A PROJECT REPORT

ON

“STREET VENDOR WEB APP”

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF

BACHELOR’S IN

COMPUTER APPLICATION

SUBMITTED TO

BANGALORE NORTH UNIVERSITY, KOLAR

SUBMITTED BY

Name of Student	Enrollment No.
ANNIE PAUL	R1920005

GUIDED BY

DR.JOSEPHINE PRAPULLA

GOODWILL CHRISTIAN COLLEGE FOR WOMEN, BANGALORE

OCTOBER-2022

CERTIFICATE

This is to Certify that the project entitled "Street Vendor Web App" has been successfully completed by Annie Paul (Register number: R1920005) of sixth semester BCA in partial fulfillment of the requirements for the award of the Bachelor's in Computer Application prescribed by Bangalore North University and submitted to the Department of BCA of GOODWILLS CHRISTIAN COLLEGE FOR WOMEN the work carried out during a period for the academic year 2021-22 as per the curriculum .

Name of Guide	**Name of HOD**
DR.JOSEPHINE PRAPULLA	**DR.JOSEPHINE PRAPULLA**

EXTERNAL EXAMINER

Name of Principal

1.---------------------------

2.---------------------------

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my project guide Dr.Josephine Prapulla as well as our principal Sujatha Christopher who gave me the golden opportunity to do this wonderful project on the topic Street Vendor Web App, which also helped me in doing a lot of research and I got to learn about so many new things. I am really thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Date : Annie Paul

Place : R1920005

3RD BCA

ABSTRACT

Vending as a profession has been carried out all along in known history and it has been an integral part of both rural and urban culture. Everyone purchases something from a street peddler at some point in their life - hot dogs, pretzels, meat, vegetables, fruit, cold drinks, flowers etc. A street vendor is someone who sells food, goods and merchandise on the street or in an open-air market rather than at a traditional store. The street vendor's "store" is either a small outside area that can be locked and shut down at the end of the night, or a cart that can be moved from location to location, and taken home at the end of the day.

A paramount issue for street vendor's is that workplace is ideally a highly trafficked area located in an area with plenty of businesses and people. The weather is an element that all vendors must deal with. Days with perfect weather will yield higher profits, but vendors will also have to contend with days of pouring rain, high winds, and biting cold in which they will be lucky to break even for the day. It can be a long day that slowly drags by when no one is coming to a vendor’s cart, or it can be just the opposite with number of people waiting to buy the items being sold.

When there are no buyers around the vendor faces a hard time, travelling from one lane to another, one street to another, and so on, to every nook and corner of the streets, to sell his items until he finds buyers in order to make profit for the day which would be his only source of income to make a living for himself.

This app intends to solve this issue by letting the sellers know the number of buyers that require the seller’s product resulting in efficient usage of the seller’s time, effort and resources in order to make his share of profit where the seller can see a buyer zone within his current locations radius of 10 kilo meters and be able to access the buyer’s details. The buyers will be able to request the sellers to their location to deliver the products necessary .Thus making the lives of both the seller and the buyer much more effortless.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.INTRODUCTION

1.1 The street vendor's "store" is either a small outside area that can be locked and shut down at the end of the night, or a cart that can be moved from location to location, and taken home at the end of the day.

A street vendor's workplace is ideally a highly trafficked area located in an area with plenty of businesses and people. The weather is an element that all vendors must deal with. Days with perfect weather will yield higher profits, but vendors will also have to contend with days of pouring rain, high winds, and biting cold in which they will be lucky to break even for the day. It can be a long day that slowly drags by when no one is coming to a vendor’s cart, or it can be just the opposite with number of people waiting to buy the items being sold. If the vendor knew the number of customer and their whereabouts and requirements, the vendor’s life would be much easier thought of.

Where does my app come into picture?

The goal is to make life easier for the seller by avoiding un necessary travel to areas where there are no buyers. This app allows buyers to fill in their details like name, location and their requirements. The seller who wishes to sell his products enters his location. The application retrieves the seller’s location and the buyer’s location from the database and compares it, by mapping the buyers by the location of the seller’s radius. If the location matches, the seller can sell the list of items in the buyer’s locations called the buyer zone who are in need of the seller’s products. Thus, the seller can deliver the necessary items to the buyers.

1.2 Process:

- Buyer enters details, requirements, and location
- Seller enters the items he sells, his details and location
- The database checks if the location of the buyer and the seller match.
- The database checks if the items needed by the buyer and the items sold by the seller match.

- If a match is found, shows the buyers on the map of the seller.
- Also has additional features like :
- Role based Authorization
- Authentication
- Update email and password and phone number and many more

2.SCOPE OF PROJECT

This is web application, suitable for Street vendors. It has different divisions to maintain the sellers and the buyers, Admins and provides accurate location of buyers.
Register as a seller or buyer, sell items and purchase items, locate buyers in the map within the seller's radius, seller and buyer authentication and authorization, update details, delete details. It an application built to make the sellers lives easier. The scope of the project can be widened to accommodate additional features in the future which include displaying buyers required items and phone number in an info box on the map. Message prompt to the buyers that the seller is arriving, online payment of bills to the seller etc...

3.TECHNICAL REQUIREMENTS

1.**ASP.NET CORE:**

ASP.NET Core is a cross-platform, high-performance, open-source framework for building modern, cloud-enabled, Internet-connected apps.

With ASP.NET Core, you can:

- Build web apps and services, Internet of Things (IoT) apps, and mobile backends.
- Use your favorite development tools on Windows, macOS, and Linux.
- Deploy to the cloud or on-premises.
- Run on .NET Core.

ASP.NET Core provides the following benefits:

- A unified story for building web UI and web APIs.
- Architected for testability.
- Razor Pages makes coding page-focused scenarios easier and more productive.
- Blazer lets you use C# in the browser alongside JavaScript. Share server-side and client-side app logic all written with .NET.
- Ability to develop and run-on Windows, macOS, and Linux.
- Open-source and community-focused.
- Integration of modern, client-side frameworks and development workflows.
- Support for hosting Remote Procedure Call (RPC) services using gRPC.
- A cloud-ready, environment-based configuration system.
- Built-in dependency injection.
- A lightweight, high-performance, and modular HTTP request pipeline.
- Ability to host on the following:
    - Kestrel
    - IIS
    - HTTP.sys
    - Nginx

    - Apache
    - Docker
- Side-by-side versioning.
- Tooling that simplifies modern web development.

ASP.NET Core MVC provides features to build web APIs and web apps:

- The Model-View-Controller (MVC) pattern helps make your web APIs and web apps testable.
- Razor Pages is a page-based programming model that makes building web UI easier and more productive.
- Razor markup provides a productive syntax for Razor Pages and MVC views.
- Tag Helpers enable server-side code to participate in creating and rendering HTML elements in Razor files.
- Built-in support for multiple data formats and content negotiation lets your web APIs reach a broad range of clients, including browsers and mobile devices.
- Model binding automatically maps data from HTTP requests to action method parameters.
- Model validation automatically performs client-side and server-side validation.

Client-side development

ASP.NET Core integrates seamlessly with popular client-side frameworks and libraries, including Blazor, Angular, React, and Bootstrap. For more information, see ASP.NET Core Blazor and related topics under *Client-side development*.

2.ASP.NET CORE MVC:

ASP.NET Core MVC provides features to build web APIs and web apps:

- The Model-View-Controller (MVC) pattern helps make your web APIs and web apps testable.
- Razor Pages is a page-based programming model that makes building web UI easier and more productive.
- Razor markup provides a productive syntax for Razor Pages and MVC views.
- Tag Helpers enable server-side code to participate in creating and rendering HTML elements in Razor files.
- Built-in support for multiple data formats and content negotiation lets your web APIs reach a broad range of clients, including browsers and mobile devices.
- Model binding automatically maps data from HTTP requests to action method parameters.
- Model validation automatically performs client-side and server-side validation.

3. MODEL VIEW CONTROLLER (MVC):

MVC is a design pattern used to decouple user-interface (view), data (model), and application logic (controller). This pattern helps to achieve separation of concerns.

Using the MVC pattern for websites, requests are routed to a Controller that is responsible for working with the Model to perform actions and/or retrieve data. The Controller chooses the View to display and provides it with the Model. The View renders the final page, based on the data in the Model.

The three parts of the MVC software-design pattern can be described as follows:

1. Model: Manages data and business logic.
2. View: Handles layout and display.
3. Controller: Routes commands to the model and view parts.

