AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

Dept. of Computer Science
Faculty of Science and Technology

CSC2210: OBJECT ORIENTED PROGRAMMING 2

Fall 2024-2025

Section: [J]

Group No: 09

Project Report On

House Rental Management System

Supervised By

Taslimur Rahman

Submitted By:

SL No:	Name	ID	Contribution
1	Md Rubai Hasan Abir	22-48963-3	Design and Database
2	Salim Sadman Evan	22-48973-3	Er Diagram Design
3	Afrin Akter Ridika	22-48986-3	Project Idea,Design and Database
4	Zobayer Al Mahmud	22-47989-2	Report and Database

CO2: Display and verify the mean of a real-life Project using the concepts of C# Graphical User Interface based environment with database integration to depict a desktop-based application.					
Assessment Criteria	**Not Attended/ Incorrect (0)**	**Inadequate (1-2)**	**Average (3)**	**Good (4)**	**Excellent (5)**
Evaluation Criteria	**Evaluation Definition**				
Requirement fulfillment	Fails to demonstrate any understanding of real-life scenario-based project development or functional requirement identification. There is no attempt to depict a project or identify functional requirements accurately.	Demonstrates limited understanding of real-life scenario-based project development and functional requirement identification. The project depicted lacks coherence or relevance to real-life scenarios, and functional requirements are inaccurately identified or insufficiently described.	Presents a basic depiction of a real-life scenario-based project and identifies some functional requirements. However, the project lacks depth or complexity, and some functional requirements may be vaguely defined or missing key details.	Effectively demonstrates a realistic scenario-based project and accurately identifies most functional requirements. The project is well-developed with appropriate complexity, and functional requirements are clearly articulated with relevant details.	Exhibits an exceptional understanding of real-life scenario-based project development and accurately identifies all functional requirements. The project is meticulously developed with thorough attention to detail, reflecting a comprehensive understanding of Object-Oriented Programming project development activities.
Validation	Fails to demonstrate any understanding or implementation of validation forms in their system. There is no attempt to deal with data validation, and validation requirements are completely ignored or incorrectly applied.	Demonstrates limited understanding of validation forms and data validation techniques. While some attempt may be made to implement validation, it is incomplete or poorly executed, leading to inadequate handling of data validation.	Shows a basic understanding of validation forms and data validation techniques. They attempt to implement validation, but some aspects may be missing or incorrectly implemented, resulting in partial or inconsistent handling of data validation.	Effectively demonstrates the use of validation forms and implements data validation techniques. Validation is mostly accurate and comprehensive, ensuring the proper handling of data input and verification in the system.	Exhibits an exceptional understanding and implementation of validation forms and data validation techniques. Validation is meticulously implemented with thorough attention to detail, ensuring robust data validation procedures and contributing to the overall reliability and integrity of the system.

Verification	Fails to demonstrate any attempt to verify the system data or functional requirements. There is no evidence of understanding or implementation of verification processes, and data flow is not considered.	Demonstrates limited understanding of verification processes and data flow in the system. Verification attempts are incomplete or inaccurate, and there is insufficient consideration given to ensuring data integrity and functionality.	Shows a basic understanding of verification processes and attempts to verify system data. However, verification efforts may be inconsistent or lack thoroughness, and there may be gaps in ensuring proper functional requirements and data flow.	Identifies and verifies system data, ensuring proper functional requirements are met. Verification efforts are mostly accurate and thorough, with attention to ensuring data integrity and appropriate data flow within the system.	Exhibits an exceptional understanding of verification processes and meticulously verifies system data. Verification efforts are comprehensive and precise, with a keen focus on ensuring all functional requirements are met and maintaining proper data flow throughout the system.

CO3: Prepare and Explain a real life desktop based application synthesizing several component of C# along with development tools to adhere the given requirements.

Assessment Criteria	Not Attended/ Incorrect (0)	Inadequate (1-2)	Average (3)	Good (4)	Excellent (5)
Evaluation Criteria	**Evaluation Definition**				
Organization of the application	Fails to identify any suitable real time application or requirements for project development activities related to OOP.	Limited understanding about the project scopes and scenarios or identification of functional requirements.	Lacks depth or relevance to OOP project development activities and may contain inaccuracies. Real-life scenarios are mentioned, but the discussion lacks depth or clarity.	Consider and integrate the idea of several core aspects of the project along with relevance to real-life scenarios. Demonstrating a solid understanding of the application presentation.	Generalize and exhibits an exceptional understanding of project preparation according to a to real-life scenarios. Also contains proper and insightful identification of the system which is comprehensive and precise.
Representation and Integration of Database	Fails to identify and present any understanding or implementation of database. Also failed to integrate the data with the project itself.	Limited understanding of the database concepts or their proper way of using in a real time project. While some attempt may be made to implement but it is incomplete or poorly executed, leading to inadequate design.	Lacks depth or relevance to database integration with the application. Shows a basic understanding but some aspects may be missing or incorrectly implemented, resulting in partial or inconsistency. May lack proper normalization.	Integrate the database with the forms properly and implements it with proper validation which is mostly accurate and comprehensive, ensuring the proper handling of data input and verification along with general normalization.	Exhibits an exceptional understanding and implementation of database ensuring attention to detail, and robust data manipulation procedures and contributing to the overall clarity.

Graphical User Interface	Fails to present or prepare GUI based application interfaces. There is no evidence of creating or integrating such things according to their usefulness.	Limited understanding of graphical user interfaces. Lack of design knowledge. Very poor attempt to make such things which are currently obsolete or can't be identified as coherent.	Shows a basic understanding of creating user interfaces. Most of them are interconnected but maybe some of them lack it. However, most of it can be described as user friendly.	Effectively identifies and meet the consider the simplicity. Design related works are mostly accurate and taken proper attention to ensuring a user-friendly coherent system.	Exhibits an exceptional work design following a high standard of simple and elegant work. Several controls and mechanism has been organized in a preferred way according to the coherent usage .

Table of Contents:

Contents	Page no.
1. **Chapter:** 01 Introduction	6
2. **Chapter:** 02 User Story	6
3. **Chapter:** 03 ER Diagram	7
4. **Chapter:** 04 SQL Queries	8-9
5. **Chapter:** 05 Screenshots	10-18

CHAPTER 01: INTRODUCTION

A House Rental Management System is a digital solution designed to facilitate and automate the process of renting, managing, and maintaining residential properties. It serves as a platform for property landlords and tenants efficiently handle rental listings, lease agreements and tenant applications. With the increasing demand for rental properties, managing multiple tenants, properties, and transactions manually becomes challenging. A well-structured House Rental Management System simplifies property management, improves efficiency, and enhances the overall rental experience for both landlords and tenants.

CHAPTER 02: User Stories

The system has two main types of users: **Admins** and **Regular Users** (Tenants and Landlords). Here's how each role interacts with the system:

1. Admin User Story
Admins are responsible for managing the system. They have full control over the property listings, user accounts, and rental agreements. As an admin, I should be able to:

- Log in securely to the admin dashboard.
- Remove property listings.
- Monitor rental agreements and resolve issues.
-

2. Regular User Story (Tenants and Landlords)
Regular users include Tenants and Landlords. They can interact with the system to manage rentals and properties. As a regular user, I should be able to:

- Log in to my account securely.
- **Tenants**: Search for available properties, view details (like rent, location) and request to rent a property.
- **Landlords**: Add new properties with proper property details.

Example Scenarios
Admin Managing Properties:

- The admin logs in and accesses the property management section.
- The admin manages the booking requests.

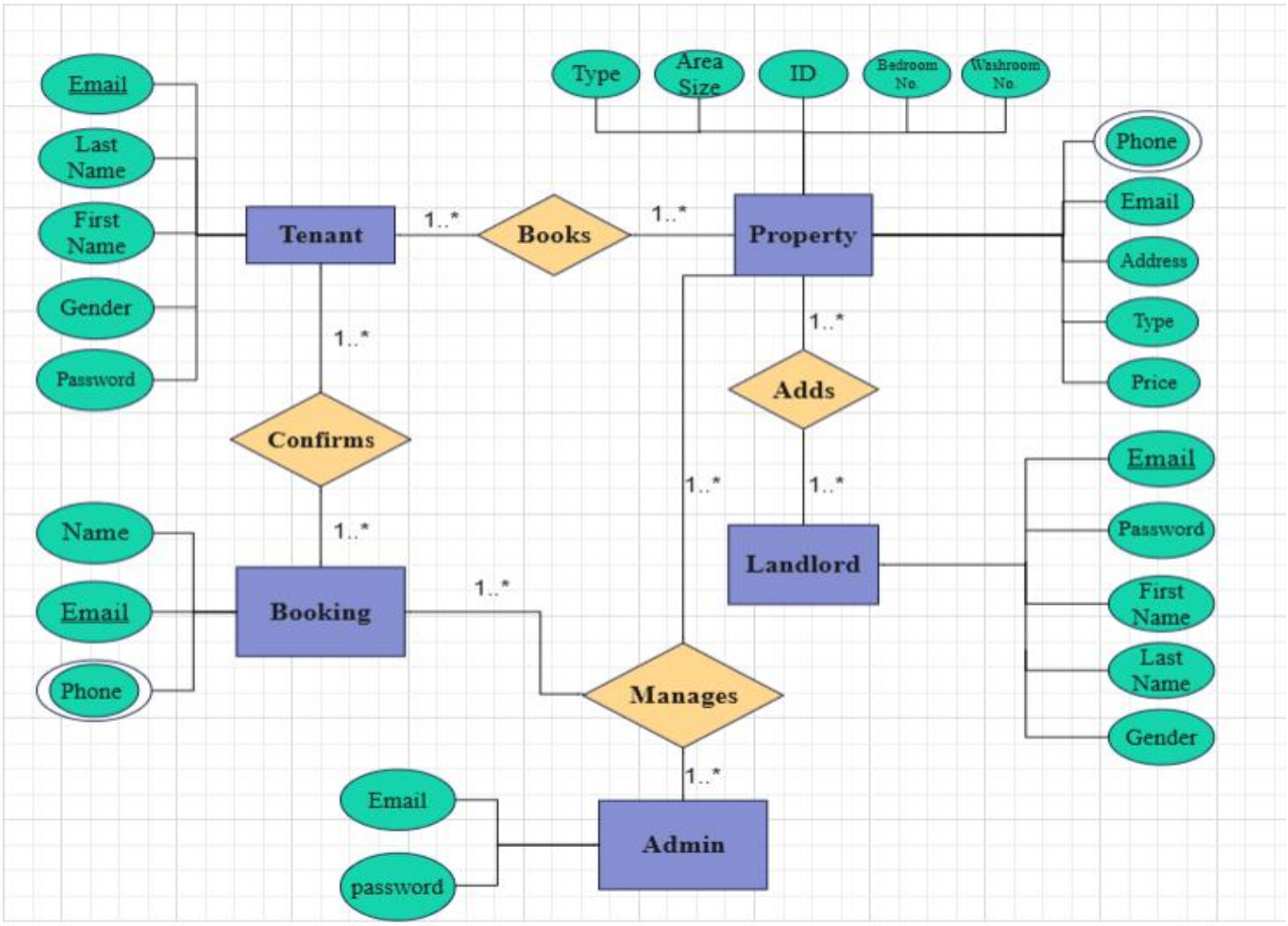
Tenant Renting a Property:

