MAHATMA EDUCATION SOCIETY'S

PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE

(Autonomous)
NEW PANVEL

PROJECT REPORT ON

"Online Property Rental or Buying System"

IN PARTIAL FULFILLMENT OF

BACHELOR OF COMPUTER SCIENCE

SEMESTER - 2024 - 25

PROJECT GUIDE
Name: Smita Hadawale Kakade

SUBMITTED BY: Sneha Kumari

ROLL NO: 5357

INTRODUCTION :

The **Online Property Rental and Buying System** is a web-based platform designed to simplify the process of renting and purchasing properties. It provides a seamless experience for users to search, list, and manage properties efficiently. The system connects property owners, buyers in a secure and organized manner, reducing the need for traditional paperwork and in-person visits.

Users can explore various property options based on their preferences, such as type, location, and price range. The platform allows property owners to add,delete , update their properties with detailed descriptions. while users can browse listings, buy/rent the properties.

This system integrates key functionalities like user authentication, property search filters and admin control.

register.jsp

```
<html>
<head>
   <meta charset="UTF-8">
   <title>Register</title>
   <link rel="stylesheet" type="text/css" href="CSS/register.css">
</head>
<body>
   <div class="form-container">
      <h2>Register</h2>
      <form action="RegisterUserServlet" method="POST">
         <label for="name">Name:</label>
         <input type="text" id="name" name="name" required placeholder="Enter your
full name">

         <label for="email">Email:</label>
         <input type="email" id="email" name="email" required placeholder="Enter your
email address">

         <label for="password">Password:</label>
         <input type="password" id="password" name="password" required
placeholder="Enter your password">
```

```
            <button type="submit">Register</button>
        </form>
    </div>
</body>
</html>
```

RegisterUserServlet.java

```
@WebServlet("/RegisterUserServlet")
public class RegisterUserServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String password = request.getParameter("password");

        try (Connection conn = DatabaseUtil.getConnection()) {
            String query = "INSERT INTO users (name, email, password, is_admin)
VALUES (?, ?, ?, FALSE)";
            try (PreparedStatement stmt = conn.prepareStatement(query)) {
                stmt.setString(1, name);
                stmt.setString(2, email);
                stmt.setString(3, password);
                stmt.executeUpdate();
            }
            response.sendRedirect("login.jsp");
        } catch (SQLException e) {
            e.printStackTrace();
            response.getWriter().println("<h3>Registration failed: " + e.getMessage() +
"</h3>");
        }}}
```

login.jsp

```
<body>
    <div class="login-container">
        <h2>Login</h2>
        <form action="UserLoginServlet" method="post">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>

            <button type="submit">Login</button>
        </form>
    </div>
</body>
```

UserLoginServlet.java

```
@WebServlet("/UserLoginServlet")
public class UserLoginServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String email = request.getParameter("email");
        String password = request.getParameter("password");

        try (Connection conn = DatabaseUtil.getConnection();
             PreparedStatement stmt = conn.prepareStatement("SELECT id, is_admin FROM users
WHERE email = ? AND password = ?")) {

            stmt.setString(1, email);
            stmt.setString(2, password);

            try (ResultSet rs = stmt.executeQuery()) {
                if (rs.next()) {
                    int userId = rs.getInt("id");
                    boolean isAdmin = rs.getBoolean("is_admin");

                    HttpSession session = request.getSession();
                    session.setAttribute("userId", userId);
                    session.setAttribute("isAdmin", isAdmin);


                    System.out.println("isAdmin: " + isAdmin);

                    if (isAdmin) {
                        System.out.println("Admin login successful, redirecting...");
                        response.sendRedirect("adminListpropertyServlet");
                    } else {
                        System.out.println("User login successful, redirecting...");
                        response.sendRedirect("UserProfileServlet");
                    }
                } else {
                    System.out.println("Login failed, forwarding to login.jsp");
                    request.setAttribute("errorMessage", "Invalid email or password.");
                    request.getRequestDispatcher("login.jsp").forward(request, response);
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
            out.println("<h3>An error occurred: " + e.getMessage() + "</h3>");
        }}}
```

UserProfileServlet.java

```
@WebServlet("/UserProfileServlet")
public class UserProfileServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession(false);

        if (session == null || session.getAttribute("userId") == null) {
            response.sendRedirect("login.jsp");
            return;
        }
        int userId = (int) session.getAttribute("userId");

        try (Connection conn = DatabaseUtil.getConnection();
             PreparedStatement ps = conn.prepareStatement("SELECT name, email FROM users
WHERE id = ?")) {
            ps.setInt(1, userId);

            try (ResultSet rs = ps.executeQuery()) {
                if (rs.next()) {
                    request.setAttribute("name", rs.getString("name"));
                    request.setAttribute("email", rs.getString("email"));
                } else {
                    response.sendRedirect("error.jsp");
                    return;
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
            response.sendRedirect("error.jsp");
            return;
        }

        request.getRequestDispatcher("user_profile.jsp").forward(request, response);
    }
}
```

user_profile.jsp

```
<body>
    <%
        Integer userId = (Integer) session.getAttribute("userId");

        if (userId == null) {
            response.sendRedirect("login.jsp");
            return;
        }
    %>

    <div class="profile-container">
        <h1>User Profile</h1>
        <p><strong>Name:</strong> ${name}</p>
        <p><strong>Email:</strong> ${email}</p>

        <h2>Edit Profile</h2>
        <form action="UpdateProfileServlet" method="post">
            <input type="hidden" name="id" value="${userId}">

            <label for="name">Name:</label>
            <input type="text" id="name" name="name" value="${name}" required>

            <label for="email">Email:</label>
            <input type="email" id="email" name="email" value="${email}" required>

            <button type="submit" class="btn" aria-label="Update Profile">Update
Profile</button>
        </form>

        <a href="LogoutServlet" class="logout" aria-label="Logout">Logout</a>
    <a href="index.jsp" class="btn" aria-label="Go to Home">Go to Home</a>


  <form action="DeleteProfileServlet" method="post"
     onsubmit="return confirm('Are you sure you want to delete your account? This
action cannot be undone.');"
     style="text-align: center; margin-top: 20px;">
    <input type="hidden" name="id" value="${userId}">
    <button type="submit" class="btn delete-btn" aria-label="Delete Profile">Delete
Profile</button>
  </form>

    </div>
</body>
```