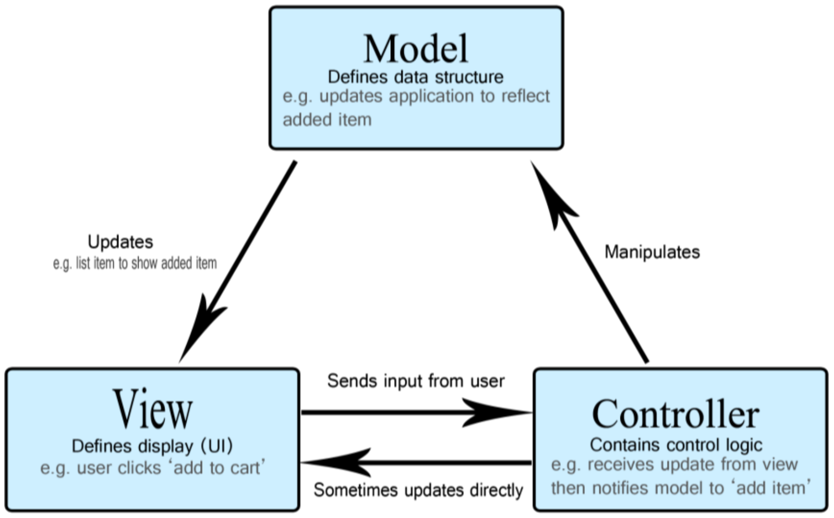

Model
Defines data structure
e.g. updates application to reflect added item
Updates
e.g. list item to show added item
Manipulates
View
Defines display (UI)
e.g. user clicks 'add to cart'
Sends input from user
Sometimes updates directly
Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'

Fig 1.1

Models & data

Create clean model classes and easily bind them to your database. Declaratively define validation rules, using C# attributes, which are applied on the client and server.

ASP.NET supports many database engines including SQLite, SQL Server, MySQL, PostgreSQL, DB2 and more, as well as non-relational stores such as MongoDB, Redis, and Azure Cosmos DB.

Model Responsibilities:

The Model in an MVC application represents the state of the application and any business logic or operations that should be performed by it. Business logic should be encapsulated in the model, along with any implementation logic for persisting the state of the application. Strongly-typed views typically use ViewModel types designed to contain the data to display on that view. The controller creates and populates these ViewModel instances from the model.

View Responsibilities:

Views are responsible for presenting content through the user interface. They use the Razor view engine to embed .NET code in HTML markup. There should be minimal logic within views, and any logic in them should relate to presenting content. If you find the need to perform a great deal of logic in view files in order to display data from a complex model, consider using a View Component, ViewModel, or view template to simplify the view.

Controller

The controller contains logic that updates the model and/or view in response to input from the users of the app.

So, for example, our shopping list could have input forms and buttons that allow us to add or delete items. These actions require the model to be updated, so the input is sent to the controller, which then manipulates the model as appropriate, which then sends updated data to the view.

You might however also want to just update the view to display the data in a different format, e.g., change the item order to alphabetical, or lowest to highest price. In this case the controller could handle this directly without needing to update the model.

Controller Responsibilities:

Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. In the MVC pattern, the controller is the initial entry point, and is responsible for selecting which model types to work with and which view to render (hence its name - it controls how the app responds to a given request).

4.RAZOR

The Razor syntax provides a simple, clean, and lightweight way to render HTML content based on your view. Razor lets you render a page using C#, producing fully HTML5 compliant web pages.

Razor Page is similar to the HTML page but it loads data easily. A Razor Page is almost the same as ASP.NET MVC's view component. It has basically the syntax and functionality same as MVC.

The basic difference between Razor pages and MVC is that the model and controller code is also added within the Razor Page itself. You do not need to add code separately.

It is similar to MVVM (Model-View-View-Model) framework. That provides two-way data binding and a simpler development experience with isolated concerns.
The ASP.NET MVC has been extremely popular nowadays for web application development, and it definitely has lots of advantages. In fact, the ASP.NET Web Forms was particularly designed as an MVVM solution in MVC.

In order to use dynamic web site, that is one where the content is regularly being added to, you have a number of options available to you. You can use a Content Management System (CMS), of which there are many to choose from including WordPress, Umbraco, Joomla!, Drupal, Orchard CMS and so on. Or you can hire someone to build a suitable site for you. Or you can build your own if you have an interest in, and an aptitude for programming.

If you choose to build your own, you can choose from a wide range of programming languages and frameworks. If you are a beginner, you will probably want to start with a framework and language that is easy to learn, well supported and robust. If you are considering making a career as a programmer, you probably want to know that the skills you acquire while learning your new framework will enhance your value to potential employers. In both cases, learning C# as a language and ASP.NET Core as a framework will tick those boxes. If you are a seasoned developer, the Razor Pages framework is likely to add to your skillset with the minimum amount of effort.

Advantages of Razor Pages:

A few very important advantages of Razor pages,

1. It supports cross-platform; hence it can be deployed on Windows, Unix, and Mac operating systems.
2. It is easy to learn.

3. Lightweight and flexible framework so it can fit with any application you want to build.
4. Can work with C# programming language with Razor markup.
5. More organized with code behind page like asp.net web forms.

5.SCAFOLDED IDENTITY IN ASP.NET CORE:

ASP.NET Core provides ASP.NET Core Identity as a Razor Class Library. Applications that include Identity can apply the scaffolder to selectively add the source code contained in the Identity Razor Class Library (RCL). You might want to generate source code so you can modify the code and change the behavior. For example, you could instruct the scaffolder to generate the code used in registration. Generated code takes precedence over the same code in the Identity RCL

Applications that do **not** include authentication can apply the scaffolder to add the RCL Identity package. You have the option of selecting Identity code to be generated.

Although the scaffolder generates most of the necessary code, you need to update your project to complete the process. This document explains the steps needed to complete an Identity scaffolding update.

We recommend using a source control system that shows file differences and allows you to back out of changes. Inspect the changes after running the Identity scaffolder.

Services are required when using Two Factor Authentication, Account confirmation and password recovery, and other security features with Identity. Services or service stubs aren't generated when scaffolding Identity. Services to enable these features must be added manually. For example, see Require Email Confirmation.Typically, apps that were created with individual accounts should ***not*** create a new data context.

6.ASP.NET CORE AUTHENTICATION:

Authentication is the process of determining a user's identity. Authorization is the process of determining whether a user has access to a resource. In ASP.NET Core, authentication is handled by the authentication service, IAuthenticationService, which is used by authentication middleware. The authentication service uses registered authentication handlers to complete authentication-related actions. Examples of authentication-related actions include:

- Authenticating a user.
- Responding when an unauthenticated user tries to access a restricted resource.

7.ASP.NET CORE AUTHORIZATION:

- Authorization refers to the process that determines what a user is able to do. For example, an administrative user is allowed to create a document library, add documents, edit documents, and delete them. A non-administrative user working with the library is only authorized to read the documents.
- Authorization is orthogonal and independent from authentication. However, authorization requires an authentication mechanism. Authentication is the process of ascertaining who a user is. Authentication may create one or more identities for the current user.

Authorization types

- Simple authorization
- Role-based authorization
- Claims-based authorization
- Policy-based authorization

SIMPLE AUTHORIZATION:

Authorization in ASP.NET Core is controlled with AuthorizeAttribute and its various parameters. In its most basic form, applying the [Authorize] attribute to a controller, action, or Razor Page, limits access to that component to authenticated users.

ROLE BASED AUTHORIZATION:

When an identity is created, it may belong to one or more roles. For example, Tracy may belong to the Administrator and User roles while Scott may only belong to the User role. How these roles are created and managed depends on the backing store of the authorization process. Roles are exposed to the developer through the IsInRoles method on the ClaimsPrincipal class.AddRoles must be Role services.

While roles are claims, not all claims are roles. Depending on the identity issuer a role may be a collection of users that may apply claims for group members, as well as an actual claim on an identity. However, claims are meant to be information about an individual user. Using roles to add claims to a user can confuse the boundary between the user and their individual claims.

CLAIMS BASED AUTHORIZATION:

When an identity is created it may be assigned one or more claims issued by a trusted party. A claim is a name value pair that represents what the subject is, not what the subject can do. For example, you may have a driver's license, issued by a local driving license authority. Your driver's license has your date of birth on it. In this case the claim name would be DateOfBirth, the claim value would be your date of birth, for example 8th June 1970 and the issuer would be the driving license authority. Claims based authorization, at its simplest, checks the value of a claim and allows access to a resource based upon that value. For example if you want access to a night club the authorization process might be:

The door security officer would evaluate the value of your date of birth claim and whether they trust the issuer (the driving license authority) before granting you access.

An identity can contain multiple claims with multiple values and can contain multiple claims of the same type.

Adding claims checks:

Claim based authorization checks:

- Are declarative.
- Are applied to Razor Pages, controllers, or actions within a controller.
- Cann***ot*** be applied at the Razor Page handler level, they must be applied to the Page.

Claims in code specify claims which the current user must possess, and optionally the value the claim must hold to access the requested resource. Claims requirements are policy based, the developer must build and register a policy expressing the claims requirements.The simplest type of claim policy looks for the presence of a claim and doesn't check the value.

POLICY BASED AUTHORIZATION:

Underneath the covers, role-based authorization and claims-based authorization use a requirement, a requirement handler, and a preconfigured policy. These building blocks support the expression of authorization evaluations in code. The result is a richer, reusable, testable authorization structure.