- A tenant logs in and searches for available properties.
- The tenant selects a property, confirms the details, and submits a rental request.
- The system updates the property status and logs the rental request in the database.

Landlord Adding a Property:

- A landlord logs in and adds a new property with details like rent, location, and features.
- The system saves the property information

CHAPTER 03: ER Diagram



CHAPTER 04: SQL Queries

DATABASE

```
CREATE DATABASE erent_db;
 USE erent_db;
```

TENANT

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  first_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  gender ENUM('Male', 'Female', 'Others') NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  password_hash VARCHAR(255) NOT NULL);
```

LANDLORD

```
CREATE TABLE landlord (
  id INT AUTO_INCREMENT PRIMARY KEY,
  first_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  gender ENUM('Male', 'Female', 'Others') NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  password_hash VARCHAR(255) NOT NULL);
```

ADMIN

```
INSERT INTO admin (email, password)
   VALUES ('rubai@erent.com', 'Admin123#');
```

BOOKING INFO

```
CREATE TABLE INFO (
  id INT AUTO_INCREMENT PRIMARY KEY,  -- Unique identifier for each record
  Name VARCHAR(100) NOT NULL,         -- Name field
  Email VARCHAR(100) UNIQUE NOT NULL, -- Email field (unique)
  ContactPhoneNumber VARCHAR(15)      -- Contact phone number (varchar to handle different
formats));
```

PROPERTY ADD INFO