index.jsp

```
<h1>Welcome to Property Management</h1>
<form action="SearchPropertyServlet" method="POST">

 <label>Property Type:</label>
  <select name="type" required>
    <option value="1BHK">1BHK</option>
    <option value="2BHK">2BHK</option>
    <option value="3BHK">3BHK</option>
    <option value="Chawl">Chawl</option>
  </select><br>

  <label>Location:</label>
  <input type="text" name="location" id="location" required><br>

  <label>Available Buildings:</label>
  <select name="building" id="building">
    <option value="">Select a building</option>
    <%
      Connection con = null;
      Statement stmt = null;
      ResultSet rs = null;
      try {
        con = com.property_management.utils.DatabaseUtil.getConnection();
        stmt = con.createStatement();
        rs = stmt.executeQuery("SELECT DISTINCT building FROM properties");

        while (rs.next()) {
    %>
          <option value="<%= rs.getString("building") %>"><%=
rs.getString("building") %></option>
    <%
        }
      } catch (SQLException e) {
    %>
        <option value="">Error fetching buildings</option>
    <%
        e.printStackTrace();
      } finally {
        if (rs != null) try { rs.close(); } catch (SQLException ignore) {}
        if (stmt != null) try { stmt.close(); } catch (SQLException ignore) {}
        if (con != null) try { con.close(); } catch (SQLException ignore) {}
      }
    %>
  </select><br>

  <label>Min Price:</label>
  <input type="number" name="minPrice" placeholder="Minimum price" min="0"><br>

  <label>Max Price:</label>
  <input type="number" name="maxPrice" placeholder="Maximum price" min="0"><br>

  <label>Type:</label>
  <div class="radio-group">
    <input type="radio" name="propertyType" value="buy" required> Buy
    <input type="radio" name="propertyType" value="rent"> Rent <br><br>
  </div>

  <button type="submit">Search</button>
</form>
```

SearchPropertyServlet.java

```
@WebServlet("/SearchPropertyServlet")
public class SearchPropertyServlet extends HttpServlet {
    private static final Logger logger =
Logger.getLogger(SearchPropertyServlet.class.getName());

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        List<Object[]> searchedProperties = new ArrayList<>();

        try (Connection con =
com.property_management.utils.DatabaseUtil.getConnection()) {
            if (con == null) {
                request.setAttribute("errorMessage", "Database connection failed.");
                request.getRequestDispatcher("search_result.jsp").forward(request,
response);
                return;
            }

            String type = request.getParameter("type");
            String location = request.getParameter("location");
            String building = request.getParameter("building");
            String minPriceStr = request.getParameter("minPrice");
            String maxPriceStr = request.getParameter("maxPrice");
            String propertyType = request.getParameter("propertyType");

            StringBuilder sql = new StringBuilder("SELECT id, type, location, building, price,
property_type FROM properties WHERE type = ? AND location = ? AND property_type
= ?");
            if (building != null && !building.isEmpty()) sql.append(" AND building = ?");
            if (minPriceStr != null && !minPriceStr.isEmpty()) sql.append(" AND price >= ?");
            if (maxPriceStr != null && !maxPriceStr.isEmpty()) sql.append(" AND price <=
?");

            try (PreparedStatement stmt = con.prepareStatement(sql.toString())) {
                int paramIndex = 1;
                stmt.setString(paramIndex++, type);
                stmt.setString(paramIndex++, location);
                stmt.setString(paramIndex++, propertyType);
                if (building != null && !building.isEmpty()) stmt.setString(paramIndex++,
building);
```

```
                if (minPriceStr != null && !minPriceStr.isEmpty()) stmt.setInt(paramIndex++,
Integer.parseInt(minPriceStr));
                if (maxPriceStr != null && !maxPriceStr.isEmpty()) stmt.setInt(paramIndex++,
Integer.parseInt(maxPriceStr));

                try (ResultSet rs = stmt.executeQuery()) {
                    while (rs.next()) {
                        searchedProperties.add(new Object[]{
                            rs.getInt("id"),
                            rs.getString("type"),
                            rs.getString("location"),
                            rs.getString("building"),
                            rs.getInt("price"),
                            rs.getString("property_type")
                        });
                    }
                }
            }
        } catch (SQLException ex) {
            logger.log(Level.SEVERE, "Database error", ex);
            request.setAttribute("errorMessage", "Database error occurred.");
        }

        request.setAttribute("searchedProperties", searchedProperties);
        request.getRequestDispatcher("search_result.jsp").forward(request, response);
    }
}
```

search_result.jsp

```
<body>

<h1>Search Results</h1>

<c:if test="${not empty errorMessage}">
    <p style="color: red;">${errorMessage}</p>
</c:if>

<div class="container">
    <c:forEach var="property" items="${searchedProperties}">
        <div class="card">
```

```
            <h2>${property[1]}</h2>
            <p><strong>Location:</strong> ${property[2]}</p>
            <p><strong>Building:</strong> ${property[3]}</p>
            <p><strong>Price:</strong> $${property[4]}</p>
            <p><strong>Property Type:</strong> ${property[5]}</p>

           <c:choose>
    <c:when test="${property[5] eq 'sale' or property[5] eq 'buy'}">
        <form action="BuyPropertyServlet" method="POST">
            <input type="hidden" name="id" value="${property[0]}">
            <button type="submit" class="btn btn-buy">Buy</button>
        </form>
    </c:when>
    <c:when test="${property[5] eq 'rent'}">
        <form action="RentPropertyServlet" method="POST">
            <input type="hidden" name="id" value="${property[0]}">
            <button type="submit" class="btn btn-rent">Rent</button>
        </form>
    </c:when>
</c:choose>
        </div>
    </c:forEach>
</div>

</body>
```

admin_dashboard.jsp

```
<h1>Admin Dashboard</h1>
<div class="container">
    <c:forEach var="property" items="${properties}">
        <div class="card">
            <h2>${property[1]}</h2>
            <p><strong>Location:</strong> ${property[2]}</p>
            <p><strong>Building:</strong> ${property[3]}</p>
            <p><strong>Price:</strong> $${property[4]}</p>
            <p><strong>Property Type:</strong> ${property[5]}</p>
           <form action="updateProperty.jsp" method="GET">
    <input type="hidden" name="id" value="${property[0]}">
    <input type="hidden" name="type" value="${property[1]}">
    <input type="hidden" name="location" value="${property[2]}">
    <input type="hidden" name="building" value="${property[3]}">
    <input type="hidden" name="price" value="${property[4]}">
    <input type="hidden" name="propertyType" value="${property[5]}">
    <button type="submit" class="btn btn-buy">Update Property</button>
</form>


            <form action="DeletePropertyServlet" method="POST">
                <input type="hidden" name="id" value="${property[0]}">
                <input type="hidden" name="type" value="${property[1]}">
                <input type="hidden" name="location" value="${property[2]}">
                <input type="hidden" name="building" value="${property[3]}">
                <input type="hidden" name="price" value="${property[4]}">
                <input type="hidden" name="propertyType" value="${property[5]}">
                <button type="submit" class="btn btn-rent">Delete property</button>
            </form>
        </div>
    </c:forEach>
</div>
```

updateProperty.jsp

```
<body>

<h2>Update Property</h2>

<form action="UpdatePropertyServlet" method="POST">
    <input type="hidden" name="id" value="<%= request.getParameter("id") %>">

    <label for="type">Type:</label>
    <input type="text" id="type" name="type" value="<%= request.getParameter("type")
%>" required>

    <label for="location">Location:</label>
    <input type="text" id="location" name="location" value="<%=
request.getParameter("location") %>" required>

    <label for="building">Building:</label>
    <input type="text" id="building" name="building" value="<%=
request.getParameter("building") %>" required>

    <label for="price">Price:</label>
    <input type="number" id="price" name="price" value="<%=
request.getParameter("price") %>" required>

    <label for="propertyType">Property Type:</label>
    <input type="text" id="propertyType" name="propertyType" value="<%=
request.getParameter("propertyType") %>" required>

    <button type="submit">Update Property</button>
</form>
</body>
</html>
```

