



Dharmsinh Desai University, Nadiad

Faculty of Technology

Department of Computer Engineering

B. Tech. CE Semester – VI

Subject: Service Oriented Computing (CE-619)

Project Title: PG Finder

By:

1] Dhruval Shah, Roll No: CE-140, Id: 19CEUON071

2] Jainil Trivedi, Roll No: CE-16, Id: 19CEUON061

Guided By:

1] Prof. Prashant M Jadav

2] Prof Ankit P Vaishnav



Certificate

This is to certify that the project entitled as “PG Finder” is a Bonafede report of the work carried out by:

1] Dhruval Shah, Roll No: CE-140, Id: 19CEUON071

2] Jainil Trivedi, Roll No: CE-166, Id: 19CEUON061

of Department of Computer Engineering, semester VI, under the guidance and supervision of Prof. Prashant M Jadhav and Prof. Ankit P. Vaishnav for the subject Advanced Technologies during the academic year 2021-2022.

Project Guide:

Prof. Ankit P Vaishnav

Prof Prashant M Jadav

Head of the department:

Dr. C.K Bhensdadia

Department of Computer Engineering,

Faculty Of Technology,

Dharmsinh Desai University,

Nadiad

Contents:

1.Front Page.....	01
2. Certificate	02
3. Contents	03
4.Introduction	04
5.Software Requirement specification	05
6.Design	
I) Use Case Diagram	07
II) Class Diagram.....	08
III)Data Dictionary	09
7.Implementation Details	
I) Modules (Brief Description)	10
II)Major functionality	11
8.Testing	12
9.Screenshots	14
10.Conclusion	18
11.Limitations and Future Extensions	18
12.Bibliography	18

Introduction

Our project PG finder will show user available PGs in Nadiad, Vadodara and Ahmedabad. User can contact the owner if he finds a PG which matches his requirements.

Technology Used

1. .NET Framework
2. Windows Communication Foundation (WCF)
3. SQL Server (For database)
4. Bootstrap (for UI of tables)
5. Font Awesome (for Icons)

Tools

1. Visual Studio (IDE)
2. Git and GitHub (version control)

Software Requirements Specifications

End Users:

1. User
2. Admin

Functional Requirements:

R1] Authentication of User

R1.1] Register User/ Sign Up

Desc: A user who doesn't have existing account can sign up.

Input: User name for the web app, email id for contact and password to log in to our website.

Output: User will be redirected to login page upon successful sign up. We may show relevant errors.

R1.2] Log in

Desc: User can log in to our web app through this.

Input: User name and password

Output: Logged in and redirected to main page

R1.3] Update User

Desc: User can change password and other details.

Input: New Detail(s)

Output: Redirected to homepage.