8.MODEL VALIDATION:

Model validation is the process of checking whether the user input is suitable for model binding and if not it should provide useful error messages to the user. The first part is to ensure that only valid entries are made. This should filter inputs which don’t make any sense. This could be a birth date in the future or an appointment date in the past. As important as checking for valid data is to inform the user about the wrong input and help him to enter the information in the expected form. Without any help, it can be frustrating and annoying for the user which might end up in losing a potential customer.

9.MIGRATION:

Migration enables you to change the data model and deploy your changes to production by updating the database schema without having to drop and re-create the database.

In real world projects, data models change as features get implemented: new entities or properties are added and removed, and database schemas need to be changed accordingly to be kept in sync with the application. The migrations feature in EF Core provides a way to incrementally update the database schema to keep it in sync with the application's data model while preserving existing data in the database. At a high level, migrations function in the following way:

- When a data model change is introduced, the developer uses EF Core tools to add a corresponding migration describing the updates necessary to keep the database schema in sync. EF Core compares the current model against a snapshot of the old model to determine the differences, and generates migration source files; the files can be tracked in your project's source control like any other source file.

- Once a new migration has been generated, it can be applied to a database in various ways. EF Core records all applied migrations in a special history table, allowing it to know which migrations have been applied and which haven't.

10.CODE-FIRST APPROACH:

Entity Framework introduced the Code-First approach with Entity Framework 4.1. Code-First is mainly useful in Domain Driven Design. In the Code-First approach, you focus on the domain of your application and start creating classes for your domain entity rather than design your database first and then create the classes which match your database design. The following figure illustrates the code-first approach.

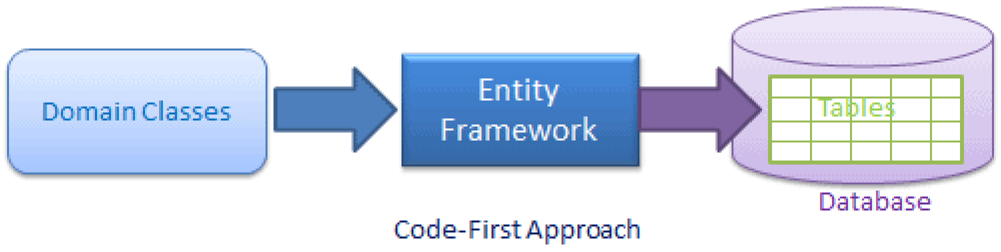


Fig 1.2

As you can see in the above figure, EF API will create the database based on your domain classes and configuration. This means you need to start coding first in C# or VB.NET and then EF will create the database from your code.

Code-First Workflow: The following figure illustrates the code-first development workflow.

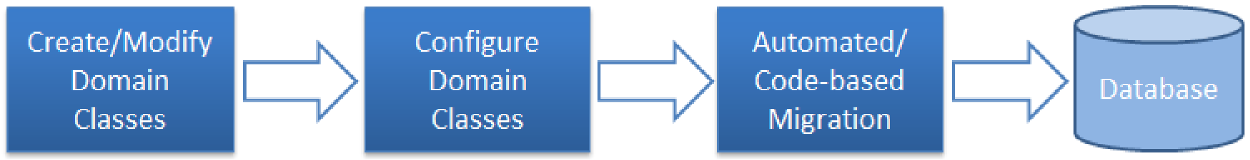


Fig 1.3

The development workflow in the code-first approach would be: Create or modify domain classes -> configure these domain classes using Fluent-API or data annotation attributes -> Create or update the database schema using automated migration or code-based migration.

11.DEPENDENCY INJECTION

ASP.NET Core has built-in support for dependency injection (DI). In ASP.NET Core MVC, controllers can request needed services through their constructors, allowing them to follow the Explicit Dependencies Principle.

Advantages of using ASP.NET Core Dependency Injection:

The ASP.NET Core Dependency Injection allows us to develop loosely coupled software components. Using the ASP.NET Core Dependency Injection, it is very easy to swap with a different implementation of a component.
In the next article, I am going to discuss the **Controllers in ASP.NET Core MVC** application. Here, in this article, I try to explain the **ASP.NET Core Dependency Injection** with an example. I hope this article will help you to understand the concept of Dependency Injection in ASP.NET Core Application.

12.ROUTING

ASP.NET Core MVC is built on top of ASP.NET Core's routing, a powerful URL-mapping component that lets you build applications that have comprehensible and searchable URLs. This enables you to define your application's URL naming patterns that work well for search engine optimization (SEO) and for link generation, without regard for how the files on your web server are organized. You can define your routes using a convenient route template syntax that supports route value constraints, defaults and optional values.

When you create a new ASP.NET MVC application, the application is already configured to use ASP.NET Routing. ASP.NET Routing is setup in two places.

First, ASP.NET Routing is enabled in your application's Web configuration file (Web.config file). There are four sections in the configuration file that are relevant to routing: the system.web.httpModules section, the system.web.httpHandlers section, the system.webserver.modules section, and the system.webserver.handlers' section. Be careful not to delete these sections because without these sections routing will no longer work.

Second, and more importantly, a route table is created in the application's Global.asax file. The Global.asax file is a special file that contains event handlers for ASP.NET application lifecycle events. The route table is created during the Application Start event.

13.TAG HELPERS

Tag Helpers enable server-side code to participate in creating and rendering HTML elements in Razor files. For example, the built-in ImageTagHelper can append a version number to the image name. Whenever the image changes, the server generates a new unique version for the image, so clients are guaranteed to get the current image (instead of a stale cached image). There are many built-in Tag Helpers for common tasks - such as creating forms, links, loading assets and more - and even more available in public GitHub repositories and as NuGet packages. Tag Helpers are authored in C#, and they target HTML elements based on element name, attribute name, or parent tag. For example, the built-in LabelTagHelper can target the HTML <label> element when the LabelTagHelper attributes are applied. If you're familiar with HTML Helpers, Tag Helpers reduce the explicit transitions between HTML and C# in Razor views. In many cases, HTML Helpers provide an alternative approach to a specific Tag Helper, but it's important to recognize that Tag Helpers don't replace HTML Helpers and there's not a Tag Helper for each HTML Helper. Tag Helpers compared to HTML Helpers explains the differences in more detail.

An HTML-friendly development experience.For the most part, Razor markup using Tag Helpers looks like standard HTML. Front-end designers conversant with HTML/CSS/JavaScript can edit Razor without learning C# Razor syntax.

A rich IntelliSense environment for creating HTML and Razor markupThis is in sharp contrast to HTML Helpers, the previous approach to server-side creation of markup in Razor views. Tag Helpers compared to HTML Helpers explains the differences in more detail. IntelliSense support for Tag Helpers explains the IntelliSense environment. Even developers experienced with Razor C# syntax are more productive using Tag Helpers than writing C# Razor markup.

A way to make you more productive and able to produce more robust, reliable, and maintainable code using information only available on the server

For example, historically the mantra on updating images was to change the name of the image when you change the image. Images should be aggressively cached for performance reasons, and unless you change the name of an image, you risk clients getting a stale copy. Historically, after an image was edited, the name had to be changed and each reference to the image in the web app needed to be updated. Not only is this very labor intensive, it's also error prone (you could miss a reference, accidentally enter the wrong string, etc.) The built-in ImageTagHelper can do this for you automatically. The ImageTagHelper can append a version number to the image name, so whenever the image changes, the server automatically generates a new unique version for the image. Clients are guaranteed to get the current image. This robustness and labor savings comes essentially free by using the ImageTagHelper.

Most built-in Tag Helpers target standard HTML elements and provide server-side attributes for the element. For example, the <input> element used in many views in the *Views/Account* folder contains the asp-for attribute.

14.IIS EXPRESS

IIS Express is a lightweight, self-contained version of IIS optimized for developers. IIS Express makes it easy to use the most current version of IIS to develop and test websites. It has all the core capabilities of IIS 7 and above as well as additional features designed to ease website development including:

- It doesn't run as a service or require administrator user rights to perform most tasks.
- IIS Express works well with ASP.NET and PHP applications.
- Multiple users of IIS Express can work independently on the same computer.

15.ENTITY FRAMEWORK CORE:

Entity Framework (EF) Core is a lightweight, extensible, open source and cross-platform version of the popular Entity Framework data access technology.

EF Core can serve as an object-relational mapper (O/RM), which:

- Enables .NET developers to work with a database using .NET objects.
- Eliminates the need for most of the data-access code that typically needs to be written.

16.GOOGLE MAPS API:

16.1 Maps Java script API **:**The Maps JavaScript API lets you customize maps with your own content and imagery for display on web pages and mobile devices. The Maps JavaScript API features four basic map types (roadmap, satellite, hybrid, and terrain) which you can modify using layers and styles, controls and events, and various services and libraries.

