**WATCH MAIL IN THE WAY YOU WANT**

**Student Names:**

Name: LaxmaReddy Kandula (Dawg Type: 856525636)

Name: SriTeja (Dawg Type: 8565)

**CONTENTS:**

**ABSTRACT ................................................................................................................................ 1**

**INTRODUCTION ..................................................................................................................... 2**

**PROJECT REQUIREMENTS …............................................................................................ 3**

**PROJECT ARCHITECTURE ................................................................................................ 4**

**PROJECT IMPLEMENTATION ................................................................................................ 5**

**CONCLUSION …....................................................................................................................... 6**

**SOURCE CODE ........................................................................................................................ 7**

**1. ABSTRACT:**

The goal is to create a web-based application that users may access from any location. The application primarily targets larger enterprises and smaller organizations who deal with many mails every day. Given that the application has a form with attributes like

* The customer's organization's mail id and password (we even can provide same as service if we deal with security for the credentials).
* One dropdown for collecting the information regarding the format user wants to convert.
* We also collect the timestamp from which we want to convert the mails into the format we want to change.
* And finally, the submission button.

**2. INTRODUCTION:**

As an individual, we see lot of mails coming to our inbox's related to social media or promotions like bonus and many others. If this is the case, it is hard to go through each mail.

Let us say, if this scenario happens for organizations. If they get lot of mails into one mail id specific to recipients like front desk or any other in different scenarios.

So, we thought we would provide a web application which handles some services in which mail converter is one of the many to come.

**3. PROJECT REQUIREMENTS:**

During the continuous discussions regarding the requirements and what we want to develop. We found most of the requirements needed to develop the project.

Some Functional Requirements:

* We need the login page for the subscriber to access the services we will provide in the web application.
* For the user to get login access, we need to provide a source for the user to register to our web application to access the services.
* We need to provide a web form to enter mail information, from which the user is going to convert the mail inbox to the format chosen in a specific interval.

Some Non-Functional Requirements:

* Validation of the application URL navigation when the site gets crashed.
* Mails should be reflected after added by the user once the application gets refreshed.

**4. PROJECT ARCHITECTURE:**

We are using MVC architecture to develop the web application. We have used the stack below for developing the web application.

1. Visual Studio 2019

2. .NET 4.5 C# MVC

3. Entity Framework 6.0.0

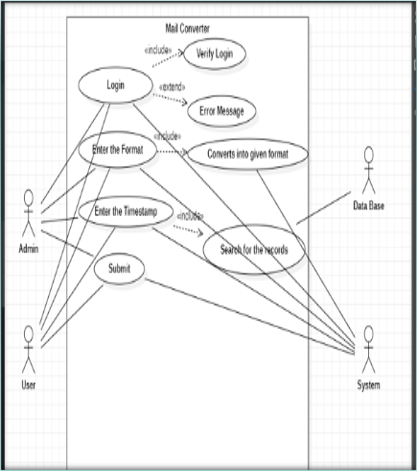
4. Bootstrap 5.4.1

5. Html

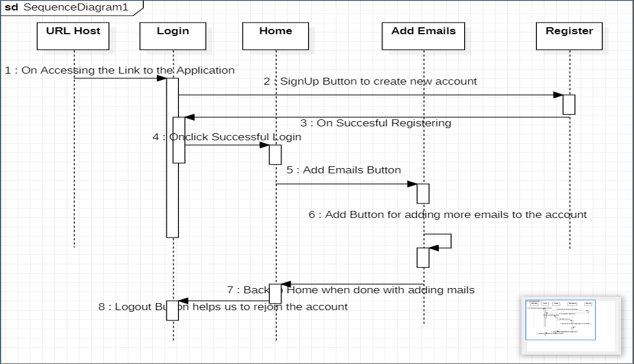
6. CSS

7. Javascript

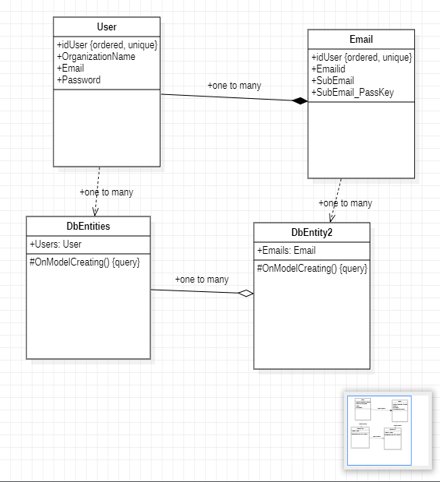
**Use Case Diagram:**



**Sequence Diagram:**



**Class Diagram**



**5. PROJECT IMPLEMENTATION:**

As we had chosen the Agile methodology. The priority user story which we decided to handle in our first sprint is to set up the environment for developing the web application.