R1.4] Log Out

Desc: Logout of the web app

Input: Click on logout button

Output: Redirected to login page and session abandoned

R2] Admin Functions

R1.5.1] Login

Desc: Login to admin dashboard

Input: username and password

Output: redirected to admin

R1.5.2] Delete User

Desc: Admin has authority to delete user

Input: Click on delete button

Output: Deleted user, shown new data in table

R1.5.2] Delete PG

Desc: Admin has authority to delete PG

Input: Click on delete button

Output: Deleted PG, shown new data in table

R1.5.2] Update PG

Desc: Admin has authority to Update PG

Input: Click on Update button and change details

Output: Updated PG details, shown new data in table

R1.5.2] Add PG

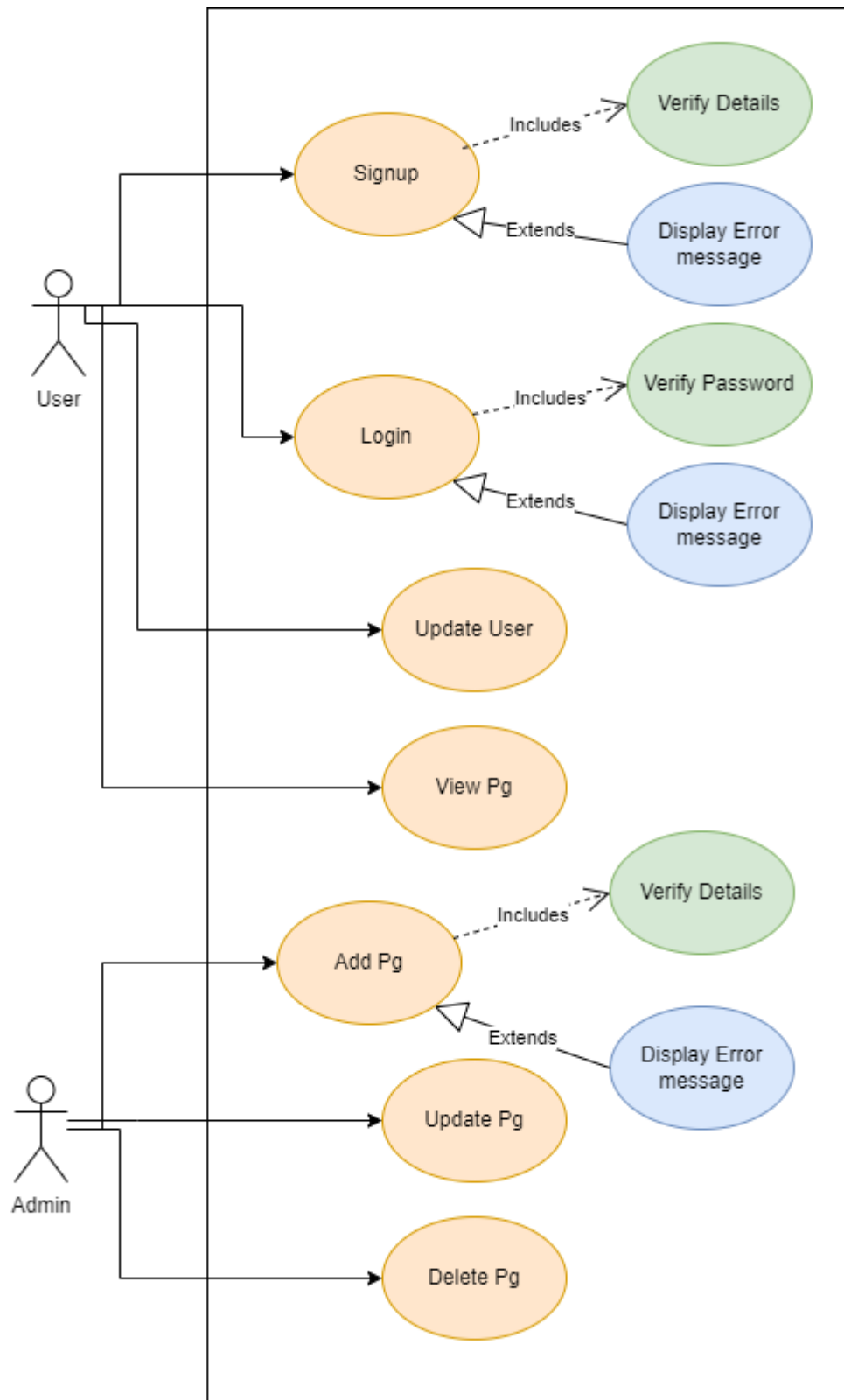
Desc: Admin has authority to Add PG

Input: Click on Add button and give details

Output: Added PG details, shown new data in table

Design

Use Case Diagram



Class Diagram



Data Dictionary

User:

Sr.No	Field	Datatype	Constraint
1	Id	int	Not null, Primary key
2	userName	Varchar	Not null, unique
3	Email	Varchar	Not null, unique
4	Password	Varchar	Not null
5	IsAdmin	bit	Not null, default = ((0))

Home:

Sr.No	Field	Datatype	Constraint
1	Id	int	Not null, Primary key
2	Location	Varchar	Not null
3	Area	int	Not null
4	Age	int	Not null
5	Bhk	int	Not null
6	Rent	int	Not null

Implementation Details

Modules Implemented

1. User

⇒ In User module, log in, log out, update user, sign up has been implemented by us.

2. PG

⇒ All of PG modules functions are only given to super user i.e., admin. He can delete, update and add new PGs on the site which will be seen by the users. To do this he has to log in with the admin's credentials.

Function prototypes which implement major functionalities

Major Functions of the user:

```
using System.ServiceModel;

namespace HomeAndUserLibrary
{
    [ServiceContract]
    2 references
    public interface IUserService
    {
        [OperationContract]
        1 reference
        DataSet GetUsers();

        [OperationContract]
        1 reference
        User GetUser(int id);

        [OperationContract]
        1 reference
        int AddUser(User user);

        [OperationContract]
        1 reference
        int DeleteUser(int id);

        [OperationContract]
        1 reference
        User LogInUser(string userName, string pass);

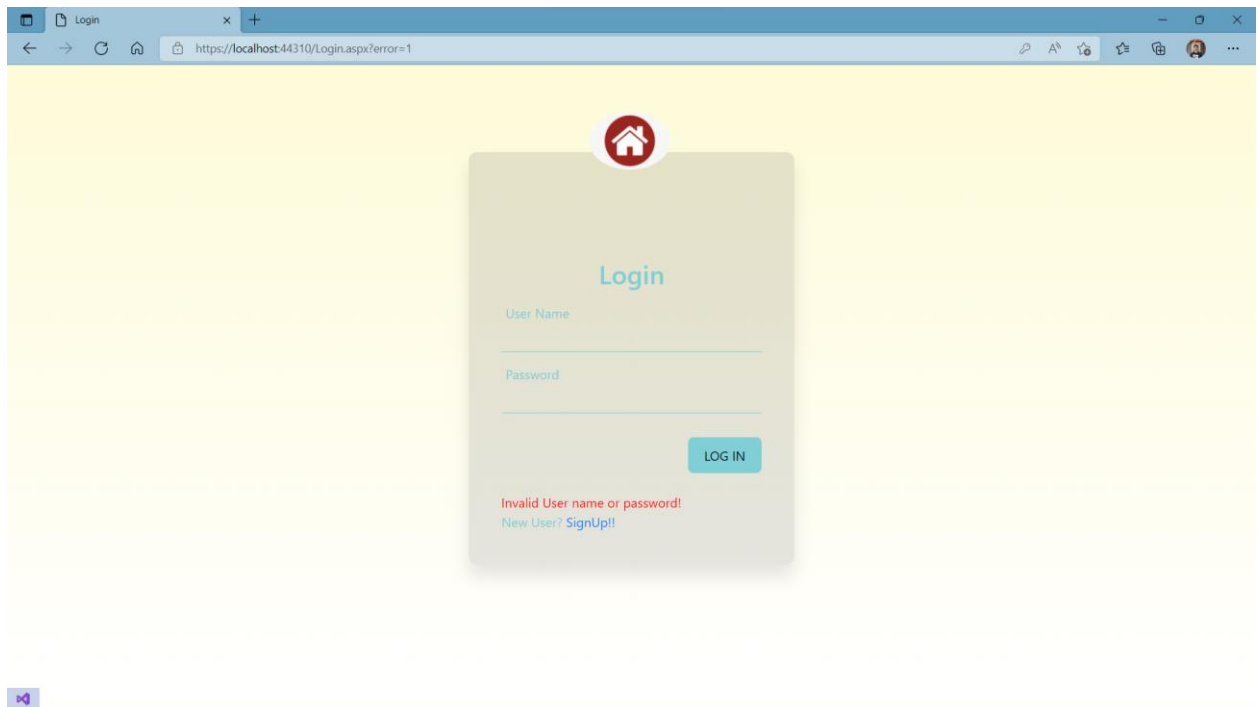
        [OperationContract]
        1 reference
        int UpdateUser(int id, User user);
    }
}
```