16.2 The Geolocation API :returns a location and accuracy radius based on information about cell towers and Wi-Fi nodes that the mobile client can detect. This document describes the protocol used to send this data to the server and to return a response to the client.

Communication is done over HTTPS using POST. Both request and response are formatted as JSON, and the content type of both is application/json.

16.3 The Places API is a service that returns information about places using HTTP requests. Places are defined within this API as establishments, geographic locations, or prominent points of interest.

16.4 **Geocoding** is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers on a map, or position the map.

Reverse geocoding is the process of converting geographic coordinates into a human-readable address.

You can also use the Geocoding API to find the address for a given place ID.

17. C#:

C# (pronounced "See Sharp") is a modern, object-oriented, and type-safe programming language. C# enables developers to build many types of secure and robust applications that run in .NET. C#

has its roots in the C family of languages and will be immediately familiar to C, C++, Java, and JavaScript programmers. This tour provides an overview of the major components of the language in C# 8 and earlier. If you want to explore the language through interactive examples, try the introduction to C# tutorials.

C# is an object-oriented, component-oriented programming language. C# provides language constructs to directly support these concepts, making C# a natural language in which to create and use software components. Since its origin, C# has added features to support new workloads and emerging software design practices. At its core, C# is an object-oriented language. You define types and their behavior.

18.Javascript:

JavaScript (**JS**) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. Read more about JavaScript.

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about APIs that are specific to Web pages, please see Web APIs and DOM.

The standards for JavaScript are the ECMAScript Language Specification (ECMA-262) and the ECMAScript Internationalization API specification (ECMA-402). As soon as one browser implements a feature, we try to document it. This means that cases where some proposals for new ECMAScript features have already been implemented in browsers, documentation and examples in MDN articles may use some of those new features. Most of the time, this happens between the stages 3 and 4, and is usually before the spec is officially published.

Do not confuse JavaScript with the Java programming language. Both "Java" and "JavaScript" are trademarks or registered trademarks of Oracle in the U.S. and other countries. However, the two programming languages have very different syntax, semantics, and use.

19.HTML:

At its heart, HTML is a language made up of elements, which can be applied to pieces of text to give them different meaning in a document (Is it a paragraph? Is it a bulleted list? Is it part of a table?), structure a document into logical sections (Does it have a header? Three columns of content? A navigation menu?), and embed content such as images and videos into a page. This module will introduce the first two of these and introduce fundamental concepts and syntax you need to know to understand HTML.

20.CSS:

Cascading Style Sheets (**CSS**) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the **open web** and is standardized across Web browsers according to W3C specifications. Previously, the development of various parts of CSS specification was done synchronously, which allowed the versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, or even CSS3. There will never be a CSS3 or a CSS4; rather, everything is now CSS without a version number.

After CSS 2.1, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to develop and release recommendations separately per module. Instead of versioning the CSS specification, W3C now periodically takes a snapshot of the latest stable state of the CSS specification and individual modules progress. CSS modules now have version numbers, or levels, such as CSS Color Module Level 5.

21.BOOTSTAP: Bootstrap makes front-end web development faster and easier. It's made for folks of all skill levels, devices of all shapes, and projects of all sizes. Bootstrap is open source. It's hosted, developed, and maintained on GitHub. Bootstrap is downloadable in two forms, within which you'll find the following directories and files, logically grouping common resources and providing both compiled and minified variations. With **Bootstrap**, you get extensive and beautiful **documentation** for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins

22.BOOTSWATCH:

It is an open-source project, that provides a number of free themes for bootstrap that a web developer can use. It helps the developer to get proper UI without spending hours and energy on styling different elements.

23.JQUERY

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

24.VISUAL STUDIO 2022:

An integrated development environment (IDE) is a feature-rich application that can be used for many aspects of software development. The Visual Studio IDE makes it easy to edit, debug, build, and publish your app.

An *integrated development environment* (IDE) is a feature-rich program that supports many aspects of software development. The Visual Studio IDE is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and many more features to enhance the software development process.

The preceding image shows Visual Studio with an open project that shows key windows and their functionality:

- In Solution Explorer, at upper right, you can view, navigate, and manage your code files. **Solution Explorer** can help organize your code by grouping the files into solutions and projects.
- The central editor window, where you'll probably spend most of your time, displays file contents. In the editor window, you can edit code or design a user interface such as a window with buttons and text boxes.
- In Git Changes at lower right, you can track work items and share code with others by using version control technologies like Git and GitHub.

25.SSMS 2019:

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure. Use SSMS to access, configure, manage, administer, and develop all components of SQL Server, Azure SQL Database , Azure SQL Managed Instance, SQL Server on Azure VM, and Azure Synapse Analytics. SSMS provides a single comprehensive utility that combines a broad group of graphical tools with many rich script editors to provide access to SQL Server for developers and database administrators of all skill levels.

MS SQL Server Management Studio is a workstation or a client tool which is used to connect to and manage your SQL Server

SQL Server Management Studio (SSMS) is a **windows software or a client tool** used to connect and work with our SQL Server from a **graphical interface** instead of using the command line. Microsoft SQL Server 2005 launched the management studio to work with SQL Server and Azure SQL databases.

It allows DBAs and database developers to **configure, manage**, and **administer** all components within SQL Server. Its main functionality is to create databases and tables, execute SQL queries for inserting, updating, and deleting data, creating and managing stored procedures, triggers, views, and cursors. It also enables us to set privileges (securities) on databases and their objects.

SSMS also includes tools for deployment, database health monitoring, and reporting. It includes **SQL Profiler**, which allows us to examine the performance of our SQL databases. It's also possible to use it to schedule background work. If we want to connect to a remote SQL Server instance, we'll need this GUI tool or similar software. It is used by Administrators, Developers, Testers, etc.

26.GITHUB:

GitHub is an online software development platform used for storing, tracking, and collaborating on software projects. It enables developers to upload their own code files and to collaborate with fellow developers on open-source projects. GitHub also serves as a social networking site in which developers can openly network, collaborate, and pitch their work.

GitHub allows software developers and engineers to create remote, public-facing repositories on the cloud for free. Once you've set up a repository on GitHub, you can copy it to your device, add and modify files locally, then "push" your changes back to the repository where your changes are displayed to the public.

4.DETAILS OF DESIGN, WORKING AND PROCESSES

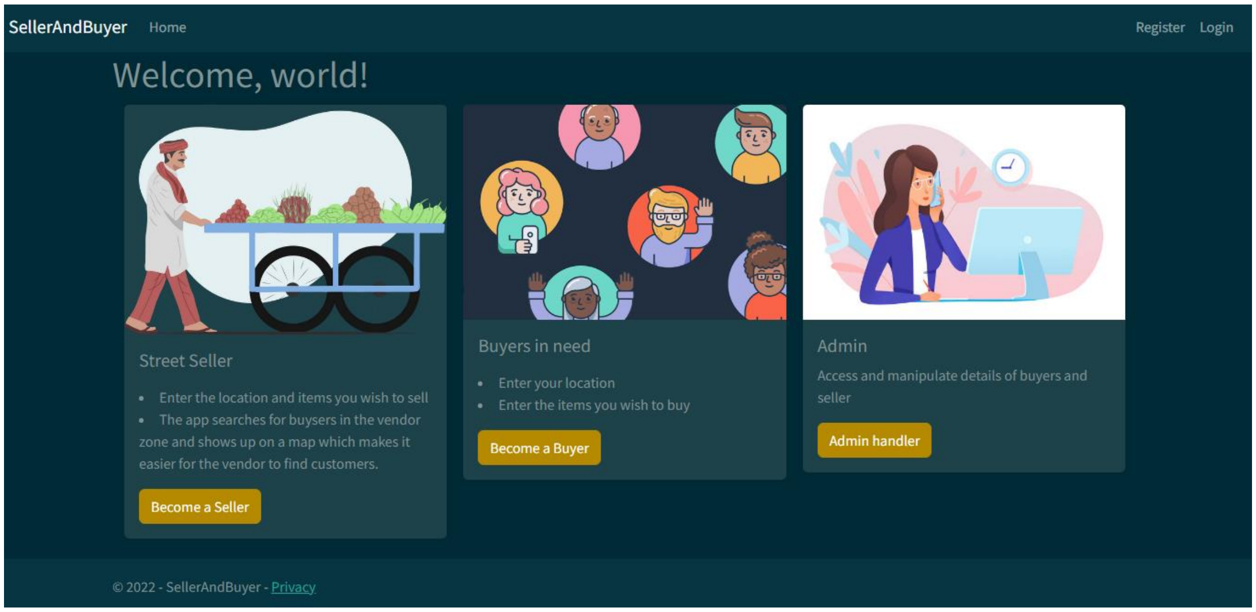
Details of working :

Street vendors have become an integral part of urban economies all over the globe. A street vendor is broadly defined as a person who offers access to a wide range of goods and services with low price.