Below are the details of our sprints:

Sprint1:

* UserStory1. 1: Gathering information for analysis and requirements.
* UserStory1.2: Making Environment ready like Information related to extensions for the project that support the following technologies:

1. .NET 4.5 C# MVC

2. Entity Framework 6.0.0

3. Bootstrap 5.4.1

4. Html

5. CSS

6. JavaScript

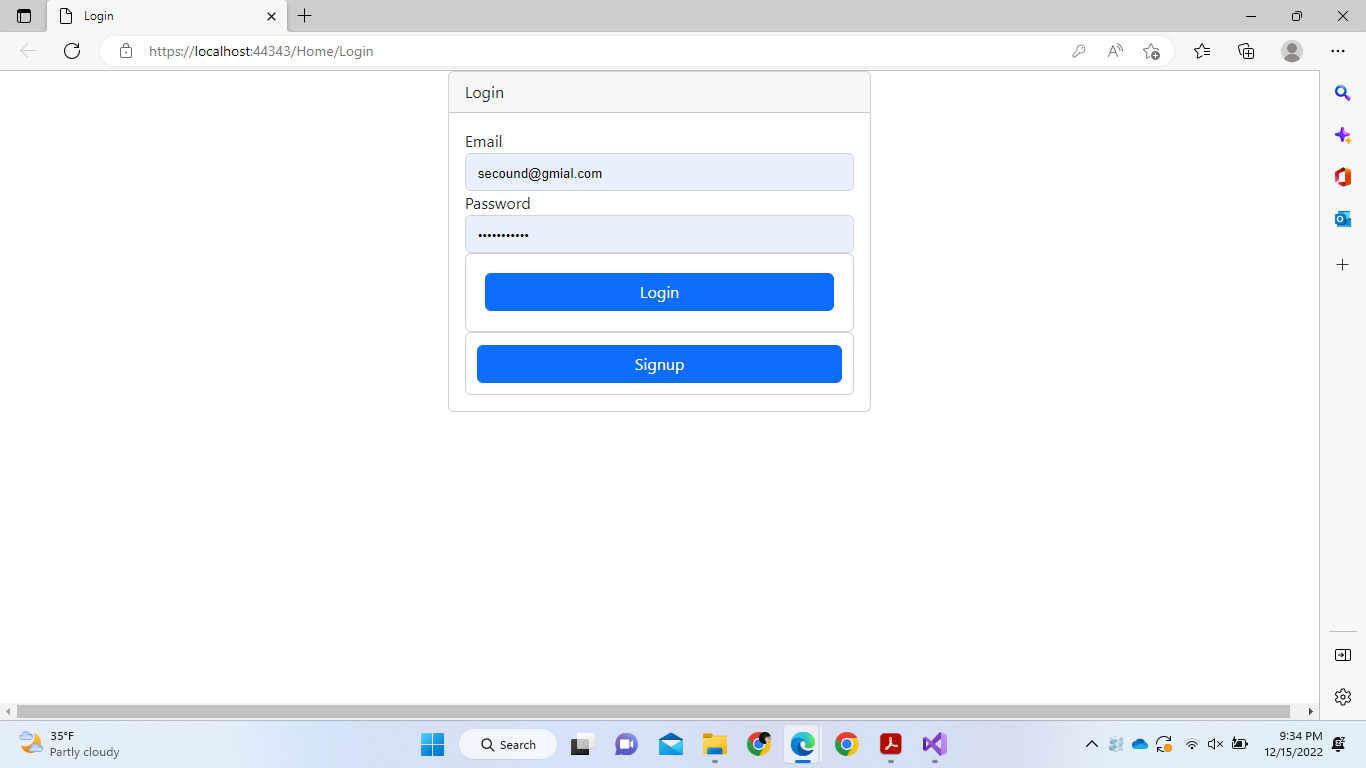
7. Git

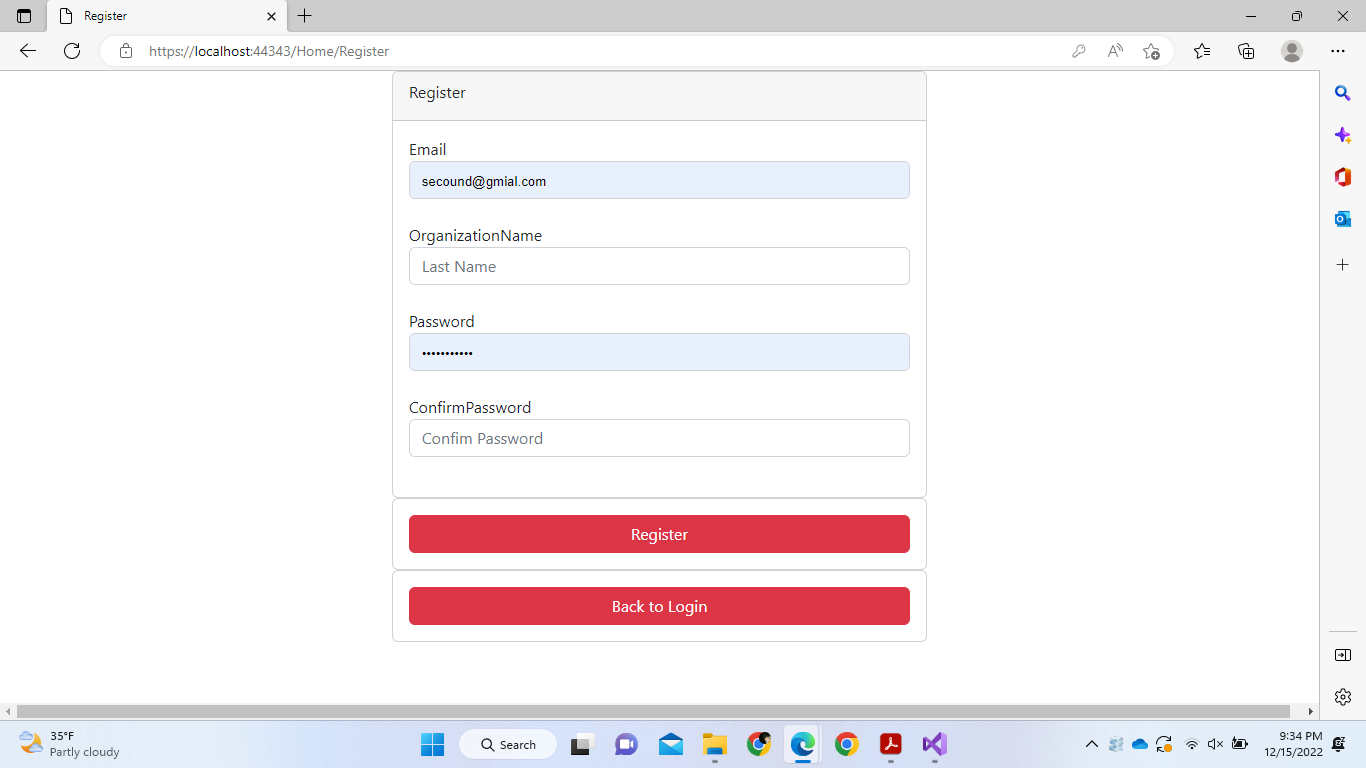
We have used the .NET framework as it provides the Entity framework to handle the database without buying any server or storage location for our project.