UpdatePropertyServlet.java

```
@WebServlet("/UpdatePropertyServlet")
public class UpdatePropertyServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        int propertyId = Integer.parseInt(request.getParameter("id"));
        String type = request.getParameter("type");
        String location = request.getParameter("location");
        String building = request.getParameter("building");
        int price = Integer.parseInt(request.getParameter("price"));
        String propertyType = request.getParameter("propertyType");

        try (Connection conn = DatabaseUtil.getConnection()) {
            String sql = "UPDATE properties SET type = ?, location = ?, building = ?, price =
?, property_type = ? WHERE id = ?";
            try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                stmt.setString(1, type);
                stmt.setString(2, location);
                stmt.setString(3, building);
                stmt.setInt(4, price);
                stmt.setString(5, propertyType);
                stmt.setInt(6, propertyId);

                int rowsUpdated = stmt.executeUpdate();
                if (rowsUpdated > 0) {
                    System.out.println("Property updated successfully!");
                } else {
                    System.out.println("Property update failed!");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
            response.getWriter().println("<h3>Failed to update property: " + e.getMessage()
+ "</h3>");
            return;
        }
        response.sendRedirect("adminListpropertyServlet");
    }
}
```

ListPropertyServlet.java

```
@WebServlet("/ListPrope
rtyServlet")
public class ListPropertyServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        List<String[]> properties = new ArrayList<>();

        try (Connection conn = DatabaseUtil.getConnection()) {
            if (conn == null) {
                throw new Exception("Failed to connect to the database.");
            }

            String sql = "SELECT * FROM properties";
            try (Statement stmt = conn.createStatement();
                 ResultSet rs = stmt.executeQuery(sql)) {

                while (rs.next()) {
                    String[] property = new String[6];
                    property[0] = String.valueOf(rs.getInt("id"));
                    property[1] = rs.getString("type");
                    property[2] = rs.getString("location");
                    property[3] = rs.getString("building");
                    property[4] = String.valueOf(rs.getInt("price"));
                    property[5] = rs.getString("property_type");
                    properties.add(property);
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        request.setAttribute("properties", properties);
        request.getRequestDispatcher("property_list.jsp").forward(request, response);
    }
```

property_list.jsp

```
<h1>Available Properties</h1>
<div class="container">
    <c:forEach var="property" items="${properties}">
        <div class="card">
            <h2>${property[1]}</h2>
            <p><strong>Location:</strong> ${property[2]}</p>
            <p><strong>Building:</strong> ${property[3]}</p>
            <p><strong>Price:</strong> $${property[4]}</p>
            <p><strong>Property Type:</strong> ${property[5]}</p>

            <form action="BuyPropertyServlet" method="POST">
                <input type="hidden" name="id" value="${property[0]}">
                <input type="hidden" name="type" value="${property[1]}">
                <input type="hidden" name="location" value="${property[2]}">
                <input type="hidden" name="building" value="${property[3]}">
                <input type="hidden" name="price" value="${property[4]}">
                <input type="hidden" name="propertyType" value="${property[5]}">
                <button type="submit" class="btn btn-buy">Buy</button>
            </form>
            <form action="RentPropertyServlet" method="POST">
                <input type="hidden" name="id" value="${property[0]}">
                <input type="hidden" name="type" value="${property[1]}">
                <input type="hidden" name="location" value="${property[2]}">
                <input type="hidden" name="building" value="${property[3]}">
                <input type="hidden" name="price" value="${property[4]}">
                <input type="hidden" name="propertyType" value="${property[5]}">
                <button type="submit" class="btn btn-rent">Rent</button>
            </form>
        </div>
    </c:forEach>
</div>
```

add_property.jsp

```
<body>

    <div class="form-container">
        <h1>Add New Property</h1>
        <form action="AddPropertyServlet" method="post">
            <label for="type">Type:</label>
            <input type="text" id="type" name="type" required>

            <label for="location">Location:</label>
            <input type="text" id="location" name="location" required>

            <label for="building">Building:</label>
            <input type="text" id="building" name="building" required>

            <label for="price">Price:</label>
            <input type="number" id="price" name="price" required>

            <label for="property_type">Property Type:</label>
            <input type="text" id="property_type" name="property_type" required>

            <input type="submit" value="Add Property">
        </form>
    </div>
</body>
</html>
```

AddPropertyServlet.java

```
public class AddPropertyServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String type = request.getParameter("type");
        String location = request.getParameter("location");
        String building = request.getParameter("building");
        int price = Integer.parseInt(request.getParameter("price"));
        String property_type = request.getParameter("property_type");

        try (Connection conn = DatabaseUtil.getConnection()) {
            String sql = "INSERT INTO properties (type, location, building, price,
property_type) VALUES (?, ?, ?, ?, ?)";
            try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                stmt.setString(1, type);
                stmt.setString(2, location);
                stmt.setString(3, building);
                stmt.setInt(4, price);
                stmt.setString(5, property_type);
                stmt.executeUpdate();  }
            response.sendRedirect("ListPropertyServlet");
        } catch (Exception e) {
            e.printStackTrace();
        }    }}
```

DatabaseUtil.java

```
package com.property_management.utils;

import java.sql.*;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseUtil {

    private static final String JDBC_URL =
"jdbc:mysql://localhost:3306/property_mgmt?useSSL=false";
    private static final String DB_USERNAME = "root";
    private static final String DB_PASSWORD = "snehak123";

    public static Connection getConnection() throws SQLException {
        Connection connection = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(JDBC_URL, DB_USERNAME,
DB_PASSWORD);
        } catch (ClassNotFoundException e) {
            throw new SQLException("MySQL Driver not found!", e);
        }
        return connection;
    }  }
```

LogoutServlet.java

```
@WebServlet("/LogoutServlet")
public class LogoutServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession(false);
        if (session != null) {
            session.invalidate(); }
  response.sendRedirect("login.jsp"); }}
```

DeletePropertyServlet.java

```
@WebServlet("/DeletePropertyServlet")
public class DeletePropertyServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        int propertyId = Integer.parseInt(request.getParameter("id"));

        try (Connection conn = DatabaseUtil.getConnection()) {
            String sql = "DELETE FROM properties WHERE id = ?";
            try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                stmt.setInt(1, propertyId);
                stmt.executeUpdate();
            }
        } catch (SQLException e) {
            e.printStackTrace();
```

