Maretse Molaoa

ST10049039

APPR 6212

**Disaster Alleviation Foundation Web Application: Setup Instructions**

**Table of Contents**

Introduction………………………………………………………………………………

Prerequisites……………………………………………………………………………………..

Setting Up Azure SQL Database……………………………………………………………

3.1 Creating Azure SQL Database…………………………………………….

3.2 Configuring Connection String………………………………………………………

Setting Up the ASP.NET Core Project…………………………………………………………………..

4.1 Creating the Project……………………………………..

4.2 Configuring Entity Framework Core……………………………………………………..

User Registration and Login…………………………………………….

5.1 User Registration Logic………………………………………….

5.2 User Login Logic………………………………….

Testing the Application……………………………………………

Conclusion………………………………………………………………………….

References………………………………………………………………………………

**1. Introduction**

This document provides a comprehensive guide for setting up and utilizing the Disaster Alleviation Foundation web application, focusing on connecting the application to Azure SQL Database for user management and data storage.

**2. Prerequisites**

An active Azure subscription.

Visual Studio installed with the ASP.NET and web development workload.

Basic knowledge of C# and ASP.NET Core development.

**3. Setting Up Azure SQL Database**

**3.1 Creating Azure SQL Database**

Log in to the Azure Portal.

In the left-hand menu, select Create a resource.

Choose Databases and then select SQL Database.

Fill in the required fields:

Subscription: Your active Azure subscription.

Resource Group: Create a new resource group or select an existing one.

Database Name: Enter a name for your database (e.g., DisasterAlleviationDb).

Server: Click on Create new to set up a new server.

Server Name: Enter a unique name for your server.

Admin Login: Create an admin username.

Password: Set a strong password for your database admin.

Location: Choose a region close to your user base.

Click Review + create and then Create.

**3.2 Configuring Connection String**

Navigate to the Azure SQL Database you just created.

Find the Connection strings section in the database menu.

Copy the connection string provided and replace {your\_password} with your actual database password.

Update the appsettings.json file in your ASP.NET Core project with the copied connection string:

{

"ConnectionStrings": {

"DefaultConnection": "Server=tcp:yourservername.database.windows.net,1433;Initial Catalog=DisasterAlleviationDb;Persist Security Info=False;User ID=youradminusername;Password=yourpassword;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;"

}

}

1. **Setting Up the ASP.NET Core Project**

**4.1 Creating the Project**

Open Visual Studio.

Select Create a new project.

Choose ASP.NET Core Web App (Razor Pages) and click Next.

Name your project (e.g., DisasterAlleviationFoundation) and click Create.

Choose the appropriate framework version and ensure Authentication Type is set to Individual User Accounts.

**4.2 Configuring Entity Framework Core**

Open Package Manager Console (Tools > NuGet Package Manager > Package Manager Console).

Run the following command to install Entity Framework Core SQL Server package:

Install-Package Microsoft.EntityFrameworkCore.SqlServer

Add the database context to your Program.cs file:

var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");

builder.Services.AddDbContext<ApplicationDbContext>(options =>

options.UseSqlServer(connectionString));

1. **User Registration and Login**

**5.1 User Registration Logic**

The user registration form is auto-generated by ASP.NET Identity. Ensure that the registration page is accessible at /Account/Register.

The registration logic is implemented automatically; when a user registers, they are added to the AspNetUsers table in the database.

**5.2 User Login Logic**

The login form is also auto-generated. Access it at /Account/Login.

ASP.NET Identity handles the authentication process. Upon successful login, the user is redirected to the Index page or the page specified in the ReturnUrl.

**6. Testing the Application**

Run your application in Visual Studio (press F5).

Navigate to the registration page and create a new account.

Log in with the newly created account and verify that the user information is saved in the database:

Connect to your Azure SQL Database using SSMS or Azure Data Studio.

Run the following SQL query:

SELECT \* FROM AspNetUsers;

**7. Conclusion**

This document provides a structured approach to setting up the Disaster Alleviation Foundation web application and connecting it to Azure SQL Database. Follow the steps carefully to ensure proper functionality and integration.

**8. References**

Microsoft. (n.d.). Create an Azure SQL Database.

Microsoft. (n.d.). ASP.NET Core Identity.

Pro ASP.NET Core MVC 7 by Adam Freeman. (2022).

Entity Framework Core in Action by Jon P Smith. (2022).

Microsoft. (n.d.). Connection strings for SQL Server.