```
CREATE TABLE LINFO (
    id INT AUTO_INCREMENT PRIMARY KEY, Unique identifier for each record
    Name VARCHAR(100) NOT NULL, Name field
   Address VARCHAR(100) NOT NULL, Address field
    Road VARCHAR(100) NOT NULL, Road field
    Bedroom VARCHAR(100) NOT NULL, Bedroom field
    Washroom VARCHAR(100) NOT NULL, Washroom field
   Email VARCHAR(100) UNIQUE NOT NULL, Email field (unique)
    AreaSize VARCHAR(100) NOT NULL, AreaSize field
    Price VARCHAR(100) NOT NULL, Price field
    ContactPhone Number VARCHAR(15)
   Contact phone number (varchar to handle different formats));
```

CHAPTER 05: Screenshots

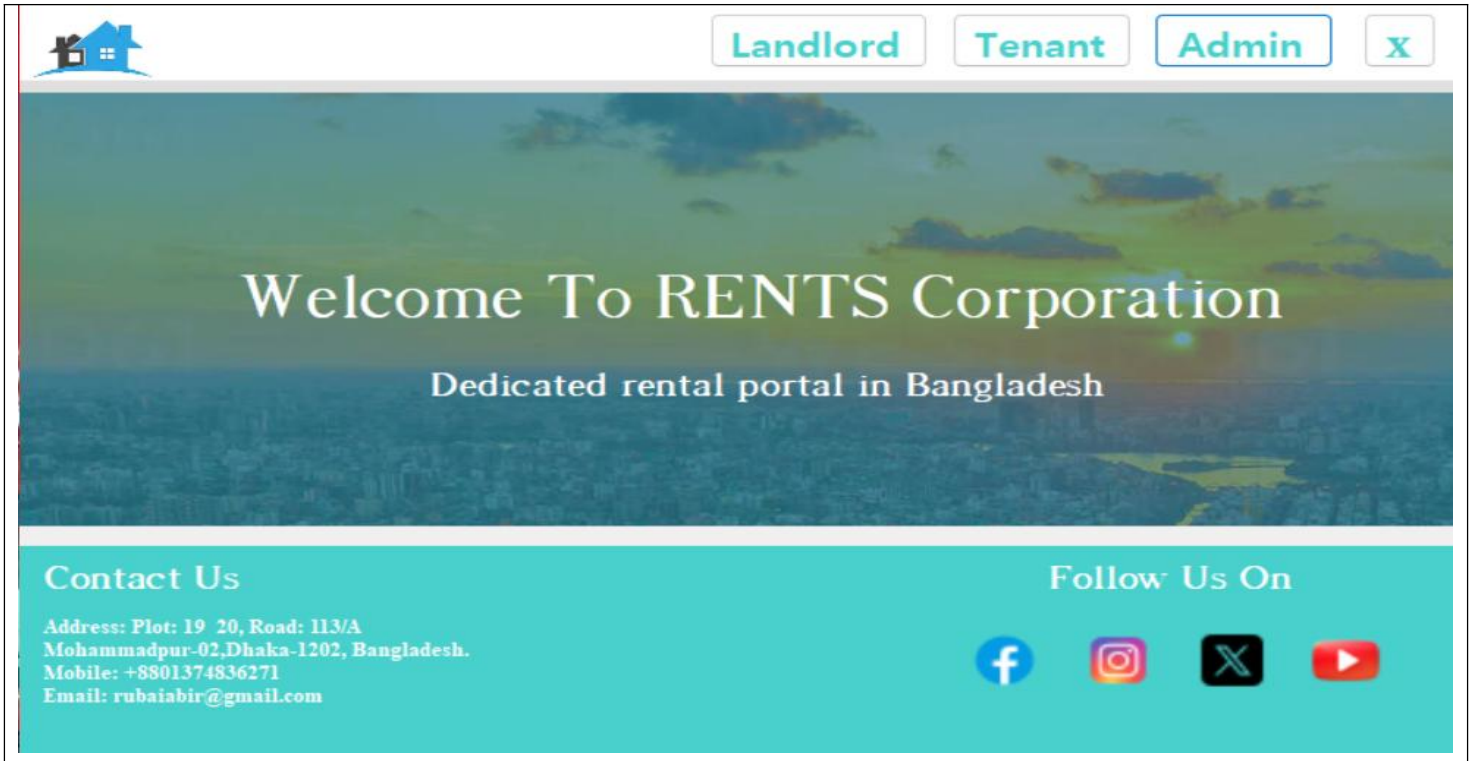