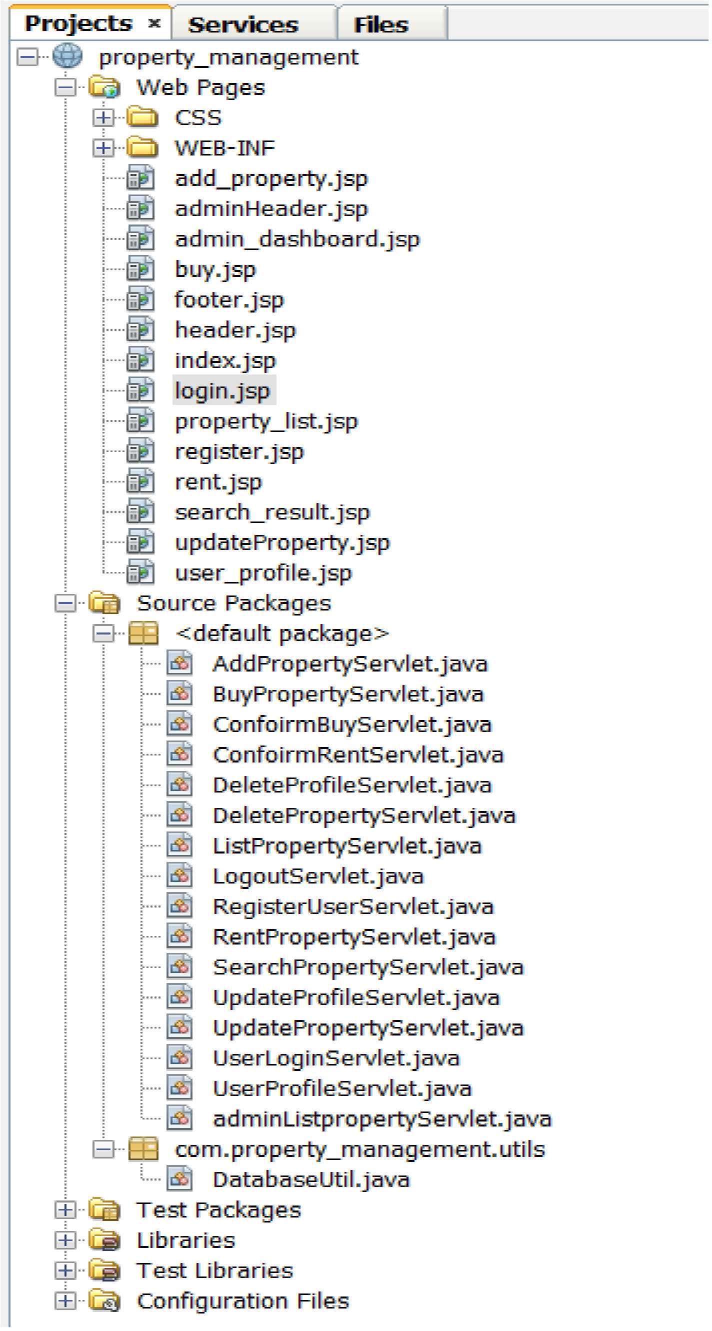
```
      response.getWriter().println("<h3>Failed to delete property: " + e.getMessage()
+ "</h3>");
      return;
    }

    response.sendRedirect("adminListpropertyServlet");
  }
}
```

buy.jsp

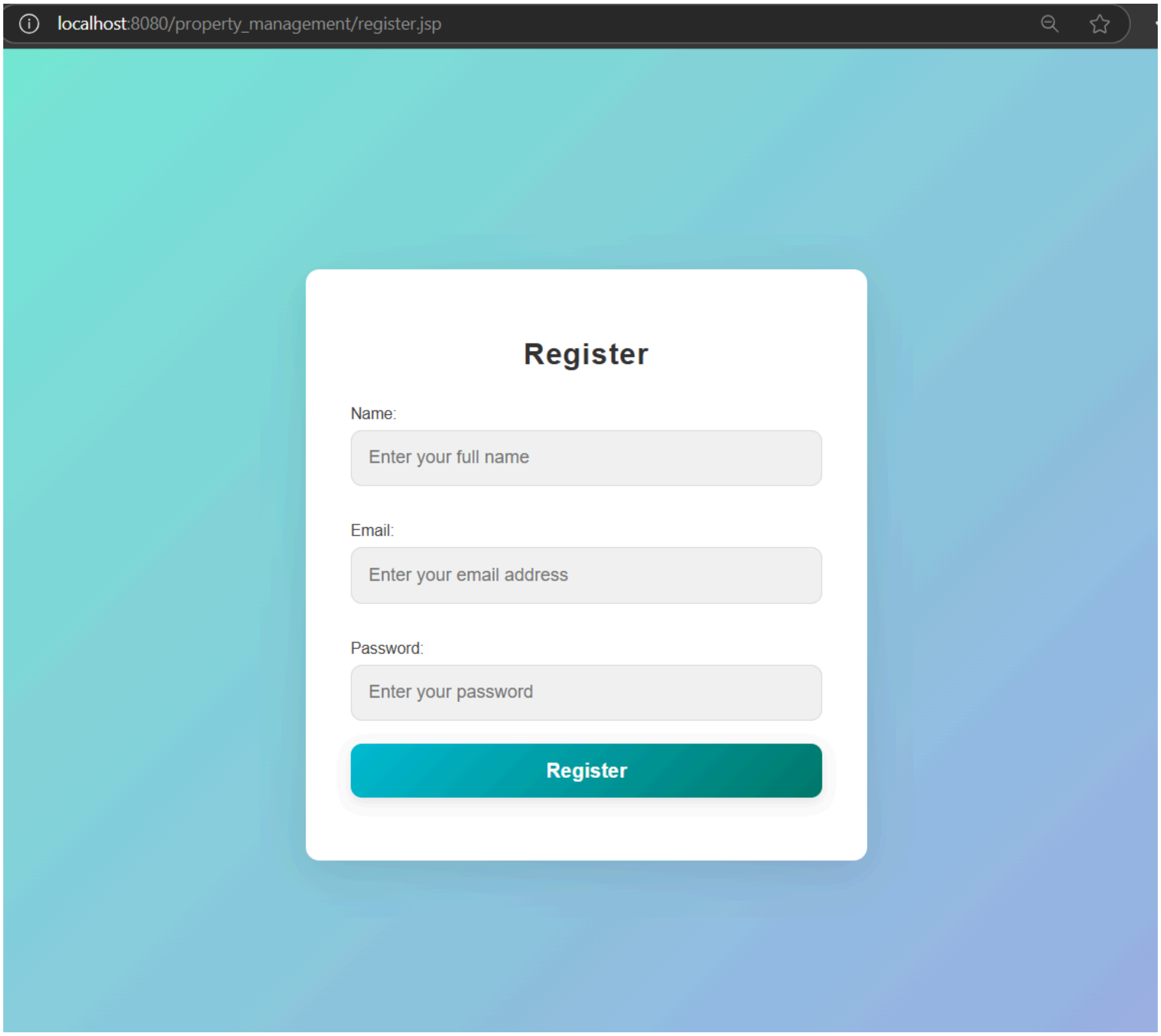
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
  <title>Buy Property</title>
  <link rel="stylesheet" type="text/css" href="CSS/buyProperty.css">