Major Functions of the PG accessible by the admin:

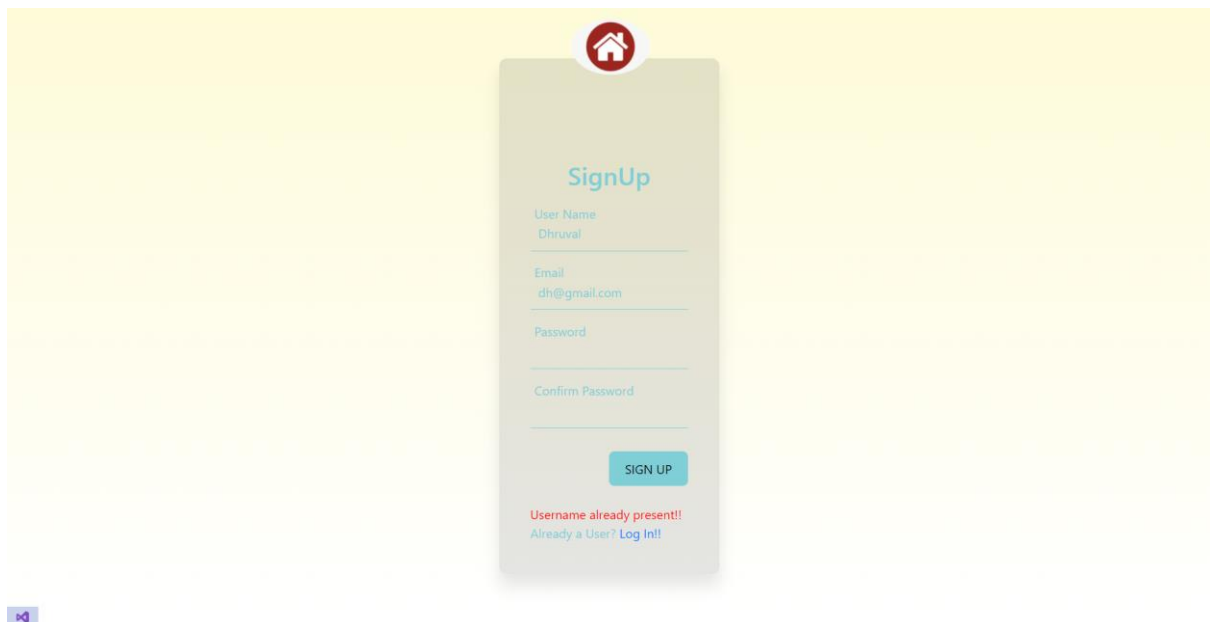
```
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data;
7 using System.ServiceModel;
8
9 namespace HomeAndUserLibrary
10 {
11     [ServiceContract]
12     public interface IHomeService2
13     {
14         [OperationContract]
15         DataSet GetHouses();
16
17         [OperationContract]
18         DataSet getHome(string feature, int limit);
19
20         [OperationContract]
21         Home getHomeWithID(int id);
22
23         [OperationContract]
24         int RemoveHome(int id);
25
26         [OperationContract]
27         int AddHome(Home home);
28
29         [OperationContract]
30         int UpdateHome(int id, Home home);
31     }
32 }
```

Testing

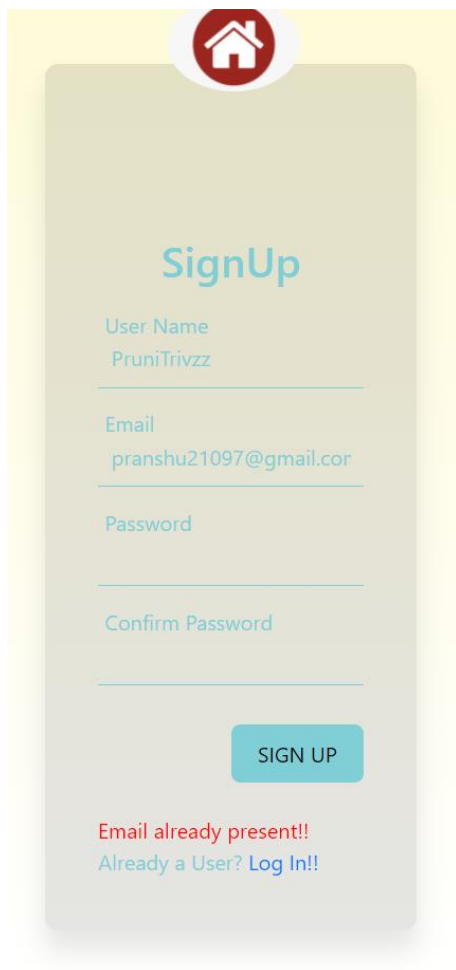
Login using wrong credentials.