Fig 1: Home page

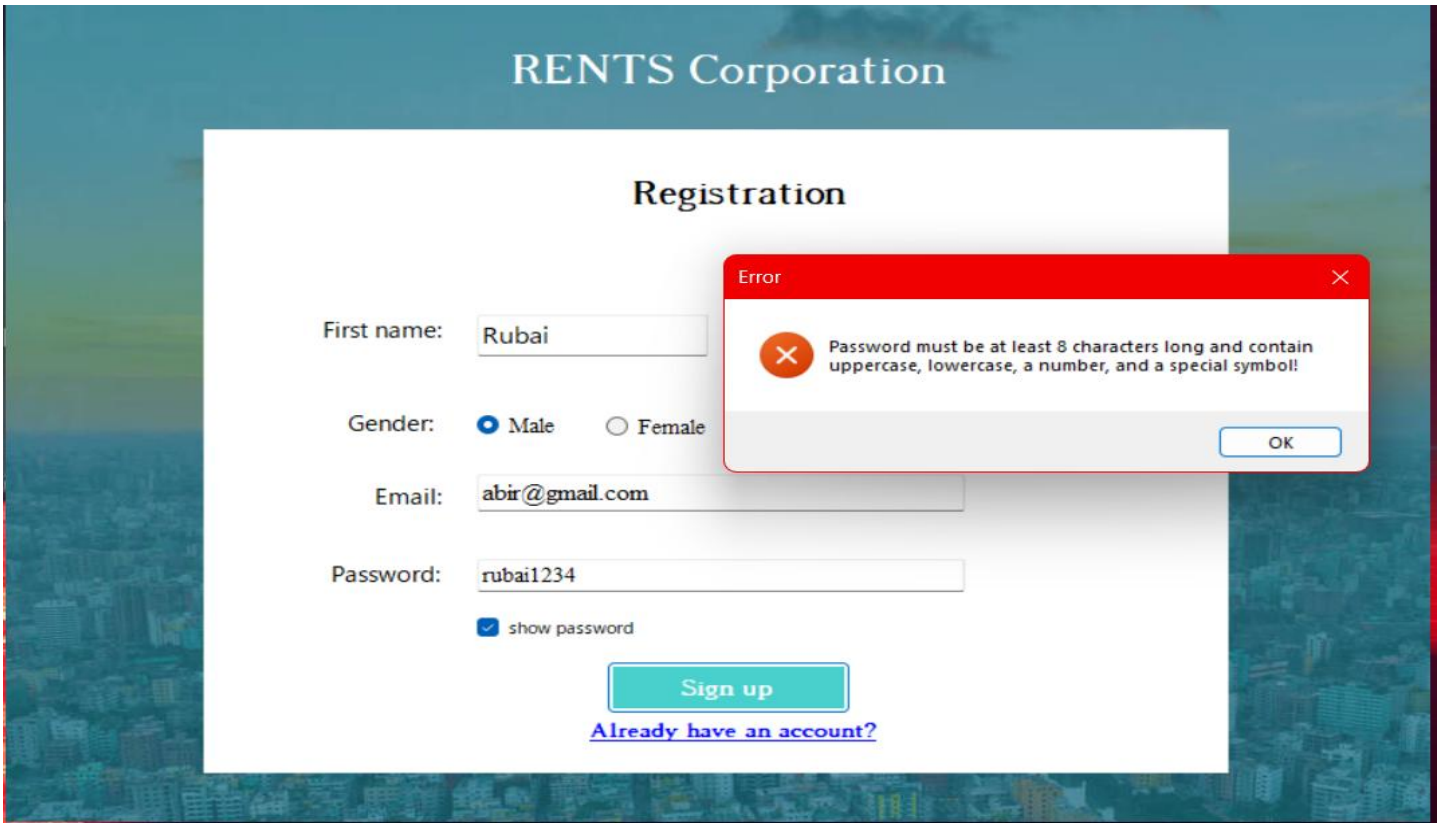


Fig 2: Registration page

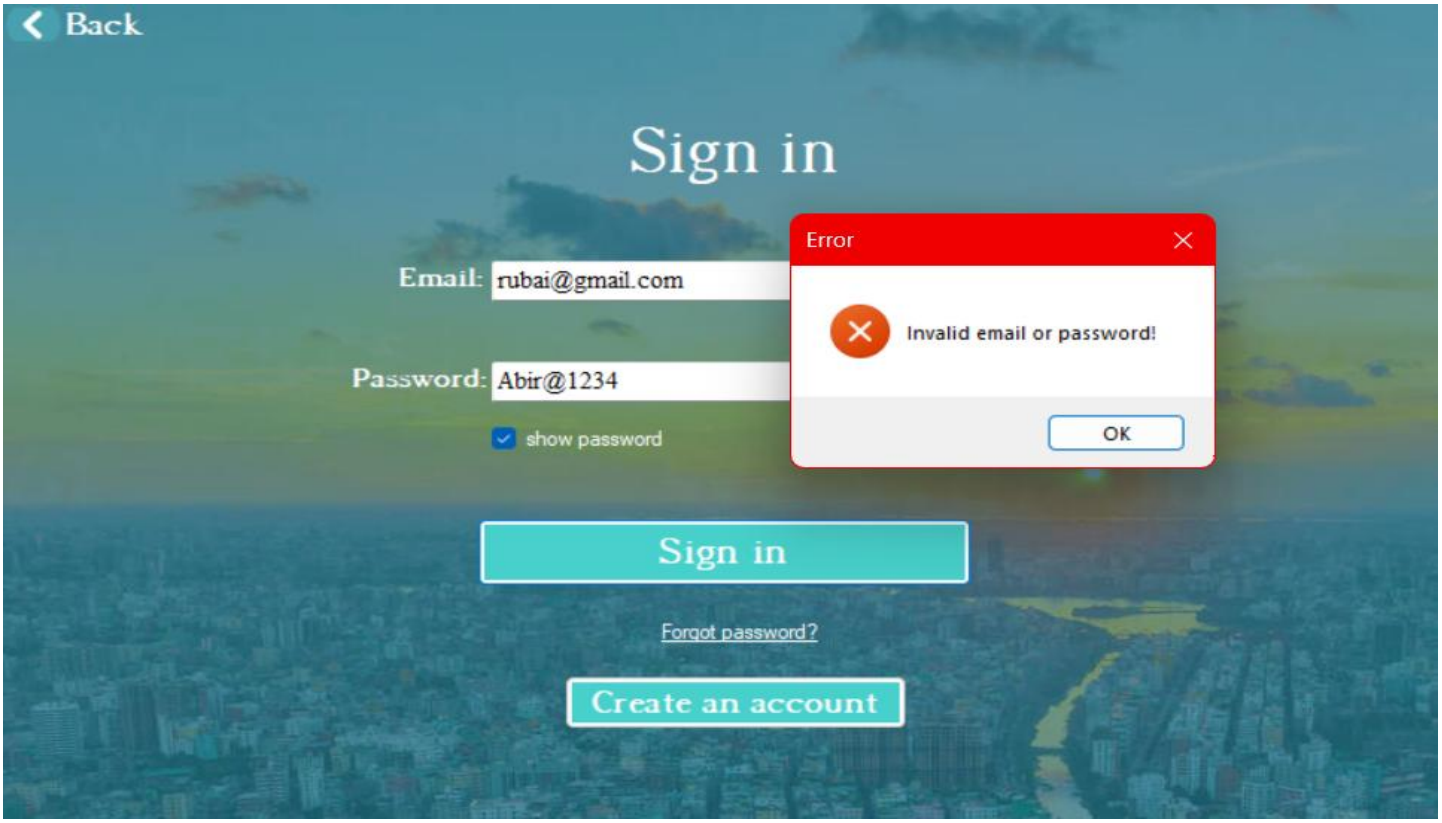


Fig 3: Sign in page

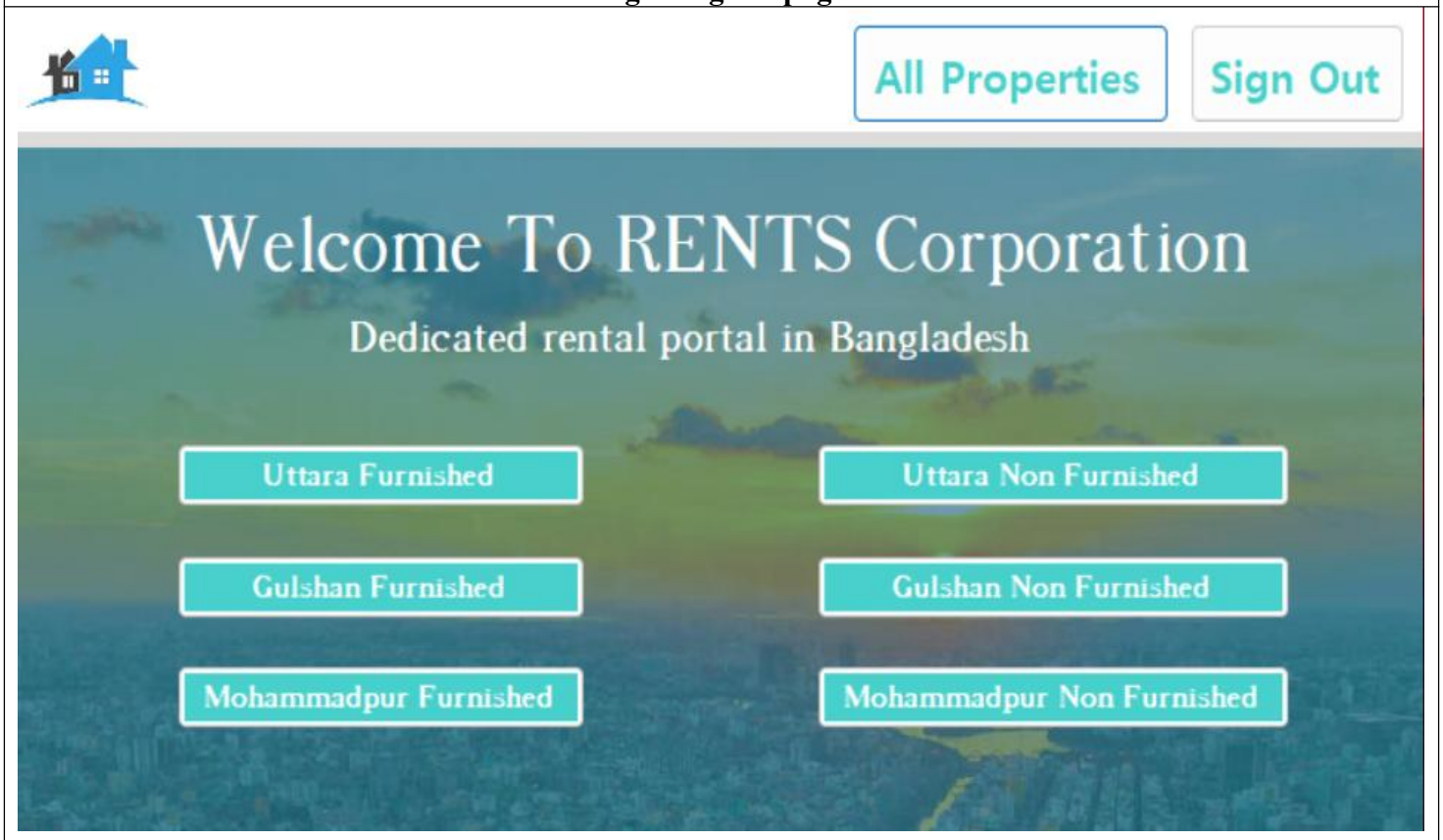


Fig 4: Tenant page

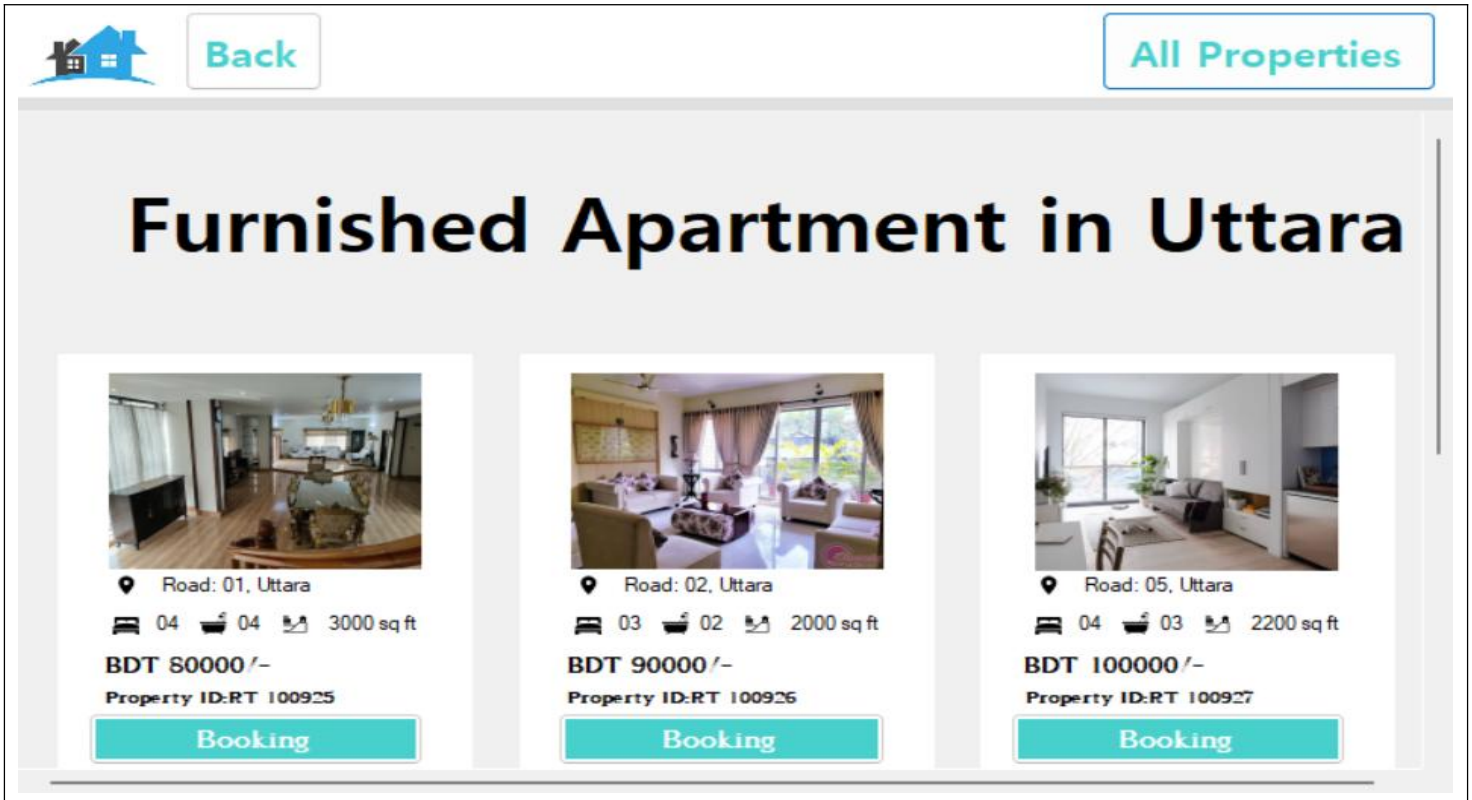


Fig 5: Uttara Furnished Apartment

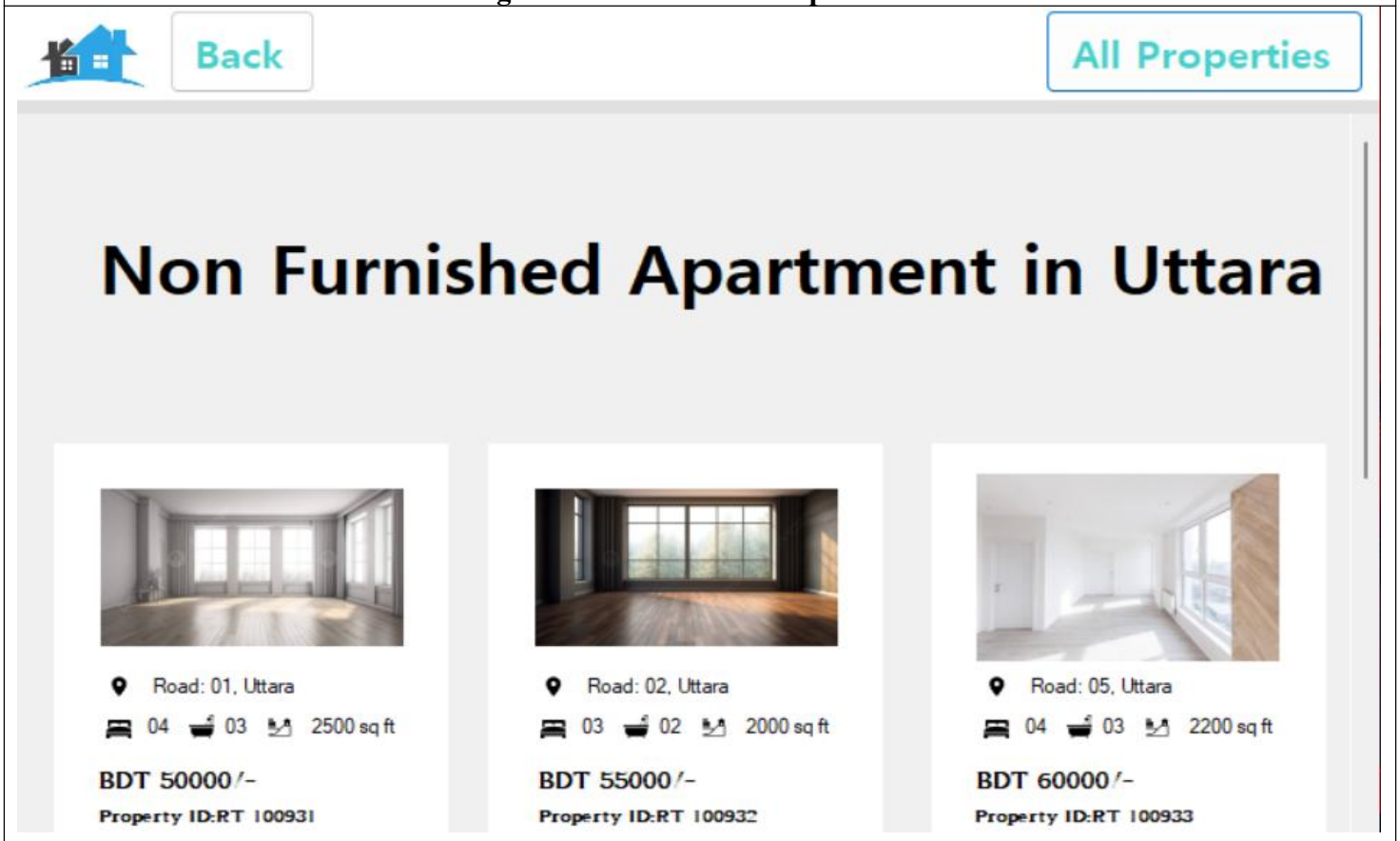