The main users in the system include:

- **Sellers**
- **Buyers**
- **Admin**
- **Unregistered users**

The common page to all users includes Login, Register and home on the nav bar menu where a user can register himself as a seller, buyer or Admin.(Fig 1.4) Unregistered users can only access the home page and can also login if an account is registered.



- Enter the location and items you wish to sell
- The app searches for buysers in the vendor zone and shows up on a map which makes it easier for the vendor to find customers.

- Enter your location
- Enter the items you wish to buy

Access and manipulate details of buyers and seller

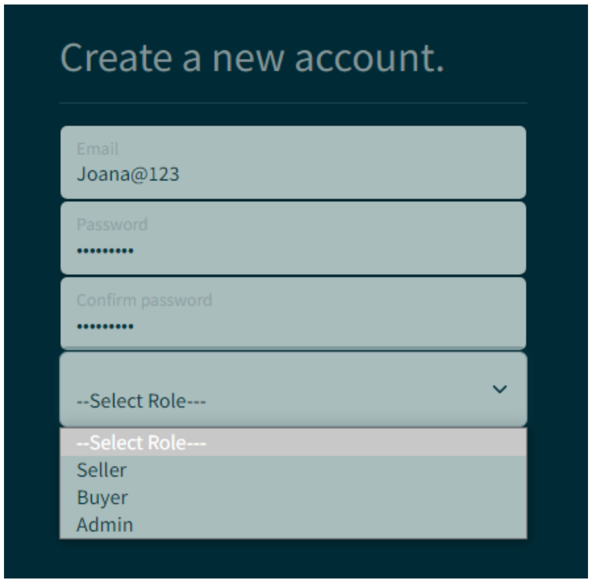
© 2022 - SellerAndBuyer - Privacy

Fig 1.4

4.1 Register module:

A Seller produces and sells goods and services at affordable rates. Sellers who wish to sell their goods must register themselves on the webpage by adding their name, password and the type of user they are from the select role drop down list.   (Fig 1.5)

A buyer is the one in need of goods that are sold by a vendor at an affordable price. The buyer must register himself on the webpage by adding their name, password and the type of user they are from the select role drop down list in order to make any orders or request the seller for goods.


Create a new account.

Email
Joana@123

Password
•••••••••

Confirm password
•••••••••

--Select Role---

Fig 1.5

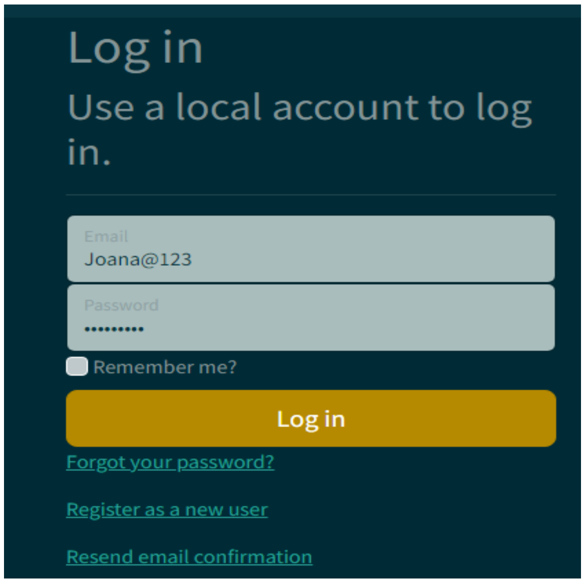
An Admin manages all the users on the system which include sellers and buyers. An admin must first be registered before he can claim access to the system by adding their name, password and the type of user they are from the select role drop down list.

While registering for an account:

- The password and confirm passwords must match
- The same email cannot be registered twice
- Email must follow the email rules

4.2 Login module:

Once the respective user is registered he is redirected to the login page where he is requested to login in order to access his details and place an order or sell items.


Log in

Use a local account to log in.

Email: Joana@123

Password: •••••••••

Remember me?

Fig 1.6

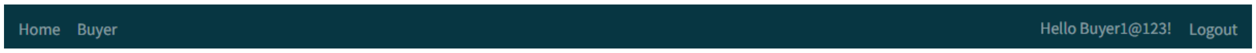
If the registered user has forgotten the password, the user can click on one of the following options:

- Create a new account
- Receive an email confirmation
- Click on forgot password

Role based Authorization is used to specify which roles the current user must be a member of in order to access the requested resource. The 3 roles in the system are Seller, buyer and Admin

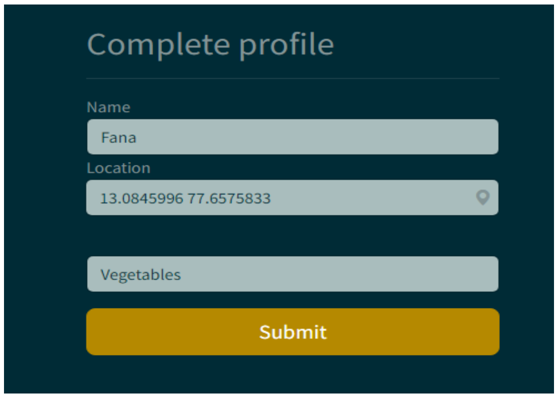
a] User role is Buyer:

The buyer is logged in, the buyer has access to the buyer page since he is part of the buyer role. The logged in buyer's email, buyer home and logout options shows on the menu bar.



If the buyer clicks on his profile name, he is redirected to his manage account page where he can update his password, phone number and email address.

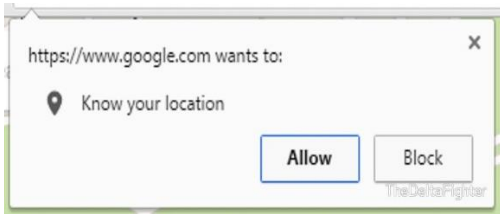
The buyer is requested to fill his user details his like Name, phone number, location and the goods he wants to buy. (Fig 1.7)

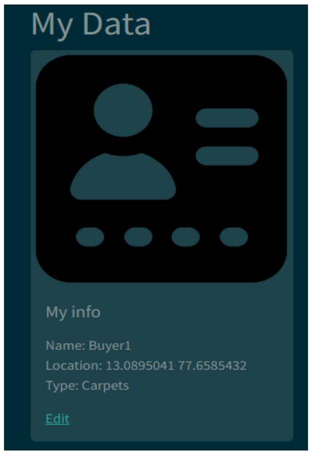

Complete profile

Name
Fana
Location
13.0845996 77.6575833
Vegetables

Fig 1.7

The app requests for the buyer's location using Google's live location tracking on the browser, If the buyer allows access to the current location by clicking on the location icon, the location co-ordinates are saved into the database. The user must allow access to track live location (fig 1.8)


https://www.google.com wants to:
Know your location


My Data

My info
Name: Buyer1
Location: 13.0895041 77.6585432
Type: Carpets

Fig 1.8

Fig 1.9

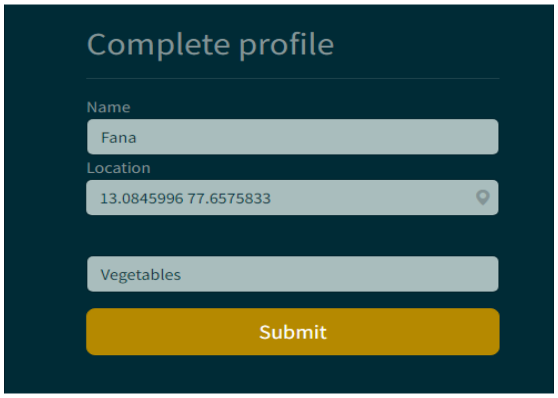
If the user has already filled in his details he can click on the edit link and edit his details. This edit feature is used if the buyer wants to come back looking for another type of seller, if wants to change his location, change the items he sells or change his name which is then updated in the database(fig 1.12).

b] User role is Seller:

If the seller is logged in and is part of the seller role, the user has access to the seller page which no other user has. The seller logs into his user account and is prompted to add in his details like Name, phone number, location and the items he wants to sell for the first login.