Username already exists:



Email Id already exists:



SignUp

User Name
PruniTrivzz

Email
pranshu21097@gmail.cor

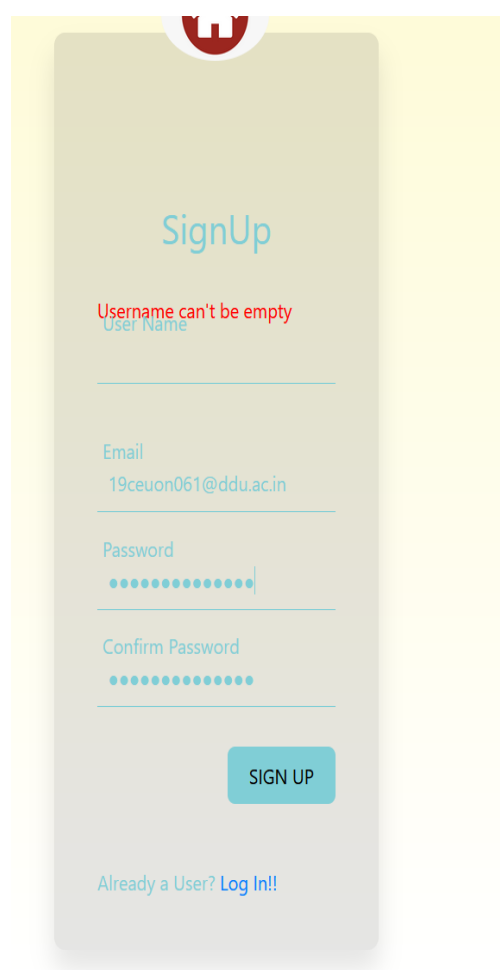
Password

Confirm Password

SIGN UP

Email already present!!
Already a User? [Log In!!](#)

Empty Username:



SignUp

Username can't be empty
User Name

Email
19ceun061@ddu.ac.in

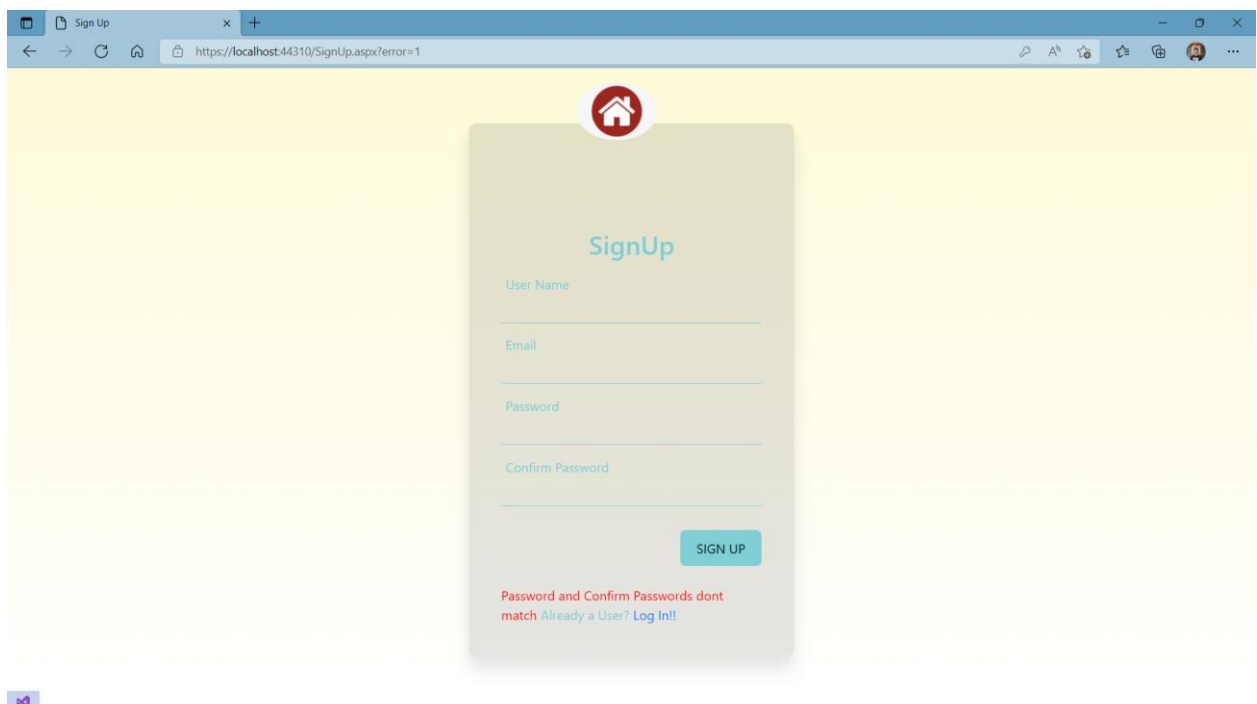
Password
.....

Confirm Password
.....