Fig 6:Uttara Non Furnished Apartment

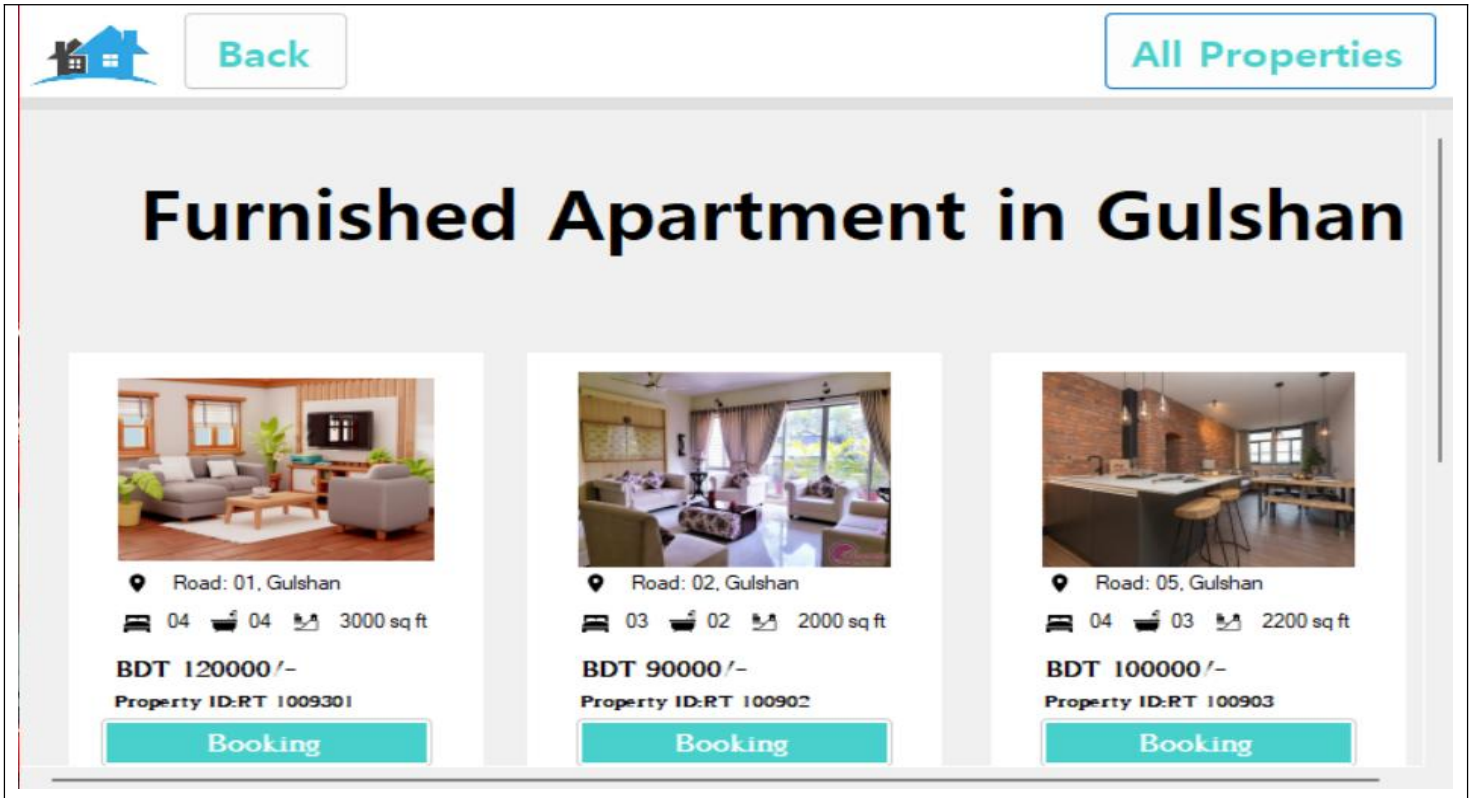


Fig 7: Gulshan Furnished Apartment

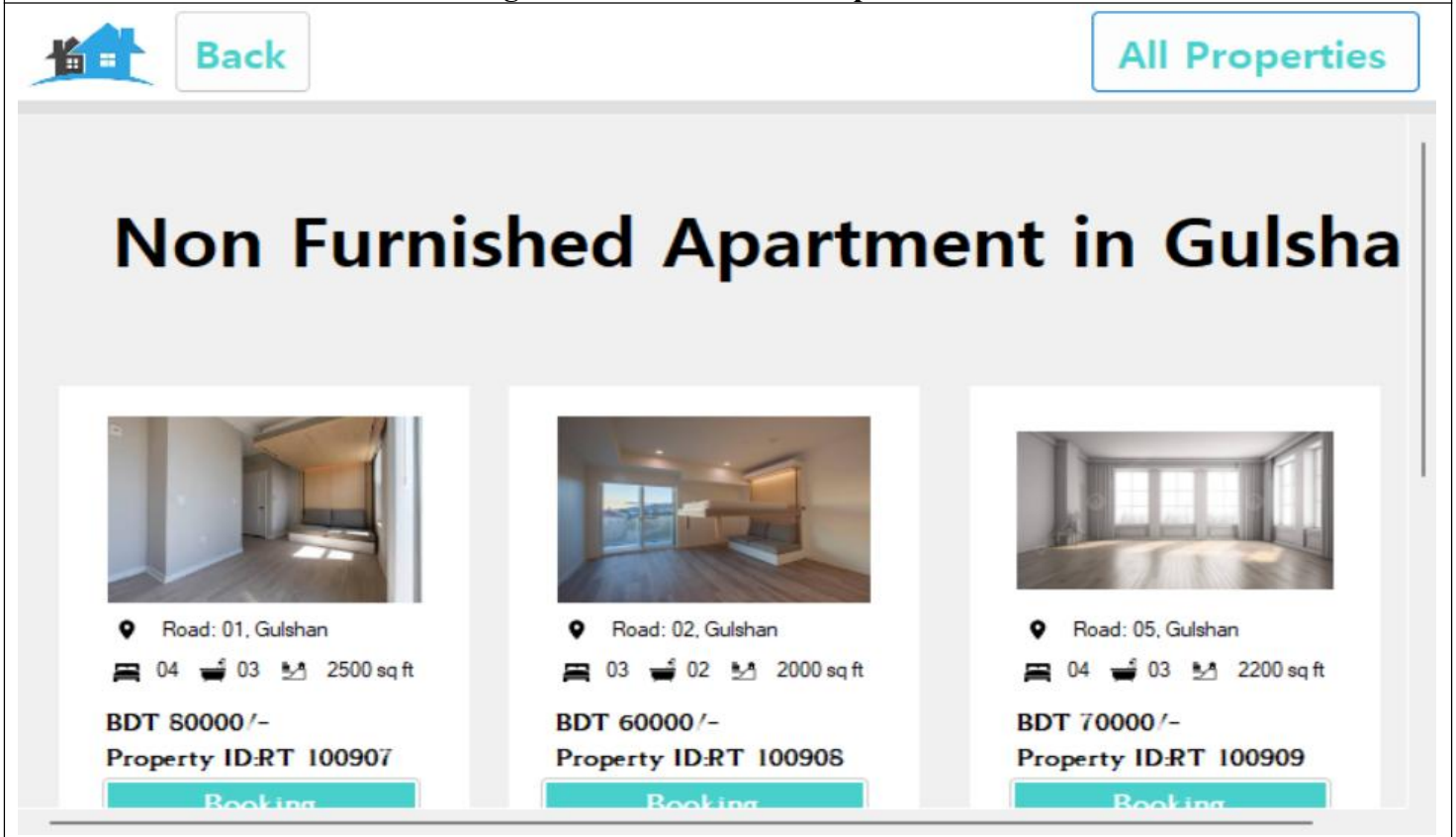


Fig 8: Gulshan Non Furnished Apartment

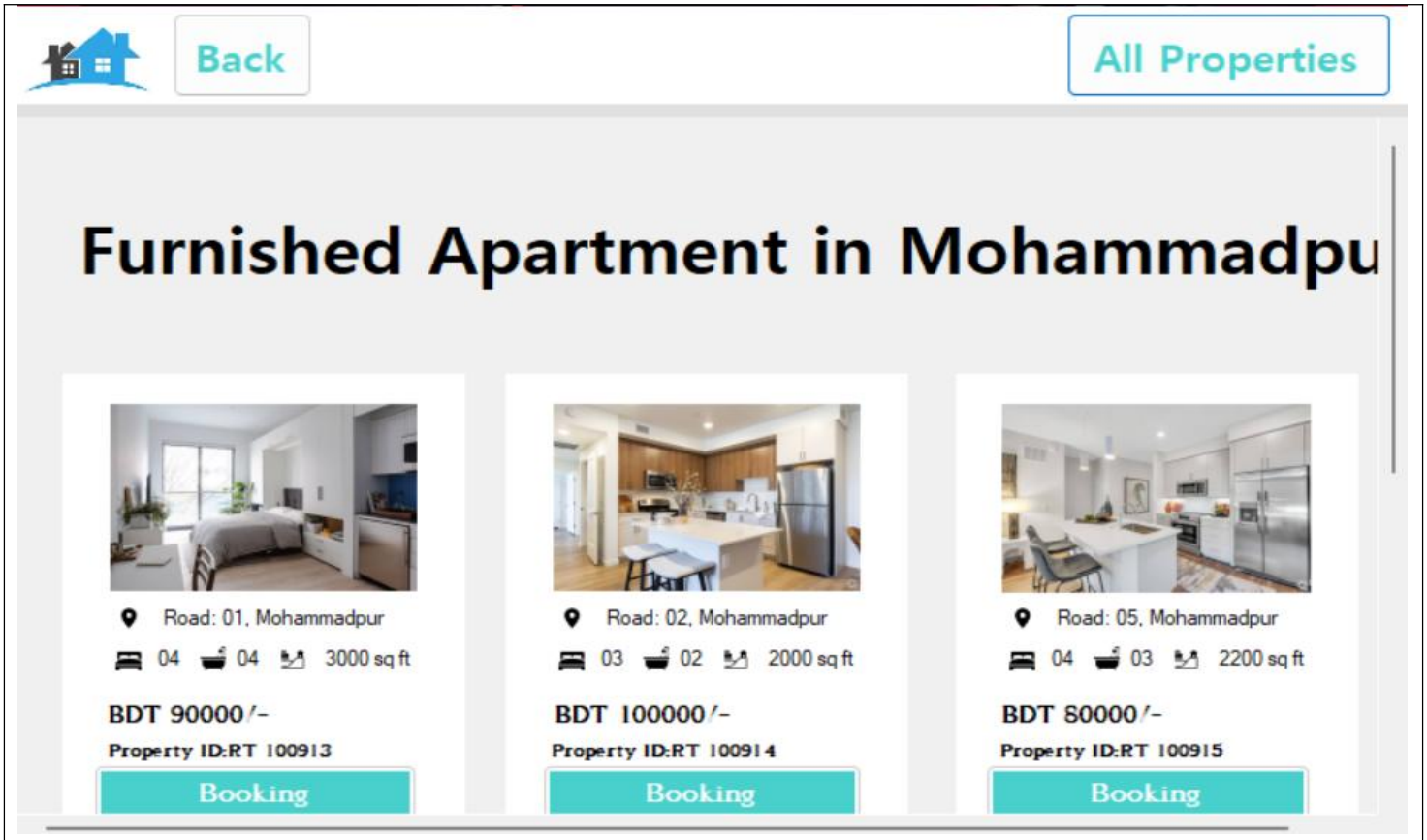


Fig 10: Mohammadpur Furnished Apartment