The app requests for the seller's live location using Google's location tracking on the browser, If the buyer allows access to the current location, the location co-ordinates are saved into the database.(fig1.10)

If the user has already filled in his details, all his details are displayed on a card with an edit and map option where the user can edit or update his details and find the list of buyers on a map.(fig 1.11) He can click on the edit link and edit his details. (fig 1.12)


Complete profile

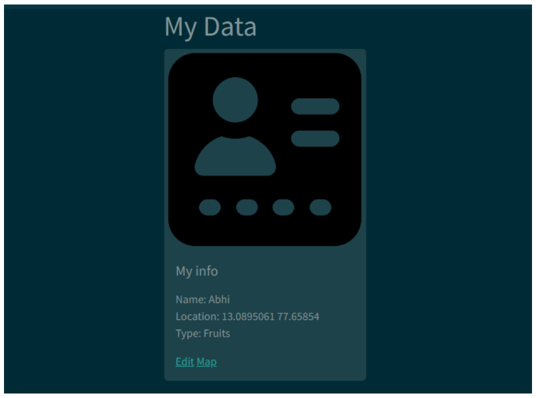
Name

Fana

Location

13.0845996 77.6575833

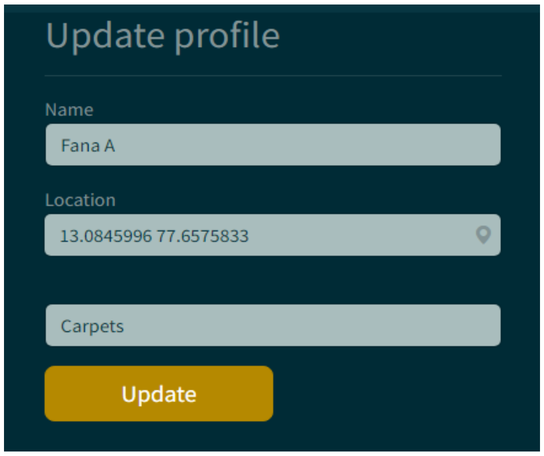
Vegetables


My Data

My info

Name: Abhi
Location: 13.0895061 77.65854
Type: Fruits

Fig 1.10 Fig 1.11


Update profile

Name

Fana A

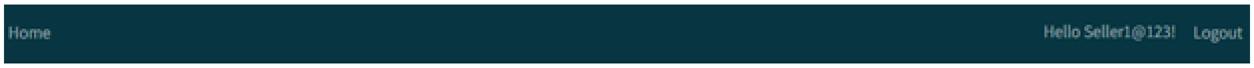
Location

13.0845996 77.6575833

Carpets

Fig 1.12

This edit feature is used if the seller wants to update his location, change the items he sells, change his name or location. The seller's name, seller link and logout options show on the menu bar.



The database is going to be checked for buyers where:

- All the buyers are within 10 kms within the seller's location
- The buyer's items are same as the items the seller sells

If a match is found then the list of buyers are shown on the seller's map along with their names in an info box with the help of a marker within a circle.

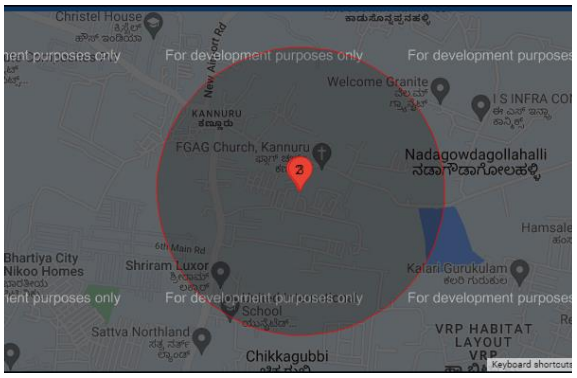


Fig 1.13

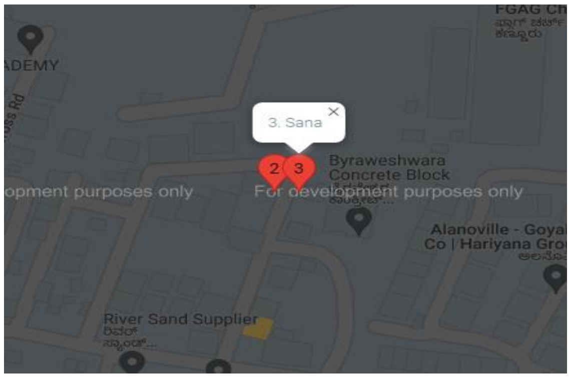


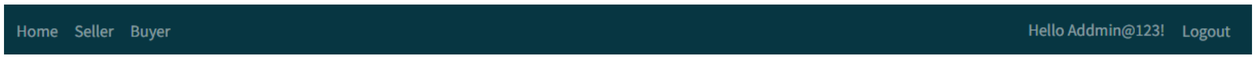
Fig 1.14

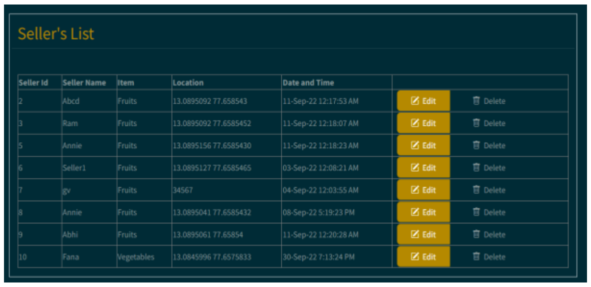
c].User logged in as Admin:

If an Admin logs in, he has access to the admin page. The admin has access to both buyer and seller page where he can update, edit and delete buyers or sellers.

When Seller is clicked the list of all the sellers are displayed in a table. (Fig 1.15)

When the Buyer option is clicked the list of all the buyers are displayed in a table(fig 1.16)

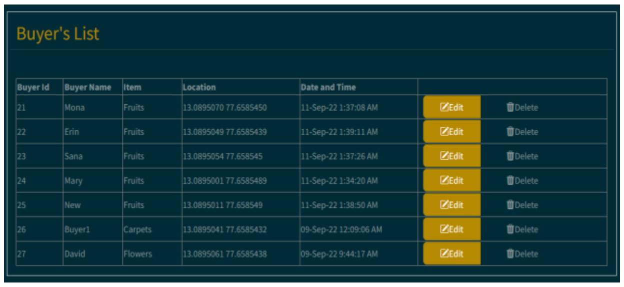




Seller's List

Seller Id	Seller Name	Item	Location	Date and Time		
2	Abcd	Fruits	13.0895092 77.658543	11-Sep-22 12:17:53 AM	Edit	Delete
3	Ram	Fruits	13.0895092 77.6585452	11-Sep-22 12:18:07 AM	Edit	Delete
5	Annie	Fruits	13.0895156 77.6585430	11-Sep-22 12:18:23 AM	Edit	Delete
6	Seller1	Fruits	13.0895127 77.6585465	03-Sep-22 12:08:21 AM	Edit	Delete
7	gv	Fruits	34567	04-Sep-22 12:03:55 AM	Edit	Delete
8	Annie	Fruits	13.0895041 77.6585432	08-Sep-22 5:19:23 PM	Edit	Delete
9	Abhi	Fruits	13.0895061 77.65854	11-Sep-22 12:20:28 AM	Edit	Delete
10	Fana	Vegetables	13.0845996 77.6575833	30-Sep-22 7:13:24 PM	Edit	Delete

Fig 1.15

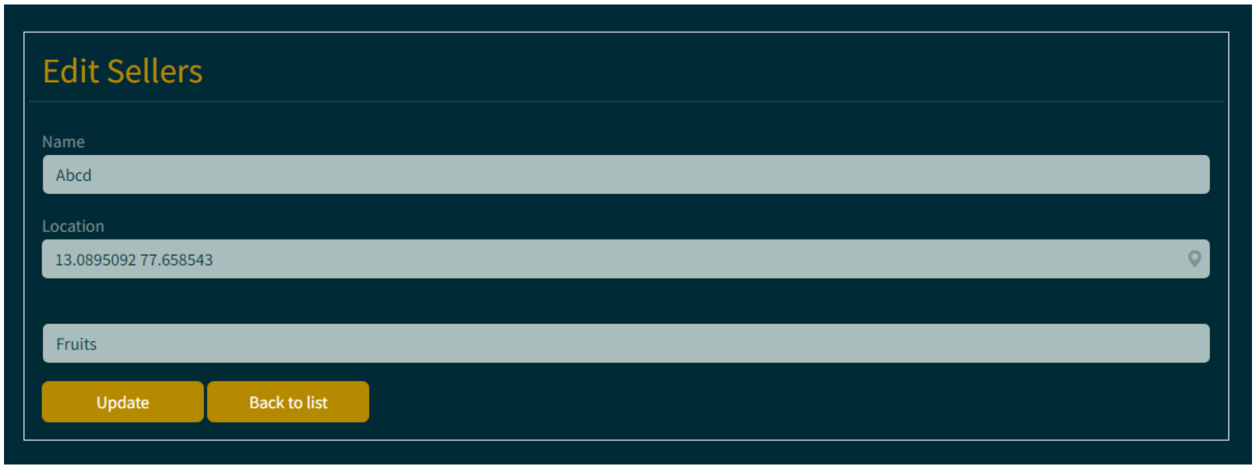


Buyer's List

Buyer Id	Buyer Name	Item	Location	Date and Time		
21	Mona	Fruits	13.0895070 77.6585450	11-Sep-22 1:37:08 AM	Edit	Delete
22	Erin	Fruits	13.0895049 77.6585439	11-Sep-22 1:39:11 AM	Edit	Delete
23	Sana	Fruits	13.0895054 77.658545	11-Sep-22 1:37:26 AM	Edit	Delete
24	Mary	Fruits	13.0895001 77.6585489	11-Sep-22 1:34:20 AM	Edit	Delete
25	New	Fruits	13.0895011 77.658549	11-Sep-22 1:38:50 AM	Edit	Delete
26	Buyer1	Carpets	13.0895041 77.6585432	09-Sep-22 12:09:06 AM	Edit	Delete
27	David	Flowers	13.0895061 77.6585438	09-Sep-22 9:44:17 AM	Edit	Delete

Fig 1.16

The table includes Id name, type, location, timestamp of account creation and edit and delete option. If edit option is clicked the Admin can edit the data of the seller/buyer. Once the update is complete, we can go back to the main previous page.