SIGN UP

Already a User? [Log In!!](#)

Password and confirm password don't match on sign up page



Sign Up

https://localhost:44310/SignUp.aspx?error=1

SignUp

User Name

Email

Password

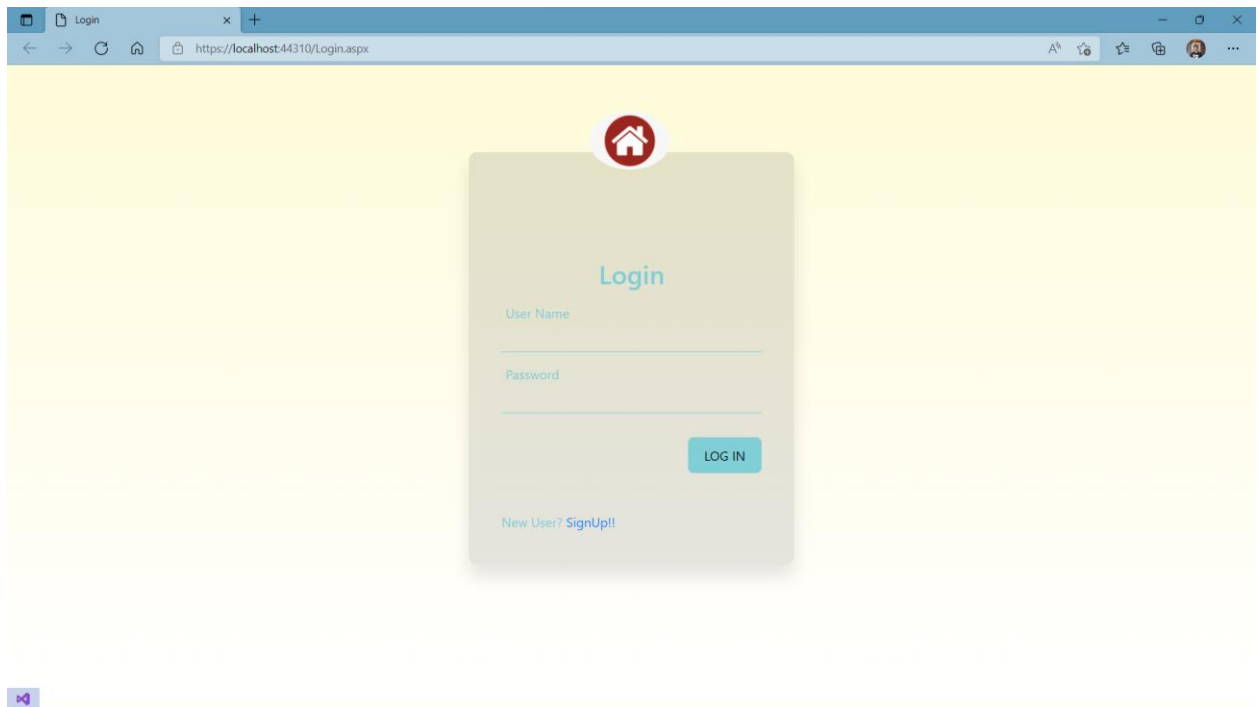
Confirm Password

SIGN UP

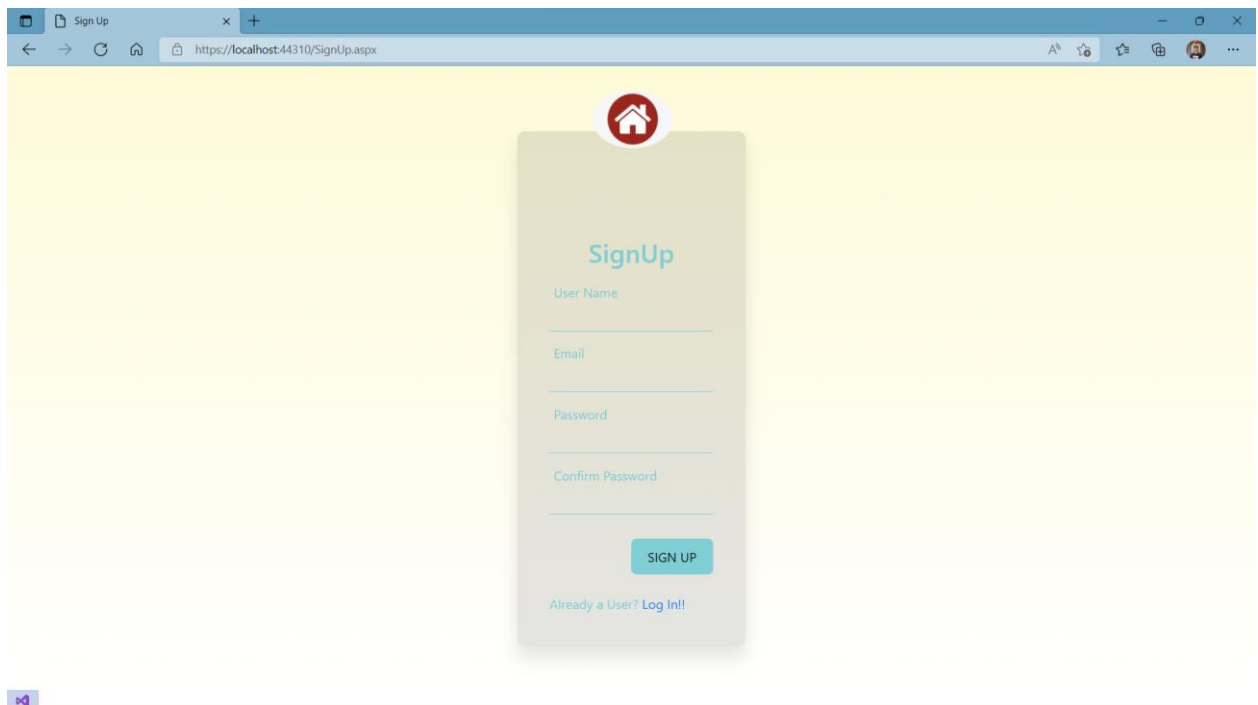
Password and Confirm Passwords dont
match Already a User? [Log In!!](#)

