.NET Core Azure
Sprint
Easy Housing Solutions

Document Revision History

Date	Revision No.	Author	Summary of Changes
23-11-2017	1.0	Sangeetha C	Case Study Created
24-12-2022	2.0	Siddhesh Prabhugaonkar	Merged both sprints into single one as per new format.
02-02-2024	3.0	Rekha Chaudhari	Modified as per new curriculum

Table of Contents

Introduction 4

Setup Checklist 4

Instructions 4

Problem Statement 5

Objective 5

Project structure 10

EXPECTATION 10

REQUIREMENT 10

Implementation 11

Summary of the functionality to be built: 11

INTRODUCTION

This document outlines a project for the .NET Line of Technology (LOT). The project is to develop Easy Housing Solutions. This document contains the requirements, workflow 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 above Windows 7
- Memory 8 GB
- Edge / Chrome browser
- SQL Server 2019 and SQL Server Management Studio
- Visual Studio 2022
- Visual Studio Code
- Git

INSTRUCTIONS

- The code modules in the Sprint should follow all the coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory Sprint. Store your Project here.
- High Level Design document should be created (HLD).
- You can refer to your course material.
- You may also look up the help provided in the MSDN
- Since this project work will span over week, you will need to take care of maintaining the code.

PROBLEM STATEMENT

OBJECTIVE

Need to develop Easy Housing Solutions.

Abstract of the project:

This project is aimed at developing Easy Housing Solution (EHS) for administrators and Customers. It is proposed to develop a system to provide web-based Housing Solution system which will allow users to study, compare, buy, sell & then maintain his/her property details in a systematic way. Typical operations include registering and login of the user, searching houses for buying or renting. Owners can add their property, upload images, and get the property verified from the Easy Housing Company.

Easy Housing Solution Ltd launched in 2017, is an India – based startup company under the New India – Make in India initiative. The EHS project has been initiated by Housing Solution to help customers. This will be achieved

1. Reducing paperwork.
2. Password secured Asset Information.
3. All property details information & all processing is Free of cost.

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

Role of Administrator

Administrator is the primary user of the application and will have the complete control over the property which would be added and listed on the site. Administrator details will be already available in the system. The following will be responsibilities of administrator in context of EHS:

- Verify Property: Administrator is responsible to verify the property added by the owners. Only verified properties would be listed on the site.
- View Property by Region: Administrator would be able to list out all the property by region.
- View Property by Owner: Administrator would be able to list all property listed by an owner either for sell or rent.

- ❖ Delete Property: This option will enable the administrator to deactivate a property from being displayed on the site. The administrator must provide proper reason for DE activing the listed property.

Role of Owner or Seller

Owner or Seller would have to register themselves with Easy Housing Solution site. Once registered to the system they can performs the following operations.

- ❖ Add Property: As a part of this operation owner will record the details of the property which they intent to sell or rent out.

  Following information can be recorded by executive

  - Property Type
  - Property Name
  - Address
  - Region
  - Property Option (Sell or Rent)
  - Description
  - Images
  - Price Range
  - Initial Deposit (In case of Rent)
  - Landmark
- ❖ Edit Property: As a part of this operation owner will be able to edit the property details.
- ❖ View all Verified Property: This option will help the owner to view all the property listed by him/her and which have be verified by the Site Administrator.
- ❖ View all De activated Property: This option will allow the owner to view all the property listed by him/her and which have been de activated by the Site Administrator.
- ❖ Upload Image: This option would allow the owner to upload images of the property. The Owner can upload a maximum of 6 images per property.

Role of Consumer or Buyer

Consumer or Buyer can visit the Site and search for property. But to buy or rent a particular property he/she would have to register. Registered Buyer can compare two properties and can performs the following operations.

- ❖ Search Property: As a part of this operation buyer would be able to search all listed property based on region or Type.
- ❖ Sort Property by price: This option would help the buyer to sort the fetched data according to price.
- ❖ Add to Cart: This option would allow the buyer to add a particular property to his/her Cart for future reference.
- ❖ Get Owner Contact Details: In the list of fetched property there would be a button called Get Contact Details. This button would display the Owner Email Id and Registered Phone No.
- ❖ View Cart: This option would allow the buyer to view all the property which are there in his/her Cart.
- ❖ Delete from Cart: This option would allow the buyer to delete a property from his/her Cart.
- ❖ View Property Details: This option would provide the details of the property which the buyer has selected.

MODULE LIST and MODULE DETAILS

Modules:

- Users: - This will contain details of the user of EHS application
- Seller: - This will contain details of seller or owner
- Buyer: - This will store the details of buyers
- Property: - This will store the details of properties
- Images: - This will be used to store images of the property
- Cart: - This will store the property liked by buyers
- State: - This will store the state information
- City: - This would store the city information

a. Develop the prototypes for following functionalities:

i. **Registration**: - This page will allow buyer/Seller to register themselves.

ii. **Add Property: -** This page allows seller to add new property into the system.

iii. **Upload Image: -** This page allows seller to upload images for a given property.

iv. **Edit Property: -** This page allows the seller to edit the details of an existing property.

v. **View All Verified Property: -** This page will allow the seller to view all the verified property which belongs to him/her.

vi. **View All De-Activated Property: -** This page allows seller to view all the de-activated property which belongs to him/her.

vii. **View Property by Region: -** This page allows administrator to view all the property which belongs to a particular region.

viii. **View Property by Owner: -** This page allows administrator to view all the property added by a particular seller.

ix. **Verify or De-Active Property: -** This page will allow the administrator to either approve or de-activate a property.

x. **Search Property: -** This page will allow buyer to search property. This page will also have the option to sort the searched data based on price. Default sort order would be based on property name.

xi. **View Property Details: -** This page will allow buyer to view the details of the property. This will also have an option which would display the seller contact details.

xii. **Add To Cart: -** This page will allow buyer to add a property to his/her cart for future reference.

xiii. **View Cart: -** This page will allow buyer to view all the property which he/she has added to their cart.

xiv. **Delete from Cart: -** This page will allow buyer to delete a property from the cart.

xv. **Login: -** This page will allow users to login to the system.