Edit Sellers

Name: Abcd

Location: 13.0895092 77.658543

Fruits

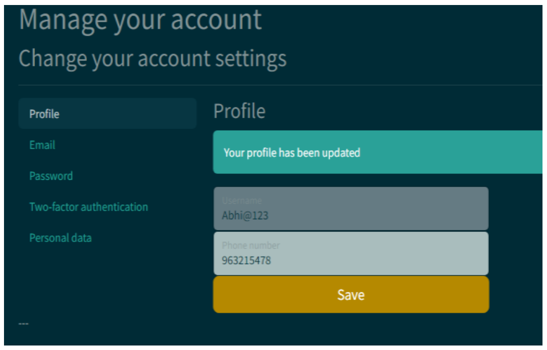
Fig 1.17

When delete button is clicked, the data of the respective is reset from the database(seller/buyer table). Thus the Admin is able to remove user or change buyer/sellers' details.

4.3 User Personal info:

The logged in users can access additional information by clicking on the username on the top right, where the email, phone number , password and other additional details can be added and changed. Users can click on their user's name on the top right menu and access all their personal details and edit them. Features include adding and updating phone number(fig), Changing email, changing password, changing email, print user data and delete user account.

User can add or update the phone number using the profile section and save it into the database.The email can be changed as well where a verfificaation email is sent to the users email account.

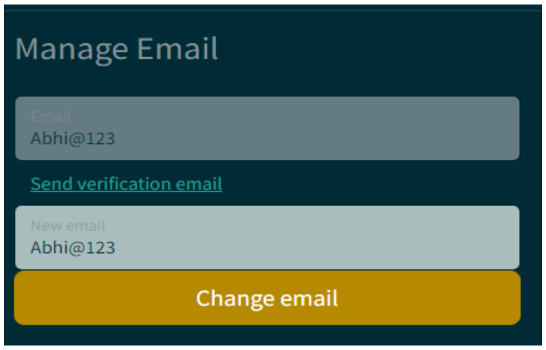


Your profile has been updated

Username: Abhi@123

Phone number: 963215478

Fig 1.18

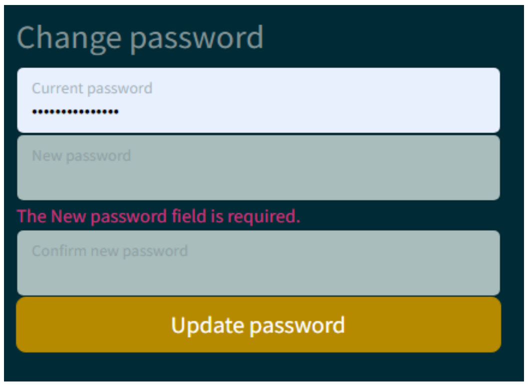


Email: Abhi@123

New email: Abhi@123

Fig 1.19

If the user wants to cange the password, it can be changed from this section but must follow the password rules, the password change requires that the current password is known and the new password is used from the database for the next login.Client side validation is added for chage password.



Current password: •••••••••••••••

New password

The New password field is required.

Confirm new password

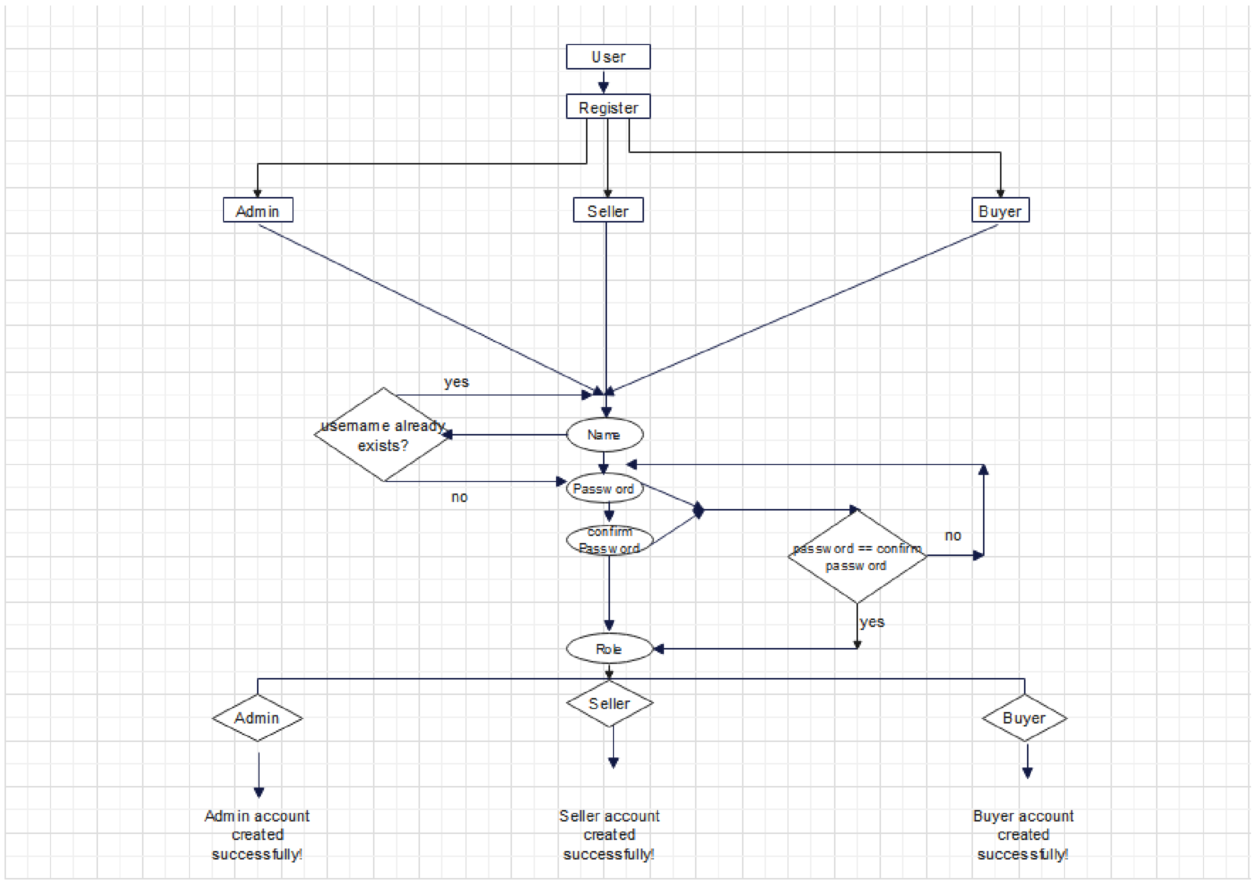
Fig 1.20

Advantages:

- User location is updated just by a single touch
- Makes life easier and hassle free for the seller
- Seller can find the buyer hotspot
- Update of data is much easier
- Admin can manage both sellers and buyers

5.ER DIAGRAM:

a]Register


User
Register
Admin
Seller
Buyer
yes
usemame already exists?
Name
no
Password
confirm Password
password == confirm password
no
yes
Role
Admin
Seller
Buyer
Admin account created successfully!
Seller account created successfully!
Buyer account created successfully!

Any user can register as any one on the role Seller, Buyer or Admin.