Screen Shots of Project

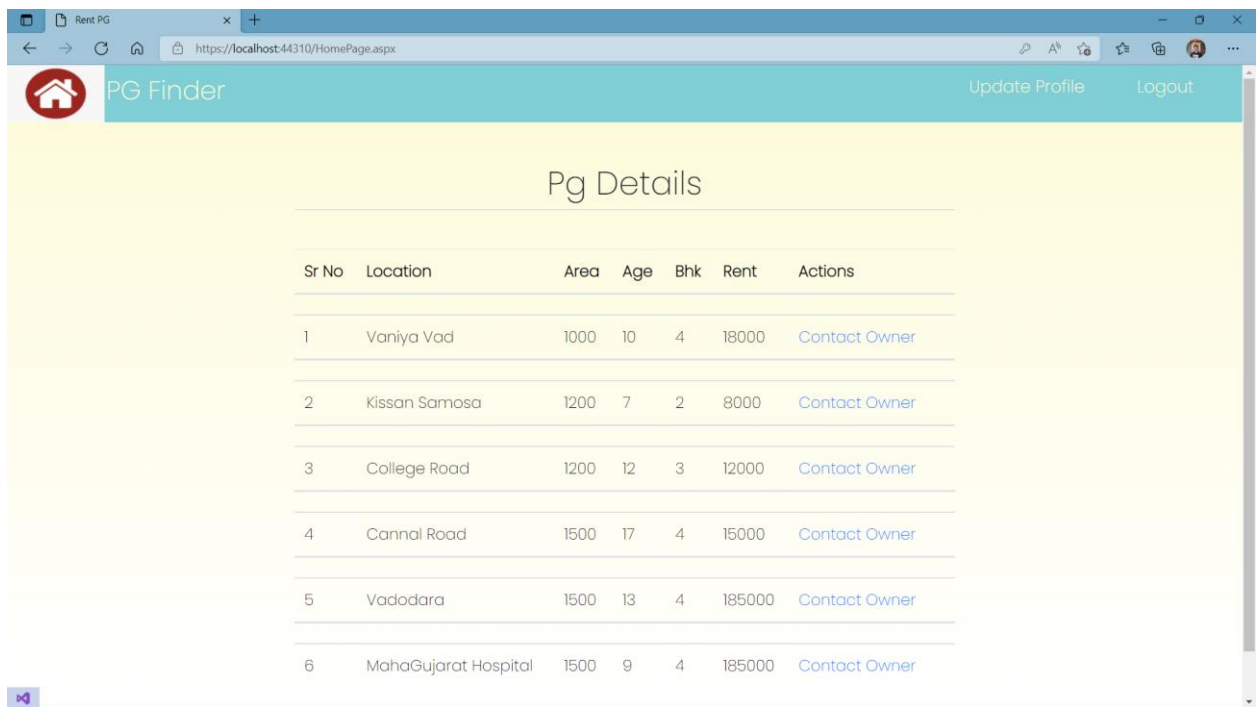
Log in Page



Sign Up Page



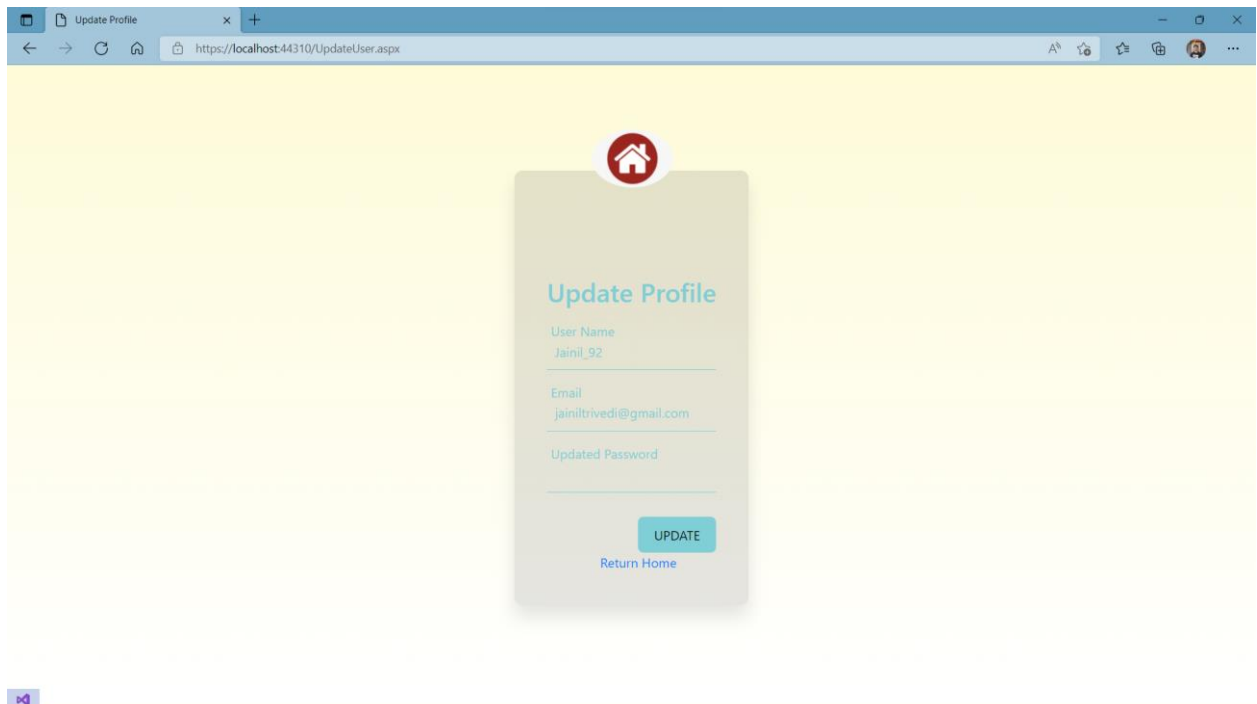
Logged In, User Home page



The screenshot shows a web browser window with the title "Rent PG" and the URL "https://localhost:44310/HomePage.aspx". The page features a teal header with a home icon, the text "PG Finder", and links for "Update Profile" and "Logout". The main content area is yellow and titled "Pg Details". It contains a table