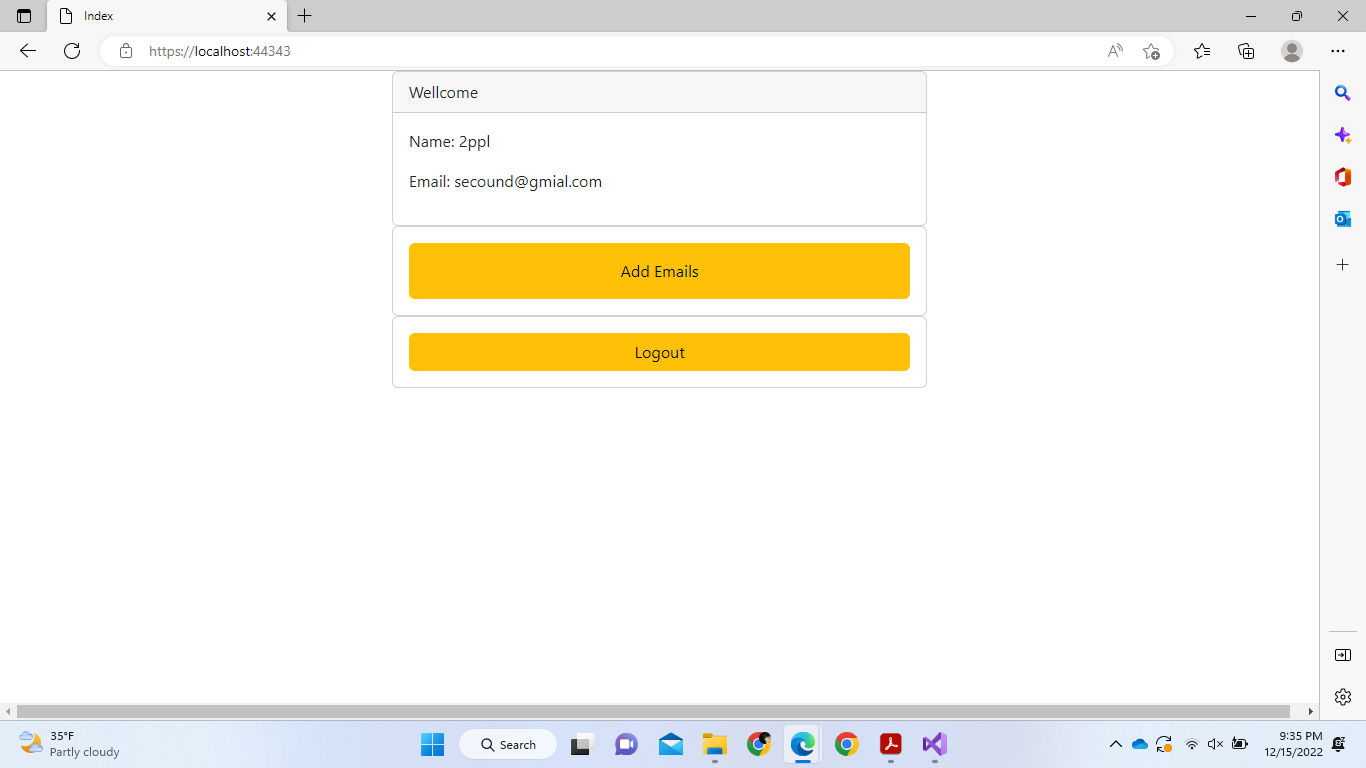
So,

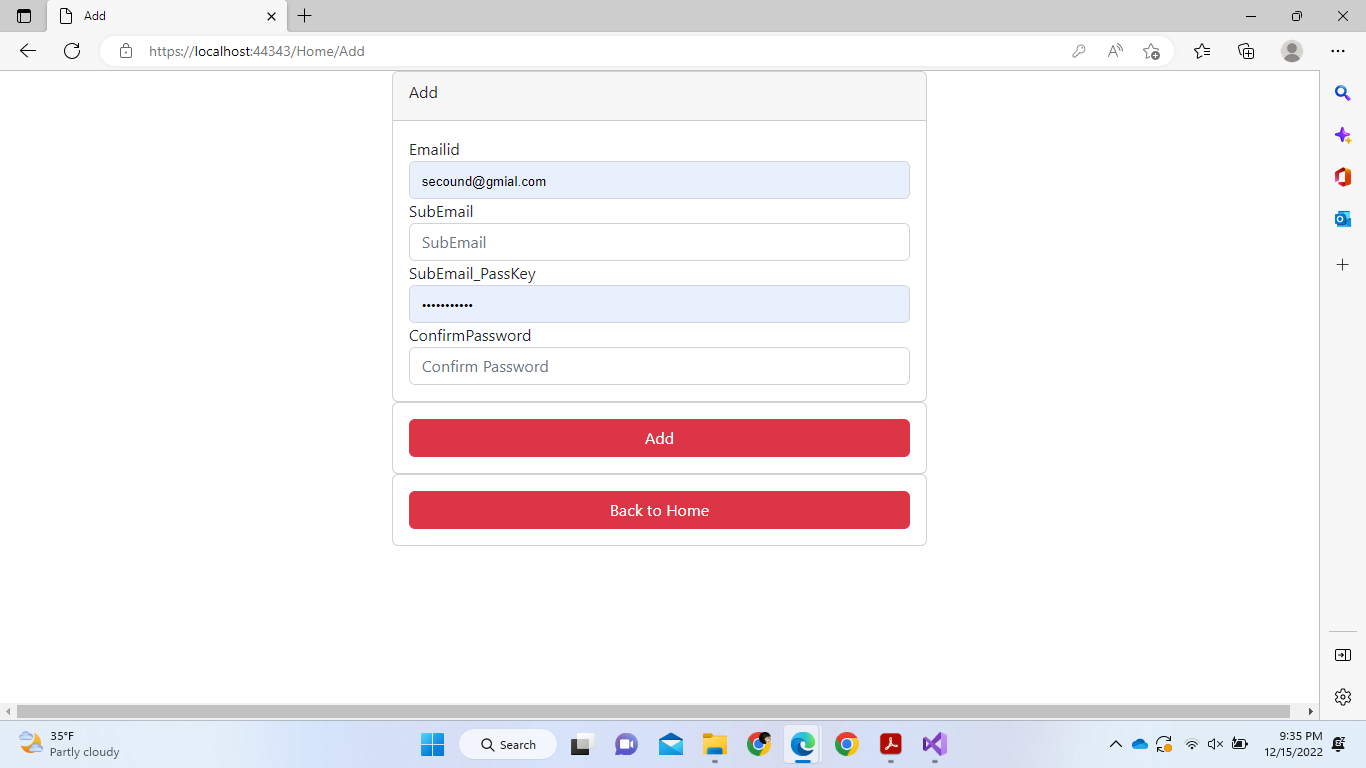
* To create controllers we use C#
* To handle models, we use Entity Framework
* To create views we use Bootstrap, Html, CSS, JavaScript

PROJECT OUTPUT SCREENSHOTS:





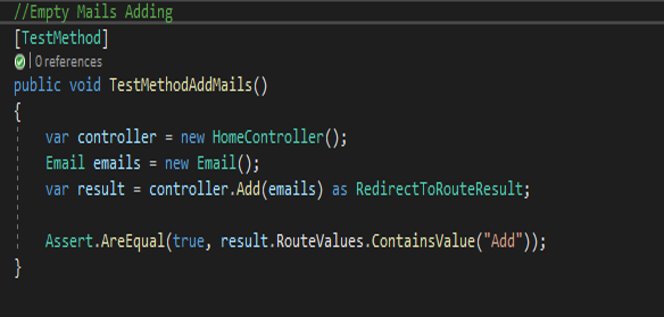


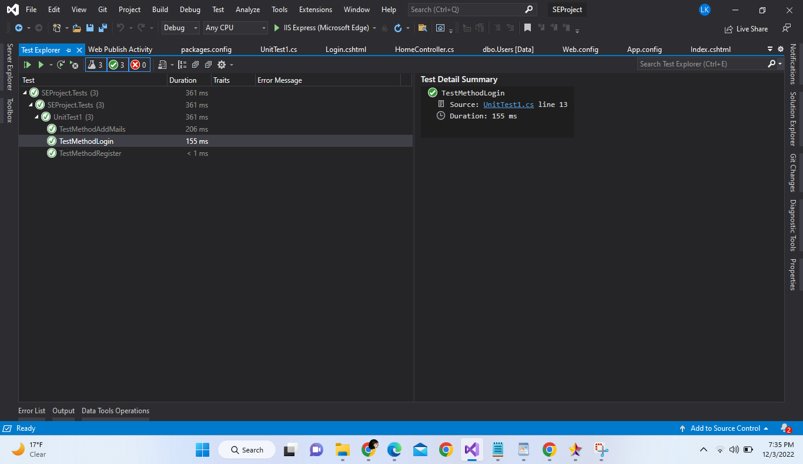


PROJECT TEST CASES:









**6. PROJECT CONCLUSION AND ENHANCEMENTS:**

As discussed earlier, they are time-consuming and physically boring tasks in our daily work handling software's. To perform those tasks if we can find solutions to it. Then we can add those services to the application.

We can provide access to those services on some conditions.

**GitHub link:** https://github.com/laxma-dev/SEProjectFinal