.NET Core Azure

Sprint-1

GIG Management System

Document Revision History

Date	Revision No.	Author	Summary of Changes
18-07-2020	1.0	Kamal Prathap Singh	Case Study Created

Table of Contents

INTRODUCTION

This document outlines a project for the .NET Line of Technology (LOT). The project is to develop Gig management system (GMS). 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

Development of Gig Management System (GMS).

ABSTRACT OF THE PROJECT

This project is aimed at developing a **Online Gig Management System**.This is a web-based application that can be accessed over the web. This is mini social network that makes it really easy for live music lover to track the gigs of their favorite artists.
Artists can sign up and list their gigs. When adding a gig,they should specify the date/time,location and genre of the gig.An artist have a page called My Upcoming Gigs. From there, they should be able to edit or remove an existing gig,or add another gig to the list.
Users should be able to view all upcoming gigs or search them by artist,genre or locations.They should be able to view details of a gig and add it to their calendar. Additionally,users should be able to follow their favorite artists. When they follow an artists,they should see the upcoming gigs of their favorite artists in the Gig Feed.
This project contains four Module viz. Authentication, Gigs, Gig calendar and Following.

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

Macro level Operations/offerings:

1. Adding/Modifying Gig Information
2. Canceling a Upcoming Gig
3. Adding/Removing Gig From Calendar.
4. Viewing All Upcoming Gig
5. Add a Gig To Calendar
6. Search a Gig
7. Create Account
8. Login

MODULE LIST and MODULE DETAILS

CREATE GIG

Following info needs to be captured

- Gig ID (Must be Unique)
- Date Time
- Venue
- Artist
- Genre
- IsCanceled

CREATE LOGIN

- Username (Must be unique)
- Password
- Name

ADD GIG TO CALENDAR

User should be able to add a Gig to his/her calendar so that they can get all information regarding upcoming gig he/she going to attend if some changes are made about gig.

SEARCH GIG DETAILS

User should be able to search the Gig using Artist Name,Venue and Date Time.

Login

User should be able to login to the application provided valid login credentials.

PROJECT STRUCTURE

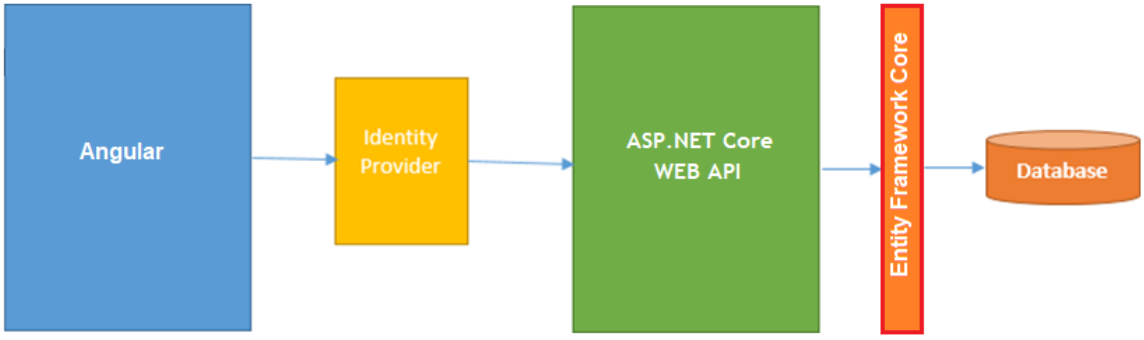
Description: Create below sample reference project structure, which will help to reuse most of module for Web application. You are allowed to bring your own project structure and project to achieve the requirement.

To evaluate, user needs to run Gig management UI, in Angular project it will prompt for option to perform actions like add gig to calendar, search gig details, login, Viewing all upcoming gigs, create an account etc.

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 Gig Management System (GMS) by building the functionality incrementally in each of the course modules of .NET LOT.

Sr. No	Course	Duration (in PDs)	Functionality to be built
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)