with 6 rows of PG listings. Each row includes a serial number, location, area, age, bhk, rent, and a "Contact Owner" link.

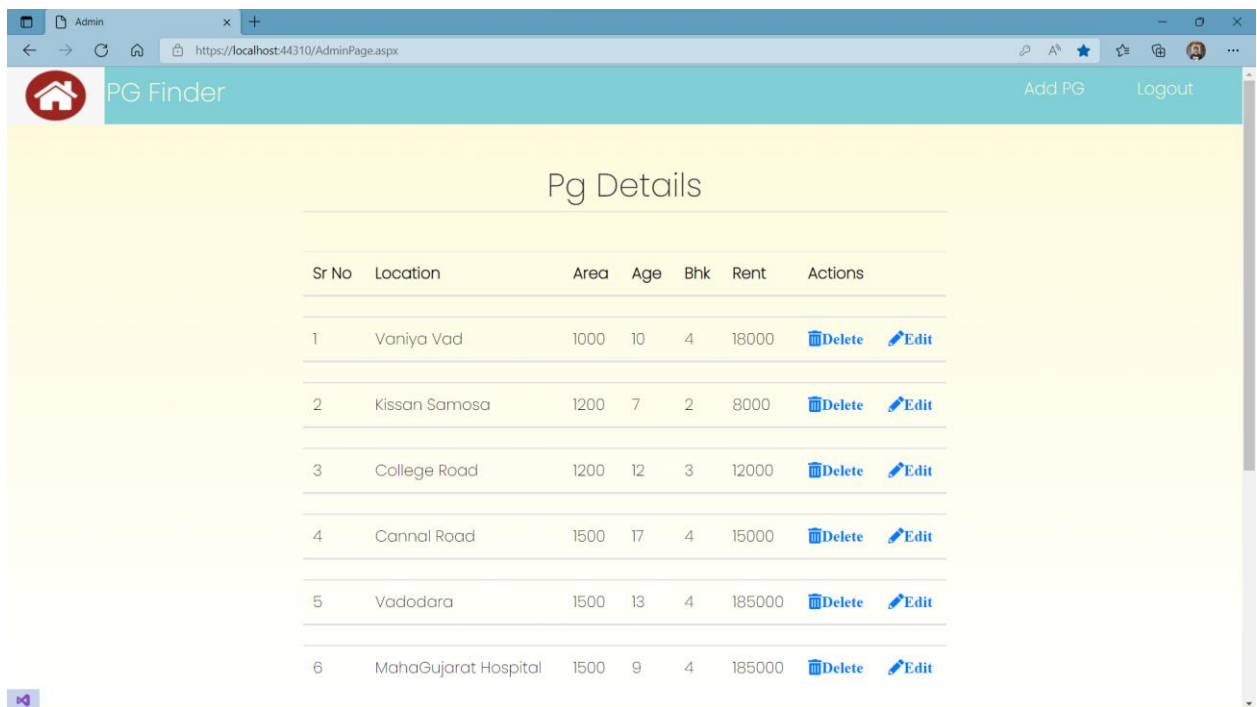
Sr No	Location	Area	Age	Bhk	Rent	Actions
1	Vaniya Vad	1000	10	4	18000	Contact Owner
2	Kissan Samosa	1200	7	2	8000	Contact Owner
3	College Road	1200	12	3	12000	Contact Owner
4	Cannal Road	1500	17	4	15000	Contact Owner
5	Vadodara	1500	13	4	185000	Contact Owner
6	MahaGujarat Hospital	1500	9	4	185000	Contact Owner