</head>
<body>

  <%
    Integer userId = (Integer) session.getAttribute("userId");
    if (userId == null) {
      response.sendRedirect("login.jsp");
    }
  %>

  <h2>Confirm Purchase</h2>
  <form action="ConfoirmBuyServlet" method="POST">
    <p><strong>Property Type:</strong> ${propertyType}</p>
    <p><strong>Location:</strong> ${location}</p>
    <p><strong>Building:</strong> ${building}</p>
    <p><strong>Price:</strong> $${price}</p>
    <input type="hidden" name="id" value="${id}">
    <button type="submit">Confirm Purchase</button>
  </form>
</body>
</html>
```

ConfoirmBuyServlet.java

```
@WebServlet("/ConfirmBuyServlet")
public class ConfoirmBuyServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        HttpSession session = request.getSession(false);
        if (session == null || session.getAttribute("userId") == null) {
            response.sendRedirect("login.jsp");
            return;
        }

        int userId = (int) session.getAttribute("userId");

        String propertyId = request.getParameter("id");

        if (propertyId != null && !propertyId.isEmpty()) {
            try (Connection conn = DatabaseUtil.getConnection()) {
                String sql = "INSERT INTO purchases (user_id, property_id) VALUES (?, ?)";
                try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                    stmt.setInt(1, userId);
                    stmt.setInt(2, Integer.parseInt(propertyId));

                    int rowsInserted = stmt.executeUpdate();
                    if (rowsInserted > 0) {
                        response.sendRedirect("user_profile.jsp");
                    } else {
                        response.getWriter().write("Purchase failed. Try again.");
                    }
                }
            } catch (Exception e) {
                log("Error processing purchase: " + e.getMessage());
                response.getWriter().write("An error occurred while processing your request.
Please try again later.");
            }
        } else {
            response.getWriter().write("Invalid property ID.");
        }
    }
}
```



