



PAM Project – Blazor App Demo - Test Report

Test Description as per Customers Requirement

We had to complete a Test Task to showcase our Skills. The Task included following:

1. Client shared two screens – Login and Project Listing
2. The screens were in form of screenshot
3. We had to create simple necessary HTML from that screen
4. We had to complete the codes using Blazor framework

Solution Provided

Code Download URL

<https://www.dropbox.com/s/690u5st0sj8shv0/PAM%20Project%20-%20Blazor%20App%20Demo%20-%20Zobi%20Web%20Solutions.zip?dl=0>

This document includes – Code base for running the project as well as SQL database file

How to Run the Project

1. There is SQL Script folder which includes the Database. You can do its setup first
2. Change the Connection String in Appsettings.json file
3. After that open the Solution file, BlazorApp_Demo_ZobiWebSolutions.sln and Run the Project

Time Taken

24 Hours



Technology Used

Programming Language: ASP.NET Core 3.0 | ASP.NET MVC

Framework: Blazor (We have created Blazor Server App. We can also create App on Client side. Both have their own pros and cons)

Database: MS SQL Server 14

Required Packages: .NET Standard 2.1 | 3.1.100-preview1-014459

IDE: Visual Studio 2019

Design Pattern Framework Used: Unit of Pattern and Repository

Benefit of Design Pattern: In Coding, a design pattern is a general repeatable solution to a commonly occurring problem in software design. A design pattern isn't a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations. Design patterns can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation. Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns. Some other Benefits of this framework and Design pattern are as follows:

- Lightweight
- 100% Extensible
- Performance improvement
- Repository Pattern - 100% extensible ready
- Unit of Work Pattern - 100% atomic & transaction ready
- Service Pattern - pattern for implementing business, domain specific logic with 100% separation of concerns e.g. ICustomerService, IOrderService
- Minimize footprint of your ORM and data access layer
- DI & IoC 100% ready
- REST, Web API & OData 100% ready
- 100% testable & mockable
- 100% support for Stored Procedures
- Repository Pattern supports IEnumerable and/or IQueryable



Test Assumptions and Clarification

1. We have created Login screen. This we thought is necessary to not allow any unwanted access. Only registered users can Login. We have not implemented any Roles/Rights. Right now for Demo purpose, we have went through Static Array and you can use username and password as:

Username: user1@gmail.com

Password: 123456

Username: user2@gmail.com

Password: 123456

Username: user3@gmail.com

Password: 123456

Username: user4@gmail.com

Password: 123456

Username: user5@gmail.com

Password: 123456

Username: user6@gmail.com

Password: 123456

2. After Login, users can view the Project Listing Page
3. On this page, we have done following tasks. All the data comes the database of projects table:
 - 3.1. Sorting (Not asked for)
 - 3.2. Pagination (Not asked for)
 - 3.3. Filters
4. I have provided you Drop box URL to download the Source code and evaluate our Coding Skills. If you face any issues in installation, let me know, I can assist you or I can share Video of the Codes as well as working App.