Update Profile



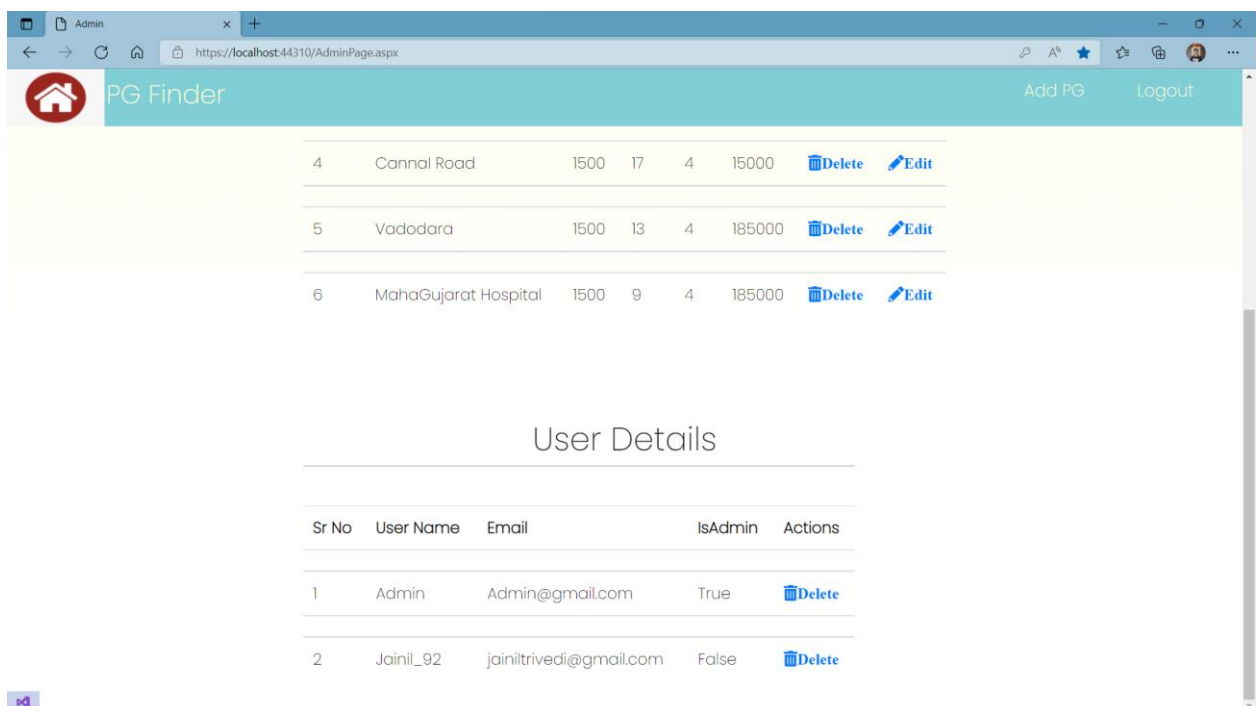
The screenshot shows a web browser window with the title "Update Profile" and the URL "https://localhost:44310/UpdateUser.aspx". The page features a yellow background with a central grey card. The card has a home icon at the top, followed by the title "Update Profile". Below the title are three input fields: "User Name" (containing "Jainil_92"), "Email" (containing "jainiltrivedi@gmail.com"), and "Updated Password". At the bottom of the card is a teal "UPDATE" button and a blue "Return Home" link.

Admin Dash board



The screenshot shows the Admin Dashboard for PG Finder. The header includes a home icon, the text "PG Finder", and links for "Add PG" and "Logout". The main content area is titled "Pg Details" and contains a table with the following data:

Sr No	Location	Area	Age	Bhk	Rent	Actions
1	Vaniya Vad	1000	10	4	18000	Delete Edit
2	Kissan Samosa	1200	7	2	8000	Delete Edit
3	College Road	1200	12	3	12000	Delete Edit
4	Cannal Road	1500	17	4	15000	Delete Edit
5	Vadodara	1500	13	4	185000	Delete Edit
6	MahaGujarat Hospital	1500	9	4	185000	Delete Edit



The screenshot shows the Admin Dashboard for PG Finder, displaying the "User Details" section. The header is the same as the previous screenshot. The main content area is titled "User Details" and contains a table with the following data:

Sr No	User Name	Email	IsAdmin	Actions
1	Admin	Admin@gmail.com	True	Delete
2	Jainil_92	jainiltrivedi@gmail.com	False	Delete

Update PG

A screenshot of a web browser displaying the 'Update PG' form. The browser's address bar shows the URL 'https://localhost:44310/UpdatePG.aspx?id=8'. The form is titled 'Update PG details' and features a red home icon at the top. It contains several input fields with pre-filled values: 'Location' (Cannal Road), 'Area' (1500), 'Age' (17), 'Rent' (15000), and 'BHK' (4). At the bottom right of the form is a teal 'UPDATE' button, and at the bottom left is a 'Return Home' link. The background of the page is a light yellow gradient.

Add new PG

A screenshot of a web browser displaying the 'Add new PG' form. The browser's address bar shows the URL 'https://localhost:44310/AddPG.aspx'. The form is titled 'New PG details' and features a red home icon at the top. It contains several empty input fields for 'Location', 'Area', 'Age', 'Bhk', and 'Rent'. At the bottom right of the form is a teal 'ADD' button, and at the bottom left is a 'Return Home' link. The background of the page is a light yellow gradient.

Conclusion

We have successfully implemented PG finder.

All the functionalities listed in our SRS have been divided into two modules user and PG/Home/Admin functions. We have also made some validations check to prevent malicious activities. We have also tested our web app as a whole and it passes those tests.

We have implemented Create user through sign in, Read user through log in and showing details on admin dashboard, Update user through update profile and Delete user by allowing admin to delete a user if he wants or finds them malicious.

We have also implemented Create PG, Delete PG and Update Pg which can be done only by admin through admin dashboard. We have also implemented Read PG by showing details of PG on both user's home page and admin dashboard.

Limitations and Future Extension

1. User can't contact owner, we have not made any arrangements for the same.

In future, we would like to make these arrangements

2. For now, only admin can add PG, while we should allow others also to add PG to expand our reach. Also add PG from more regions.

We would like to add these features in future.

3. Add other features to identify users, so that they can't cheat. Like email verification, biometrics etc.

In future, we would like to add this feature.

4. We can have/save more details of the PG like photos, exact address, owner contact (for now we have email) etc. to easy users in finding their choice of PG.

This must be implemented in near future.

Bibliography

- <https://docs.microsoft.com/en-us/dotnet/framework/wcf/>
- Lecture materials provided.