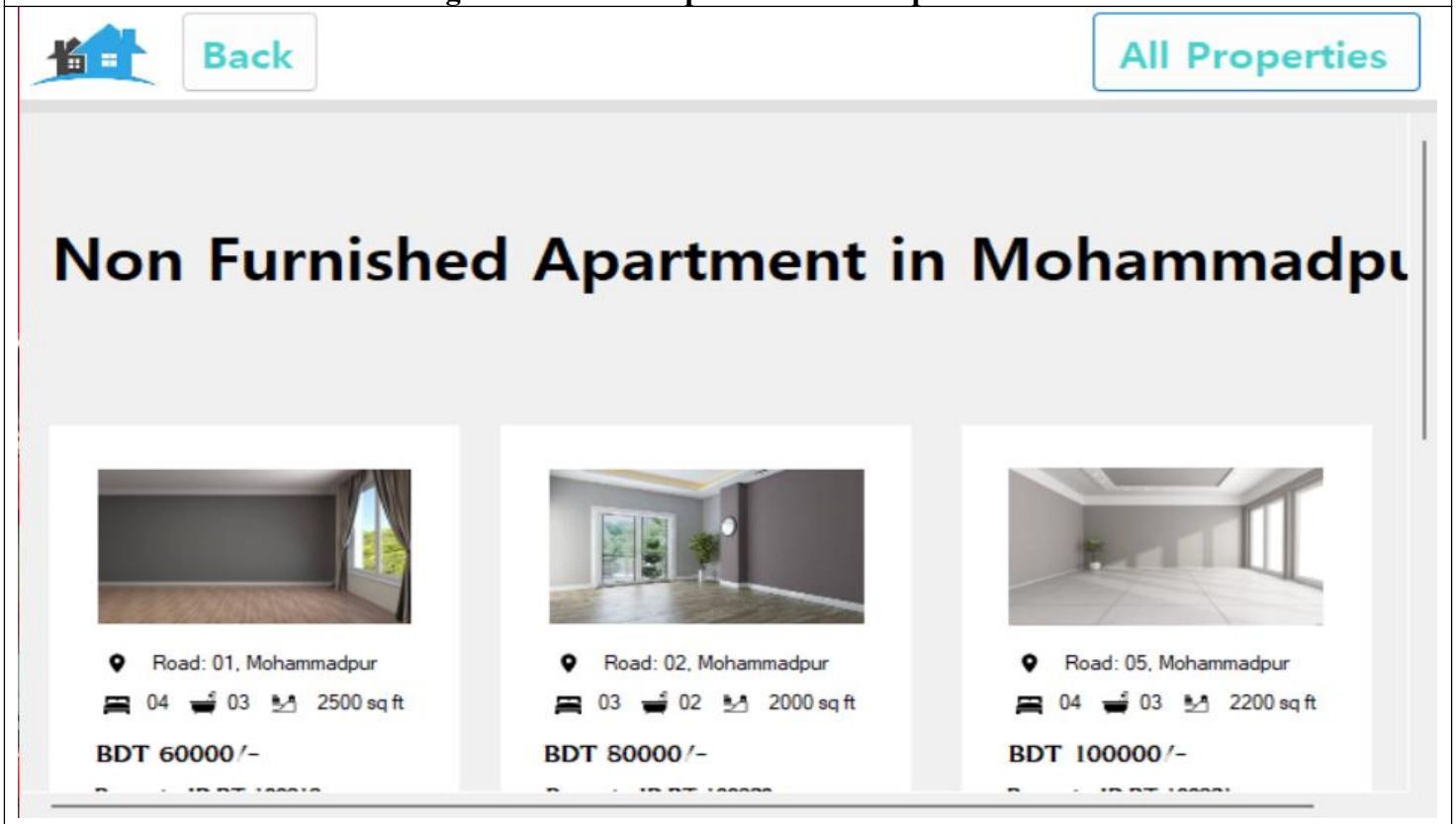


Fig 11: Mohammadpur Non Furnished Apartment

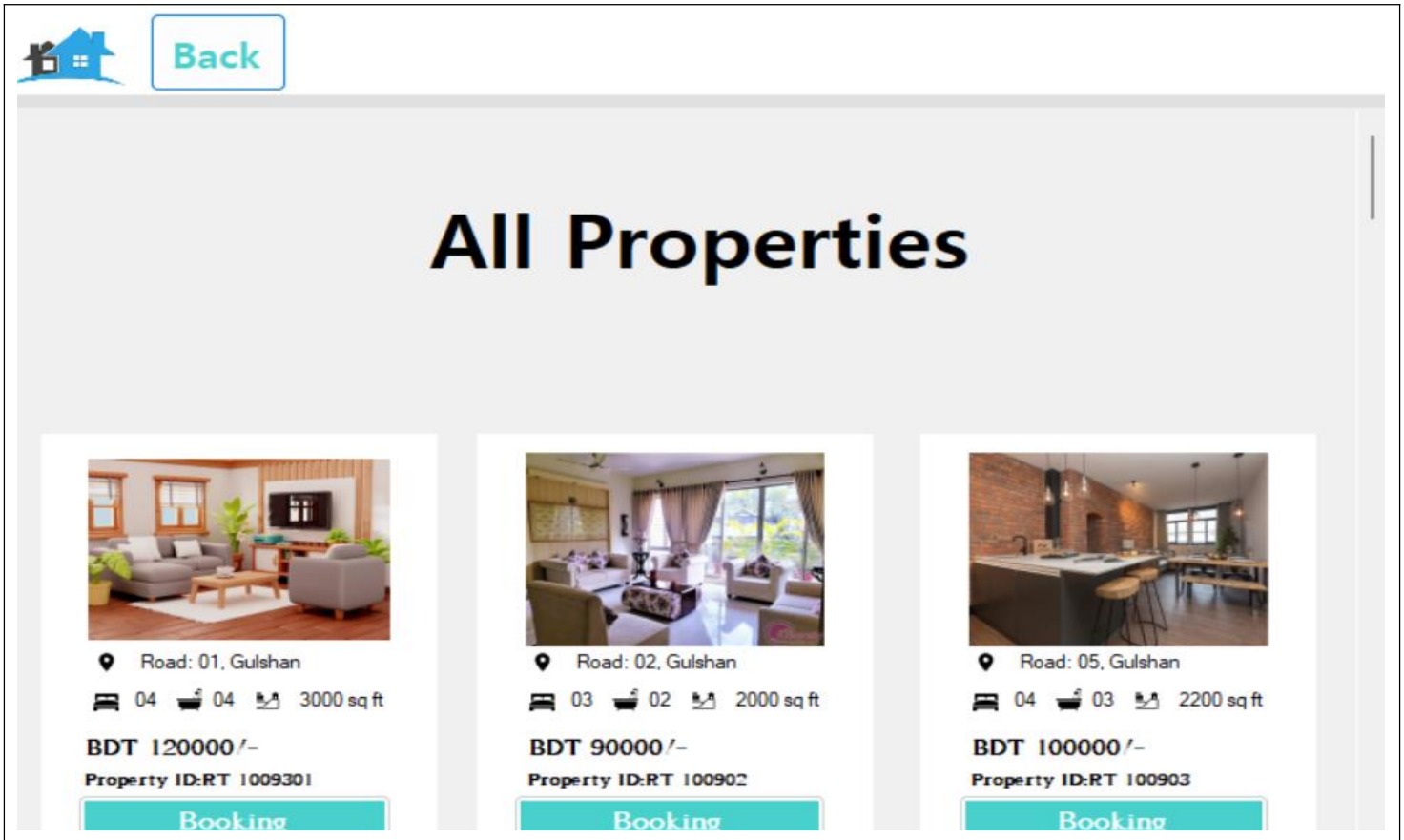


Fig 12: All Properties Page

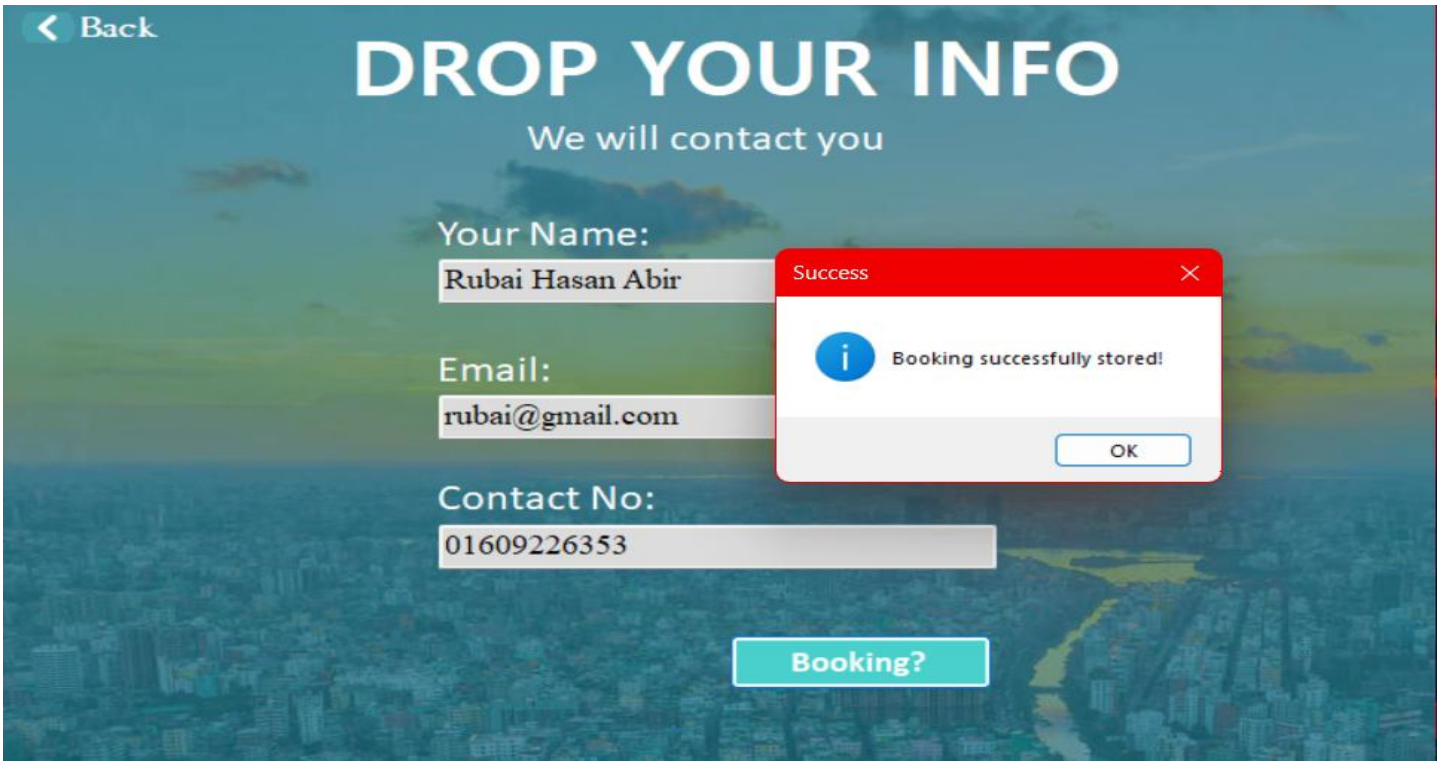


Fig 13:Booking Informtion

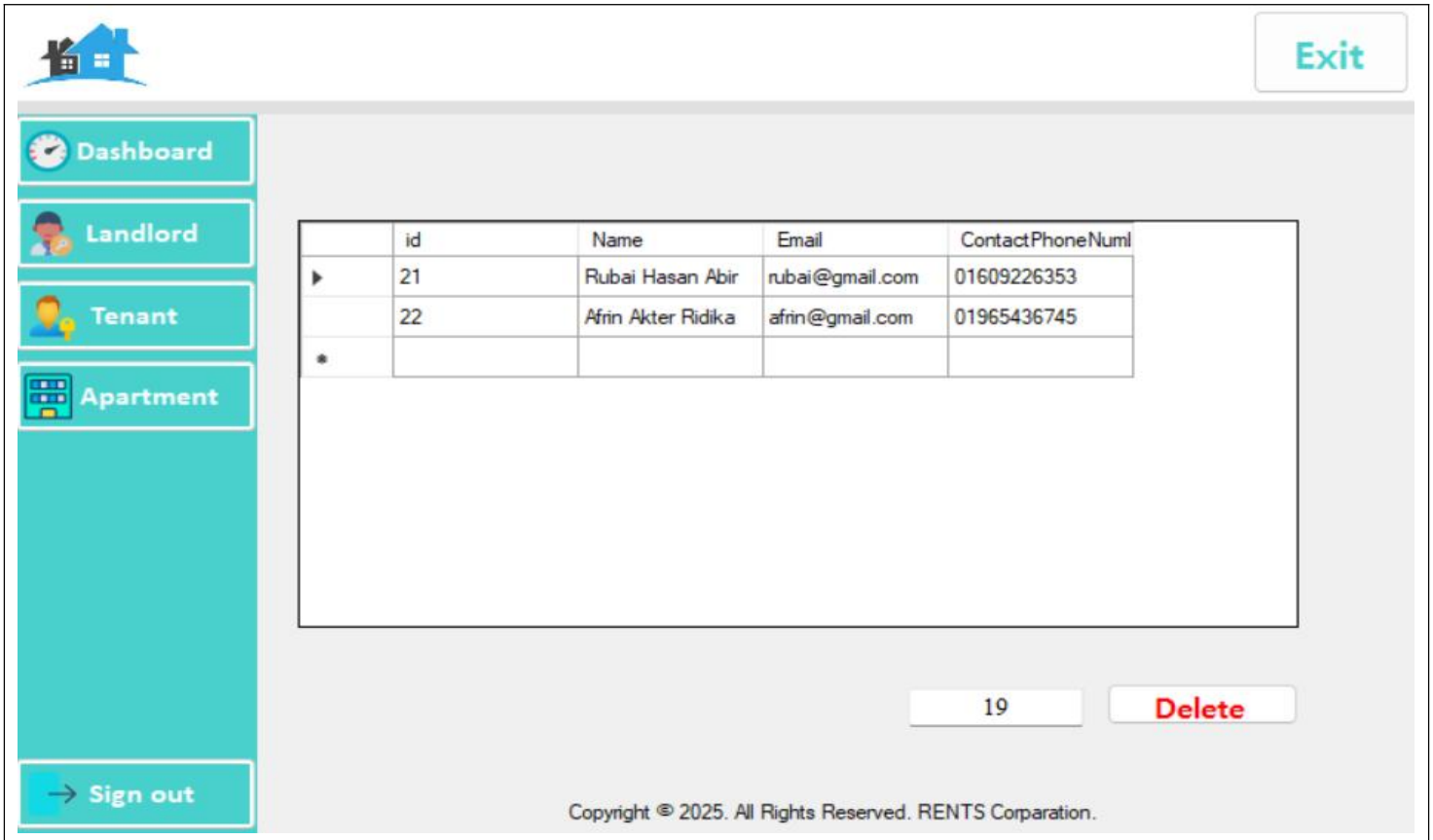


Fig 14: Admin Dashboard (Tenant)

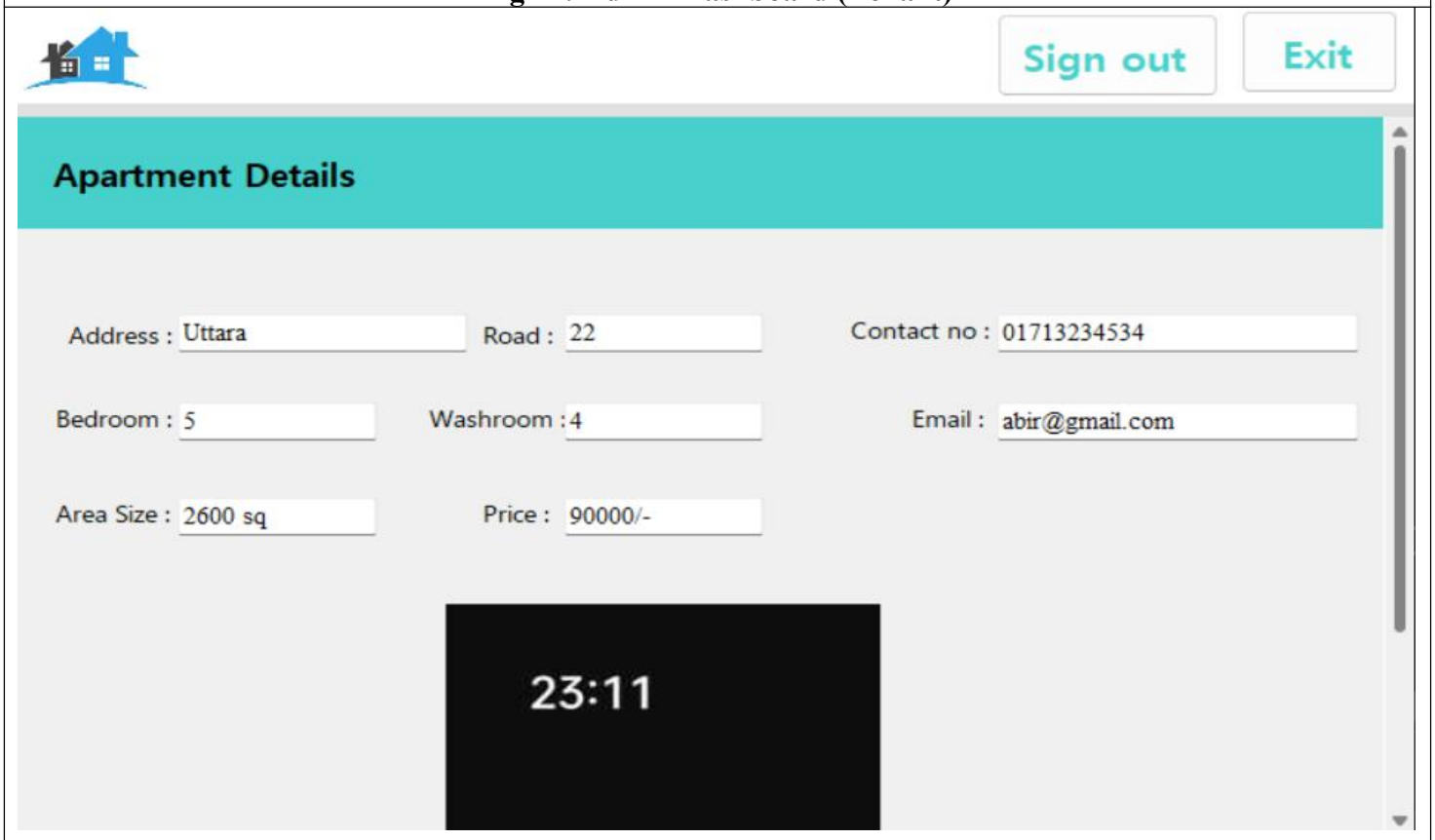