```
property_management
    Web Pages
        CSS
        WEB-INF
        add_property.jsp
        adminHeader.jsp
        admin_dashboard.jsp
        buy.jsp
        footer.jsp
        header.jsp
        index.jsp
        login.jsp
        property_list.jsp
        register.jsp
        rent.jsp
        search_result.jsp
        updateProperty.jsp
        user_profile.jsp
    Source Packages
        <default package>
            AddPropertyServlet.java
            BuyPropertyServlet.java
            ConfoirmBuyServlet.java
            ConfoirmRentServlet.java
            DeleteProfileServlet.java
            DeletePropertyServlet.java
            ListPropertyServlet.java
            LogoutServlet.java
            RegisterUserServlet.java
            RentPropertyServlet.java
            SearchPropertyServlet.java
            UpdateProfileServlet.java
            UpdatePropertyServlet.java
            UserLoginServlet.java
            UserProfileServlet.java
            adminListpropertyServlet.java
        com.property_management.utils
            DatabaseUtil.java
    Test Packages
    Libraries
    Test Libraries
    Configuration Files
```



localhost:8080/property_management/register.jsp

Register

Name:

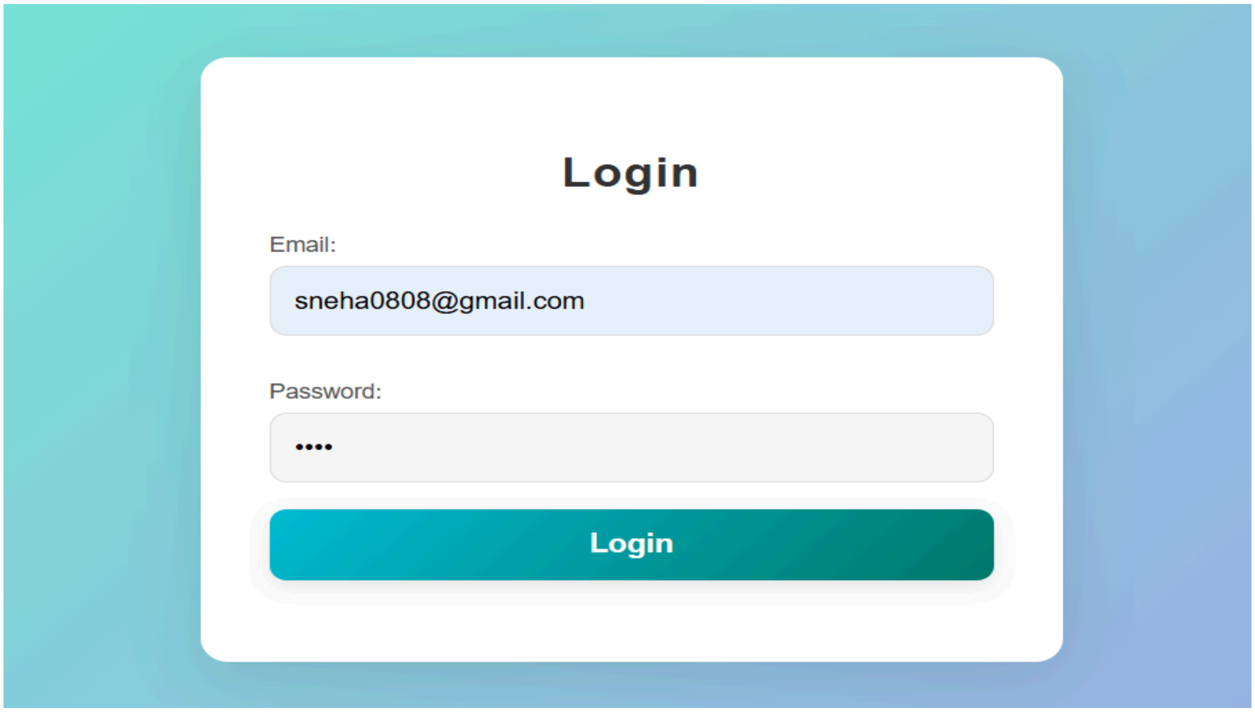
Enter your full name

Email:

Enter your email address

Password:

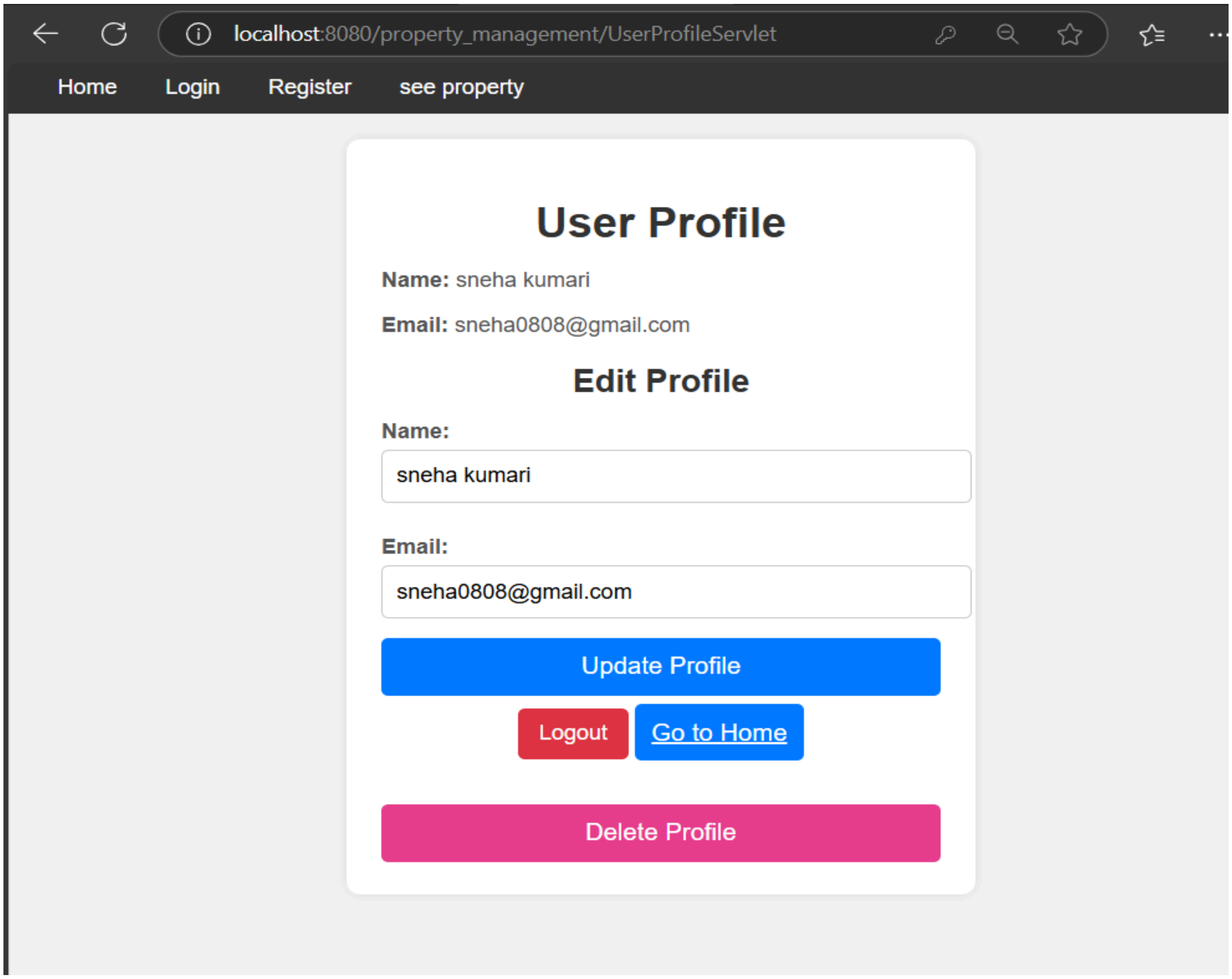
Enter your password



Login

Email: sneha0808@gmail.com

Password: ••••



localhost:8080/property_management/UserProfileServlet

User Profile

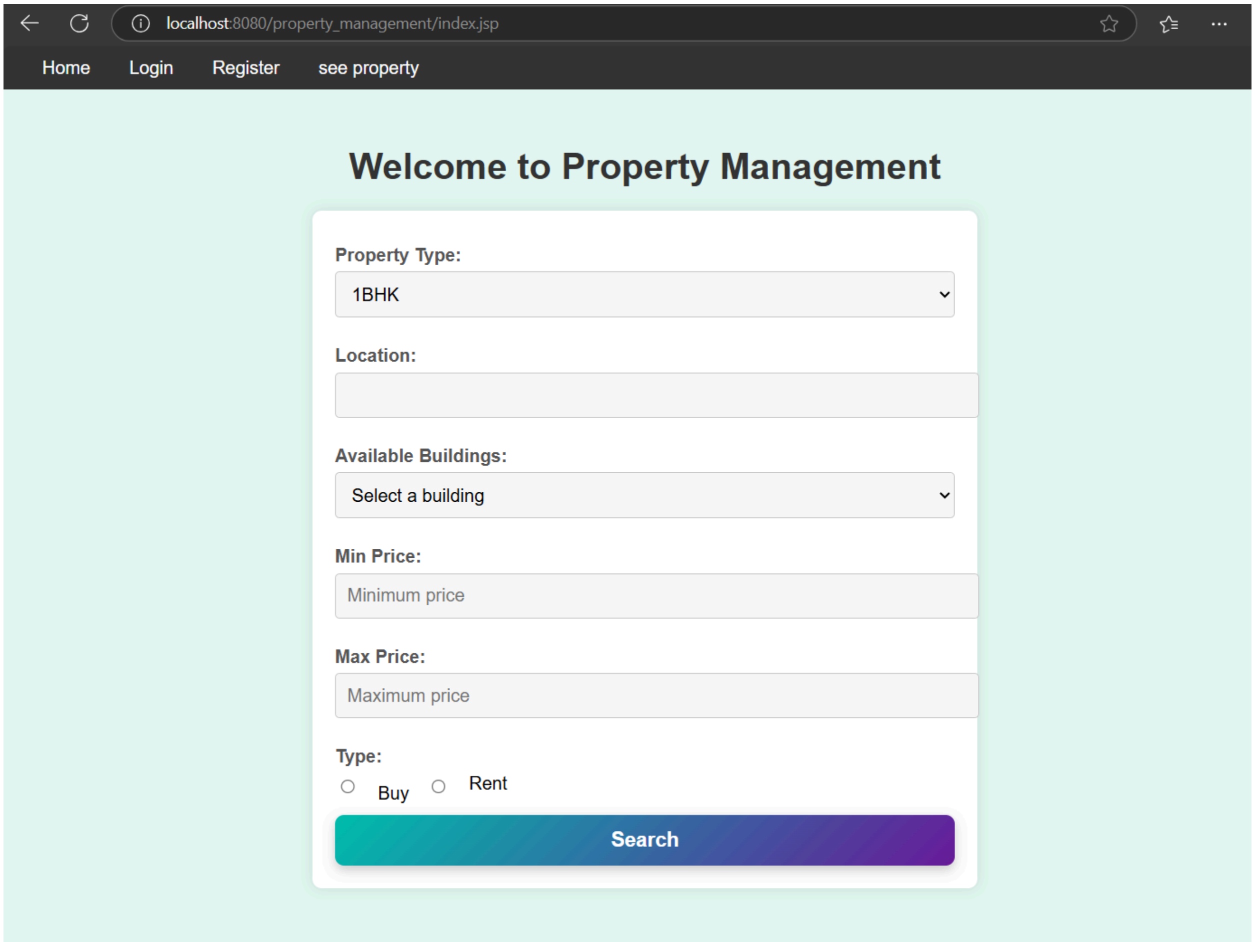
Name: sneha kumari

Email: sneha0808@gmail.com

Edit Profile

Name: sneha kumari

Email: sneha0808@gmail.com



Welcome to Property Management

Property Type:
1BHK

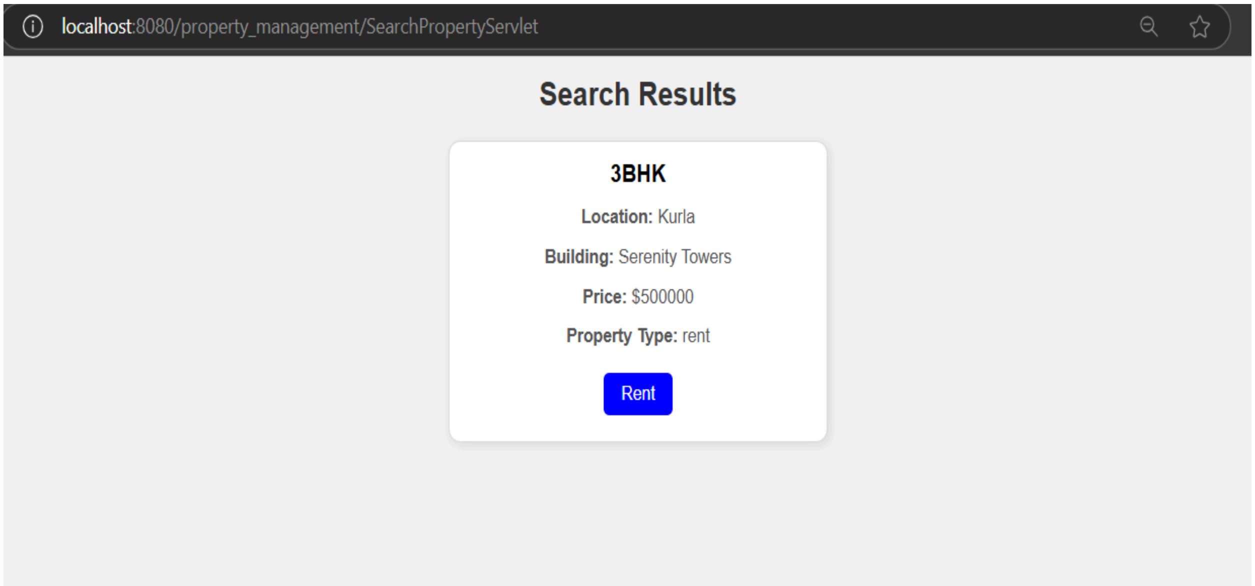
Location:

Available Buildings:
Select a building

Min Price:
Minimum price

Max Price:
