'Password'. If the supplied user credentials are valid the HomePage would be displayed, else appropriate Login error message would be displayed

ii. **Home Page**: On successful user authentication (validation of userid/password provided by the user in login screen) the homepage would be displayed. The Homepage would contain below sections/contents:

Header section: The header section would be common across all the pages and would mainly have –

- Sign In Link – On click it would take to Login page
- Search: Search for information

Main content section: It would display generic Welcome message giving overview of the site. Should have menu based on user

c. For Faculty Entry :

i. Faculty will get option to add personal information, job history, publications, courses taught and grants/awards

ii. Faculty will get edit option if the information already available

iii. Faculty can view the information

d. For Administrator:

i. Administrator will create new users (student, faculty)

ii. Administrator will be able to add new department

iii. Administrator can add new designation

iv. Administrator will be able to view Faculty Information and can change current job information

v. Administrator will print faculty publication reports yearwise, monthwise, publicationwise or recent

vi. Administrator will add courses and subjects

b. Create the following database tables with following fields: [make your assumptions in case you require few more fields]

1. Users : UserID, UserName, Password
2. Faculty : FacultyID, FirstName, LastName, Address, City, State, Pincode, MobileNo, HireDate, EmailAddress, DateofBirth, DeptID, DesignationID
3. Publications : PublicationID, FacultyID, PublicationTitle, ArticleName, PublisherName, PublicationLocation, CitationDate
4. WorkHistory : WorkHistoryID, FacultyID, Organization, JobTitle, JobBeginDate, JobEndDate, JobResponsibilities, JobType
5. Degrees : DegreeID, FacultyID, Degree, Specialization, DegreeYear, Grade
6. Grants : GrantID, FacultyID, GrantTitle, GrantDescription
7. Courses : CourseID, CourseName, CourseCredits, DeptID
8. Department : DeptID, DeptName
9. Designation : DesginationID, DesignationName
10. Subjects : SubjectID, SubjectName
11. CourseSubject : CourseID, SubjectID
12. CoursesTaught : CourseID, FacultyID, SubjectID, FirstDateTaught

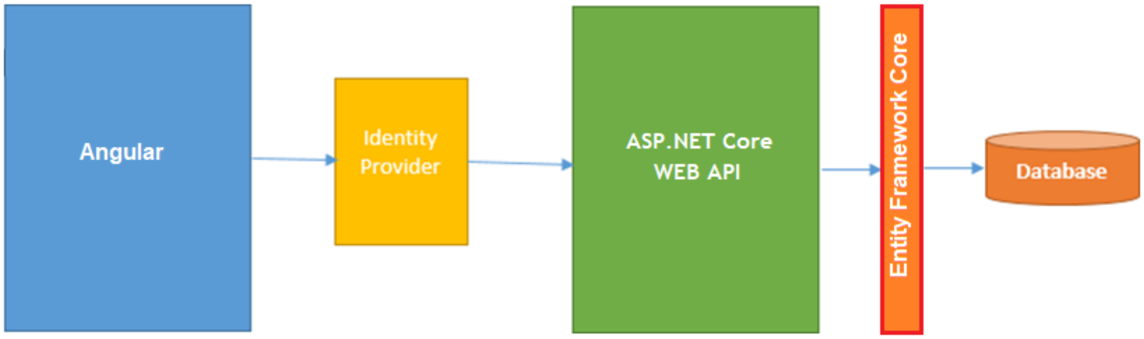
PROJECT STRUCTURE

Description: Create below sample reference project structure, which will help to reuse most of module for Web application. You are allowed to bring your own project structure and project to achieve the requirement.

In this sprint you have to use

- Angular UI as the presentation layer
- ASP.NET Core WEB API to create services
- Entity Framework Core to interact with the database.

The project should have minimum target framework – .Net Core 3.1 for Core WEB API Services and Angular 10 for UI Design



Design guidelines

- All the exceptions/errors to be captured and user friendly message to be displayed on the Common Error page.

IMPLEMENTATION

SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop the Faculty Information System by building the functionality incrementally in each of the course modules of .NET LOT.

Sr. No	Course	Duration	Functionality to be built
		(in PDs)	
1	Angular ASP.NET Core Web API Entity Framework Core SQL Server	5	Developing Presentation components (Angular), Business components (ASP.NET Core WEB API) and Data access components (Entity Framework Core)