Fig 15: Property Details for Add

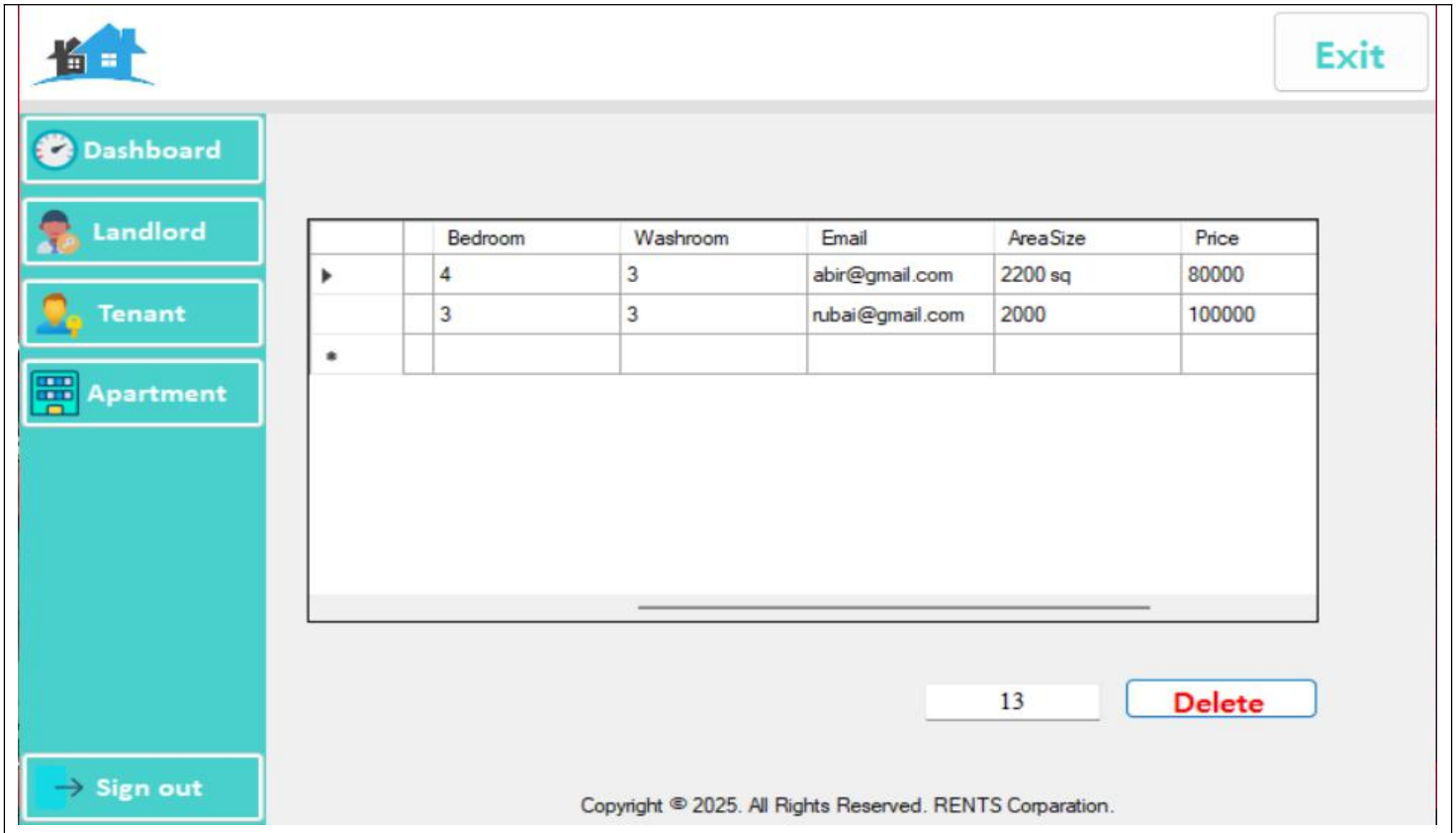


Fig 16: Admin Dashboard(Landlord)

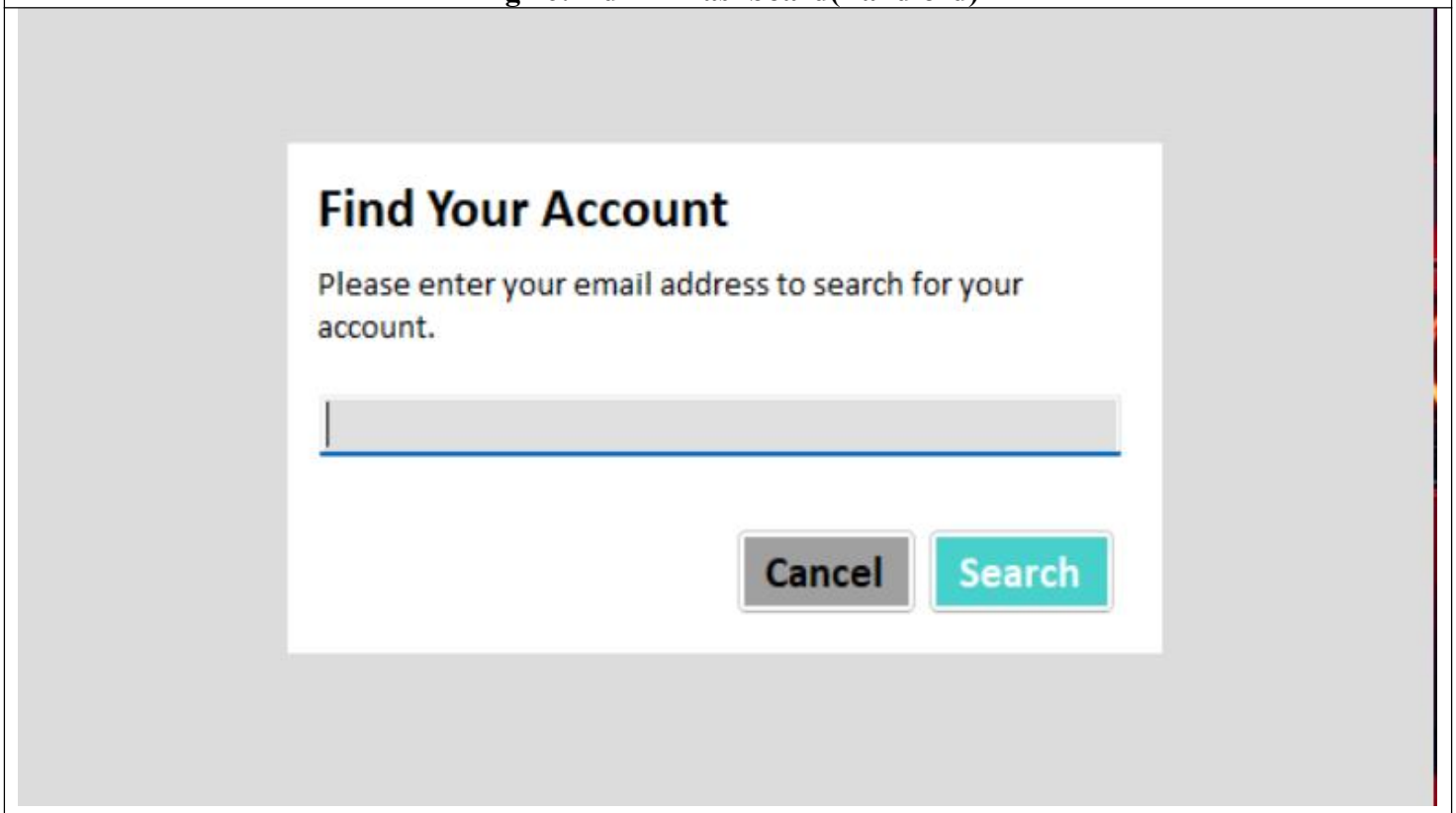


Fig 17: Forgot Password

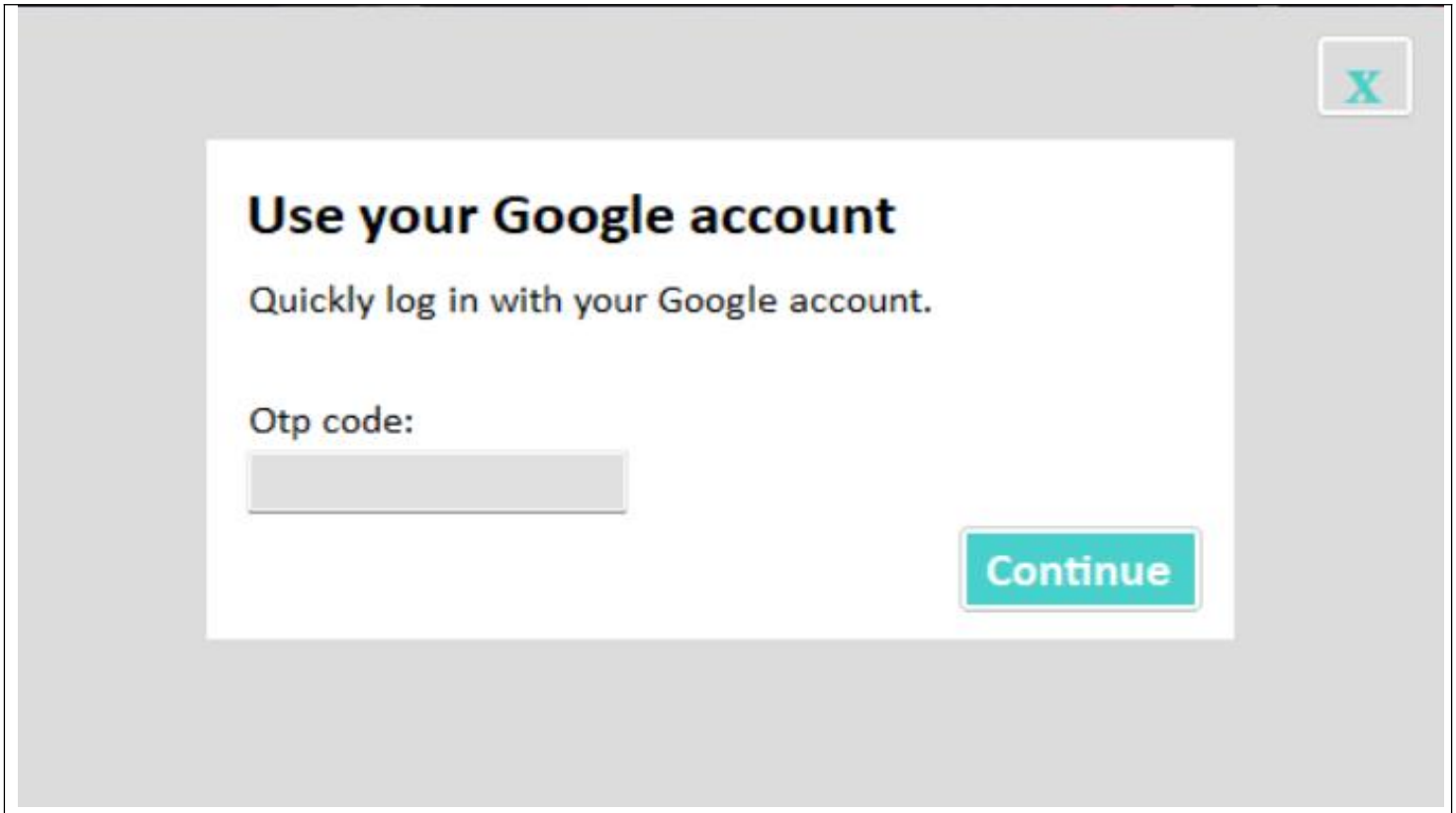

X
Use your Google account
Quickly log in with your Google account.
Otp code:
Continue

Fig 18: OTP