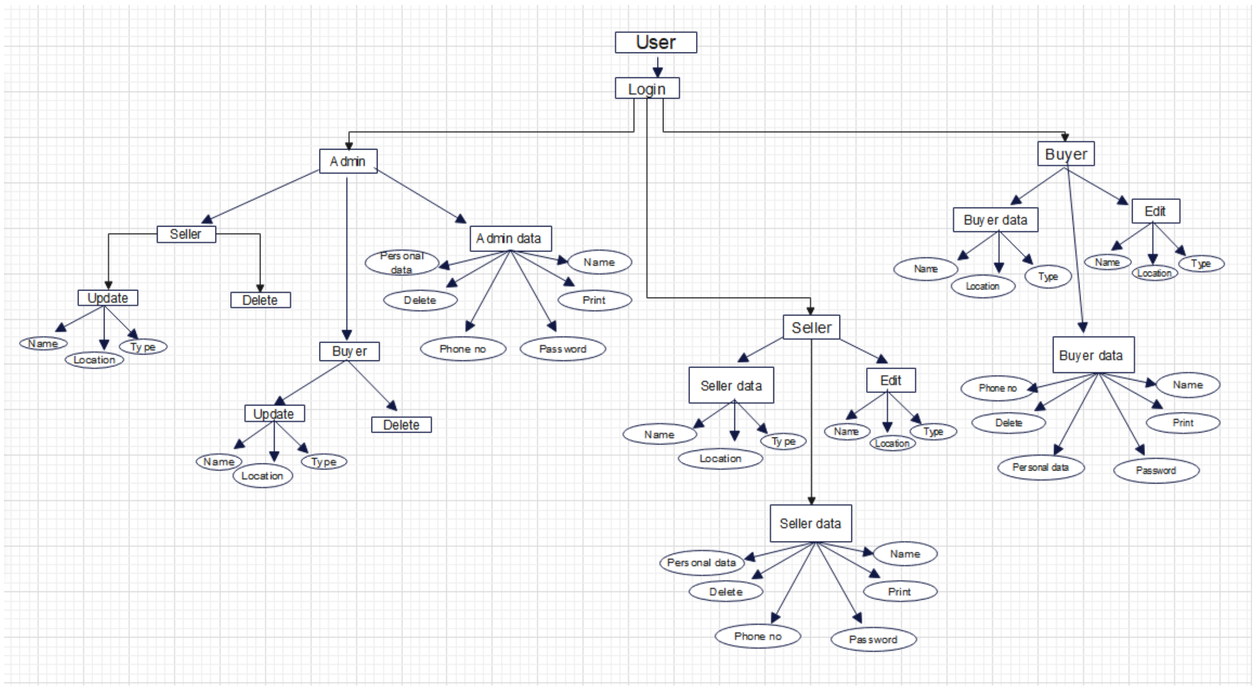
For creating an account, client-side validation is done where the user has to:

-fill in the email/name. If name already exists in the database the user will have to choose another name. If the name is valid:

-the password field and the confirm password must match, if it does not user will have to re-enter the password. If it passes the criteria:

-the user can select the type of role he wishes to be.

b]Login



Once the user has successfully created an account,

If the user is an Admin:

- He can update or delete the buyers/seller’s data
- He has access to modify and view his profile details

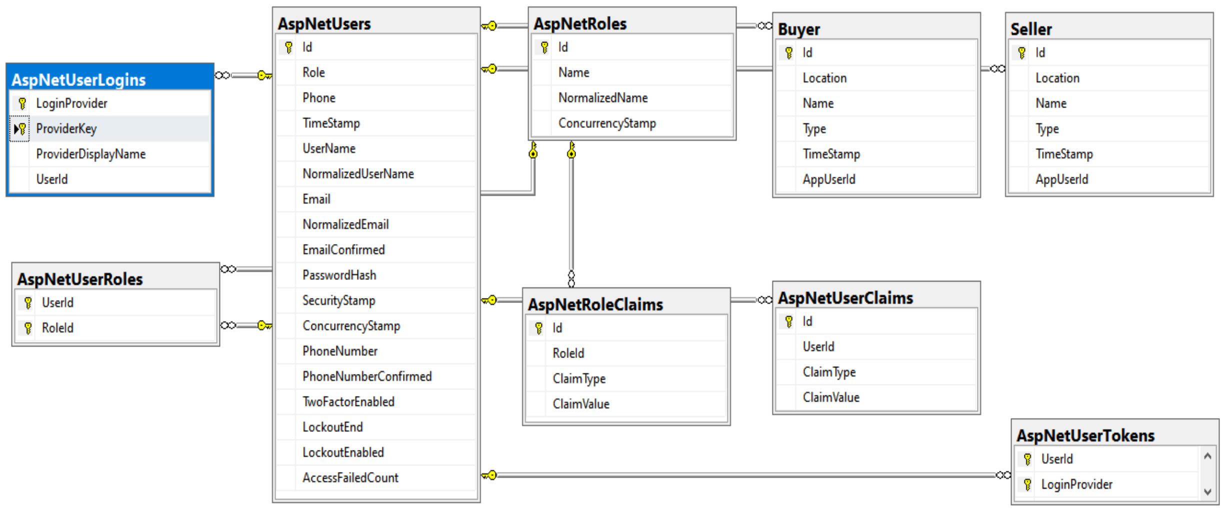
If the user is a seller

- He has access to his personal details
- His Seller data which can be modified
- The buyers on the map

If the user is a Buyer:

- He has access to his personal details
- His Buyer data which can be modified

6. DATABASE TABLES AND RELATIONSHIPS



AspNetUserLogins:

In Asp.net Identity, the Identity system uses the AspNetUserLogins table to hold information about 3rd party/external logins, for example users who login into your site via Google, Facebook, Twitter etc. For example if the user logs into your site via Facebook, then the LoginProvider is the name of the service which provided the login, so in this case "**Facebook**", the ProviderKey is a unique Facebook key associated with the user on Facebook.

-AspNetUsers:

The AspNetUsers table is the primary table to store user information, this is linked to AspNetUserLogins via UserId -> AspNetUsers.Id. This table is where the authenticated users are being saved.

-**AspNetRoles:** are a standard & common approach for implementing authorization in Applications. **Identity** can contain **roles** & **roles**, in turn, contain permissions for performing actions in the application. You can assign multiple roles to a user. When a user is created it can be linked to one or more roles.

-AspNetUserRoles:

This table is **a mapping table**. You can access the table using AspNetUser object or AspNetRole object. Assuming you want to show all users and their roles, you can do that by getting it from var userRoles = user. AspNetRoles; for each of the users.

-AspNetUserClaims:

What is AspNetUserClaims table?

AspNetUserClaims" table is **holding claims assigned to a user**. A claim is different from a role because a claim is a key-value pair. You can have a role or not have a role. Claim also provides a value for a specified claim

-AspNetRoleClaims:

AspNetRoleClaims" table is **holding claims assigned to a specific role**. "dbo. AspNetRoles" table is holding a list of roles. It is a lookup table of all possible roles that exist and can be assigned to a user

- AspNetUserTokens:

The table AspNetUserTokens is for **external authentication token storage** and is filled by SignInManager.

The table AspNetUserTokens is for external authentication token storage and is filled by SignInManager.UpdateExternalAuthenticationTokensAsync method. Internal authentication tokens are stored in memory by default and if you want to store them into the database, you must create your own table and your own logic to store.

-Buyer:

Has all the buyer details like location, buyer ID, the current time, Buyer name and buyer type.

-Seller:

Has all the seller details like location, buyer ID, the current time, seller name and seller type.

Entity type relationships

The entity types are related to each other in the following ways:

- Each User can have many UserClaims.
- Each User can have many UserLogins.
- Each User can have many UserTokens.
- Each Role can have many associated RoleClaims.
- Each User can have many associated Roles, and each Role can be associated with many Users. This is a many-to-many relationship that requires a join table in the database. The join table is represented by the UserRole entity.

7.RESULTS AND APPLICATIONS

Applications:

- Sellers can sell his items efficiently by finding the buyer zone.
- Buyers can easily add and edit their profile details
- Easy Admin page to manage sellers and buyers
- Sellers can see the buyers on the map
- Buyers or sellers can update their details after login if required
- User friendly
- Sellers enable their location and share the items they sell, Buyers share their location, contact and product that they want to sell to the system
- If the seller’s location, products matches the buyers requirements and location (Location calculated in radius).

8.CONCLUSION AND FUTURE SCOPE

By making goods and services available at doorsteps, or at places that are conveniently accessible, street vendors not only **reduce the transaction costs of everyday purchases**, but also play a significant role in increasing the labour hours and street vendors can become key enablers of economic growth and employment across urban landscapes.

9.REFERENCES

1. Asp.net MVC - https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-6.0
2. Google maps API - https://developers.google.com/maps/documentation/javascript/overview
3. Entity framework core - https://learn.microsoft.com/en-us/ef/core/get-started/overview/first-app?tabs=netcore-cli
4. SSMS (Microsoft SQL) - https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16
5. Identity scaffolding - https://learn.microsoft.com/en-us/aspnet/core/security/authentication/scaffold-identity?view=aspnetcore-6.0&tabs=visual-studio
6. Razor pages - https://learn.microsoft.com/en-us/aspnet/core/tutorials/razor-pages/razor-pages-start?view=aspnetcore-6.0&tabs=visual-studio
7. C# - https://www.w3schools.com/cs/index.php
8. JavaScript - https://developer.mozilla.org/en-US/docs/Web/JavaScript
9. CSS - https://developer.mozilla.org/en-US/docs/Web/CSS
10. Html - https://developer.mozilla.org/en-US/docs/Web/HTML
11. Bootswatch designs - https://bootswatch.com/
12. Bootstrap - https://getbootstrap.com/docs/4.1/getting-started/introduction/
13. Migrations - https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/migrations-and-deployment-with-the-entity-framework-in-an-asp-net-mvc-application
14. Visual Studio 2022 - https://visualstudio.microsoft.com/
15. SQL Server Management Studio - https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16