The Structure of the above-mentioned tables.

i) **Users**: - UserName varchar (25) primary key, Password varchar(25) not null, UserType varchar(15)

UserType – Admin, Buyer, or Seller

ii) **Seller**: - SellerId int identity primary key, UserName varchar(25) not null, FirstName varchar(25) not null, LastName varchar(25), DateofBirth date not null, PhoneNo varchar(10) not null, Address varchar(250) not null, StateId int foreign key not null, CityId int foreign key not null, EmailId varchar(50) not null.

iii) State: - StateId int identity primary key, StateName varchar (30)

iv) **City: -** CityId int identity primary key, CityName varchar (50) not null, StateId int foreign key not null

v) **Buyer: -** BuyerId int identity primary key, FirstName varchar(25) not null, LastName varchar(25), DateOfBirth date not null, PhoneNo varchar(10) not null, EmailId varchar(50) not null

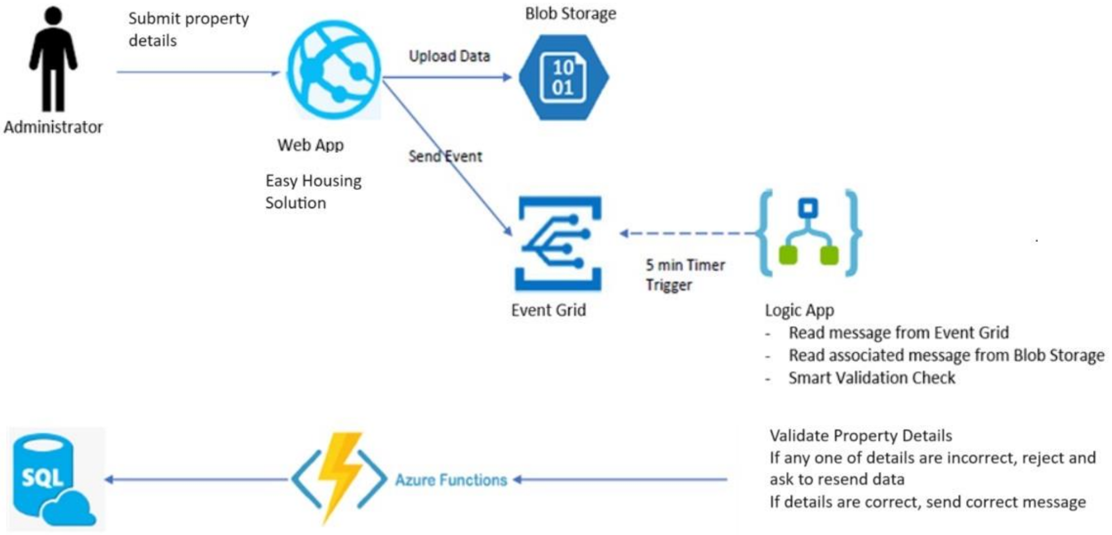
vi) Property: - PropertyId int identity primary key, PropertyName varchar(50) not null, PropertyType varchar(15) not null, PropertyOption varchar(10) not null, Description varchar(250), Address varchar(250) not null, PriceRange money not null, InitialDeposit money not null, Landmark varchar(25) not null, IsActive bit not null, SellerId int foreign key not null

vii) **Images: -** ImageId int identity primary key, PropertyId int foreign key not null, Image not null

viii) **Cart**: - CartId int identity primary key, BuyerId int foreign key not null, PropertyId int foreign key not null

PROJECT STRUCTURE

The project should have minimum target framework – .NET 6 for Asp.net Core MVC for UI Design


Administrator
Submit property details
Web App
Easy Housing Solution
Upload Data
Blob Storage
10
01
Send Event
Event Grid
5 min Timer Trigger
Logic App
- Read message from Event Grid
- Read associated message from Blob Storage
- Smart Validation Check
Validate Property Details
If any one of details are incorrect, reject and ask to resend data
If details are correct, send correct message
Azure Functions
SQL

EXPECTATION

Create Asp.net Core MVC app using

- Visual Studio 2022 IDE
- NET 6.0 / C# 10.0
- SQL Server 2019
- Entity Framework Core

REQUIREMENT

In this sprint, you must use:

- Asp.net Core MVC
- Use Entity Framework Core
- Azure Web App And Function App
- Azure BLOB Storage
- Azure Function

- Azure Logic App

Create Client Application in Asp.net Core MVC and host it on Azure Cloud.

Design guidelines

- All the exceptions/errors to be captured and user-friendly message to be displayed on the Common Error page.
- Asp.net Core MVC app must be created using Code First Approach.

IMPLEMENTATION

SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop the Easy Housing Solutions 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	ASP.NET Core MVC, Entity Framework Core, SQL Server, Azure services	3	Developing Asp.net Core MVC App hosting it on Web App (Azure App Service) Using Azure BLOB Storage, Using Logic App to validate data, Using Function app to write data to SQL Server