Maximum price

Type:
○ Buy ○ Rent



Search Results

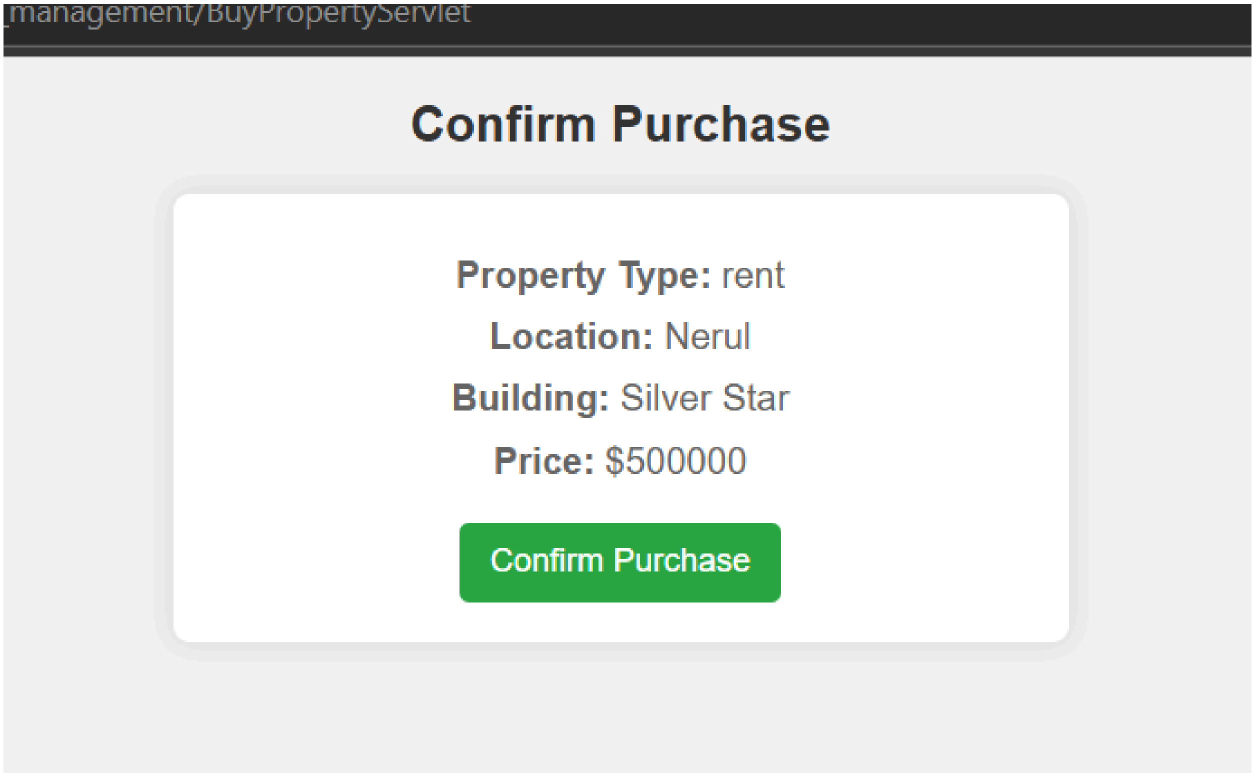
3BHK

Location: Kurla

Building: Serenity Towers

Price: $500000

Property Type: rent



_management/BuyPropertyServlet

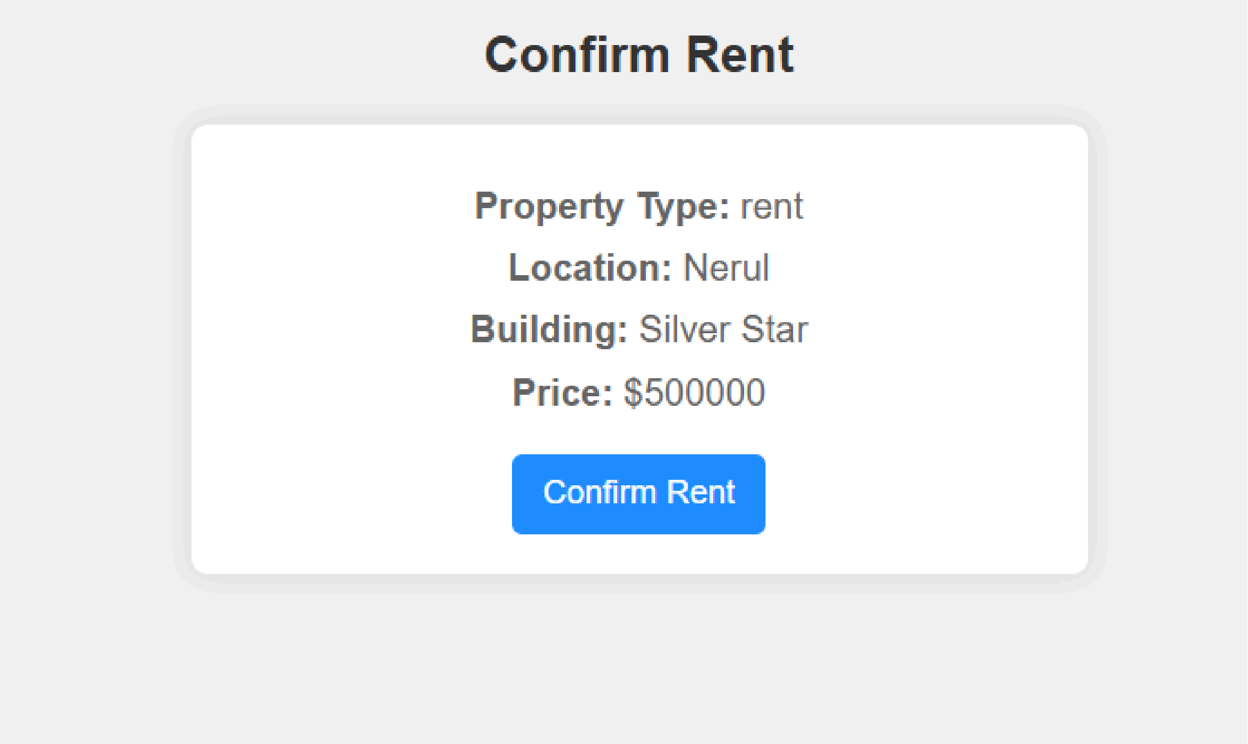
Confirm Purchase

Property Type: rent

Location: Nerul

Building: Silver Star

Price: $500000



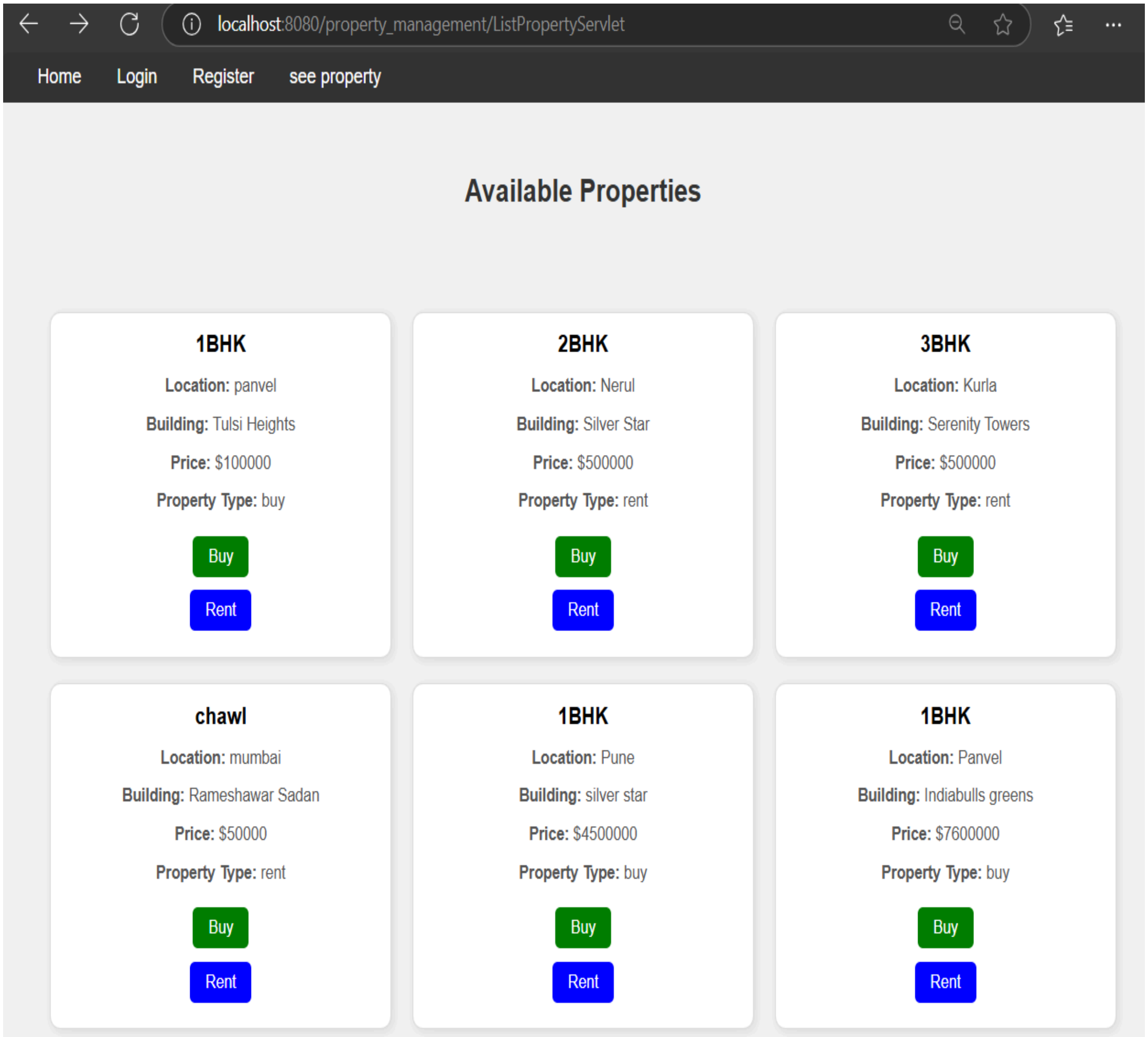
Confirm Rent

Property Type: rent

Location: Nerul

Building: Silver Star

Price: $500000



Available Properties

1BHK

Location: panvel

Building: Tulsi Heights

Price: $100000

Property Type: buy

2BHK

Location: Nerul

Building: Silver Star

Price: $500000

Property Type: rent

3BHK

Location: Kurla

Building: Serenity Towers

Price: $500000

Property Type: rent

chawl

Location: mumbai

Building: Rameshawar Sadan

Price: $50000

Property Type: rent

1BHK

Location: Pune

Building: silver star

Price: $4500000

Property Type: buy

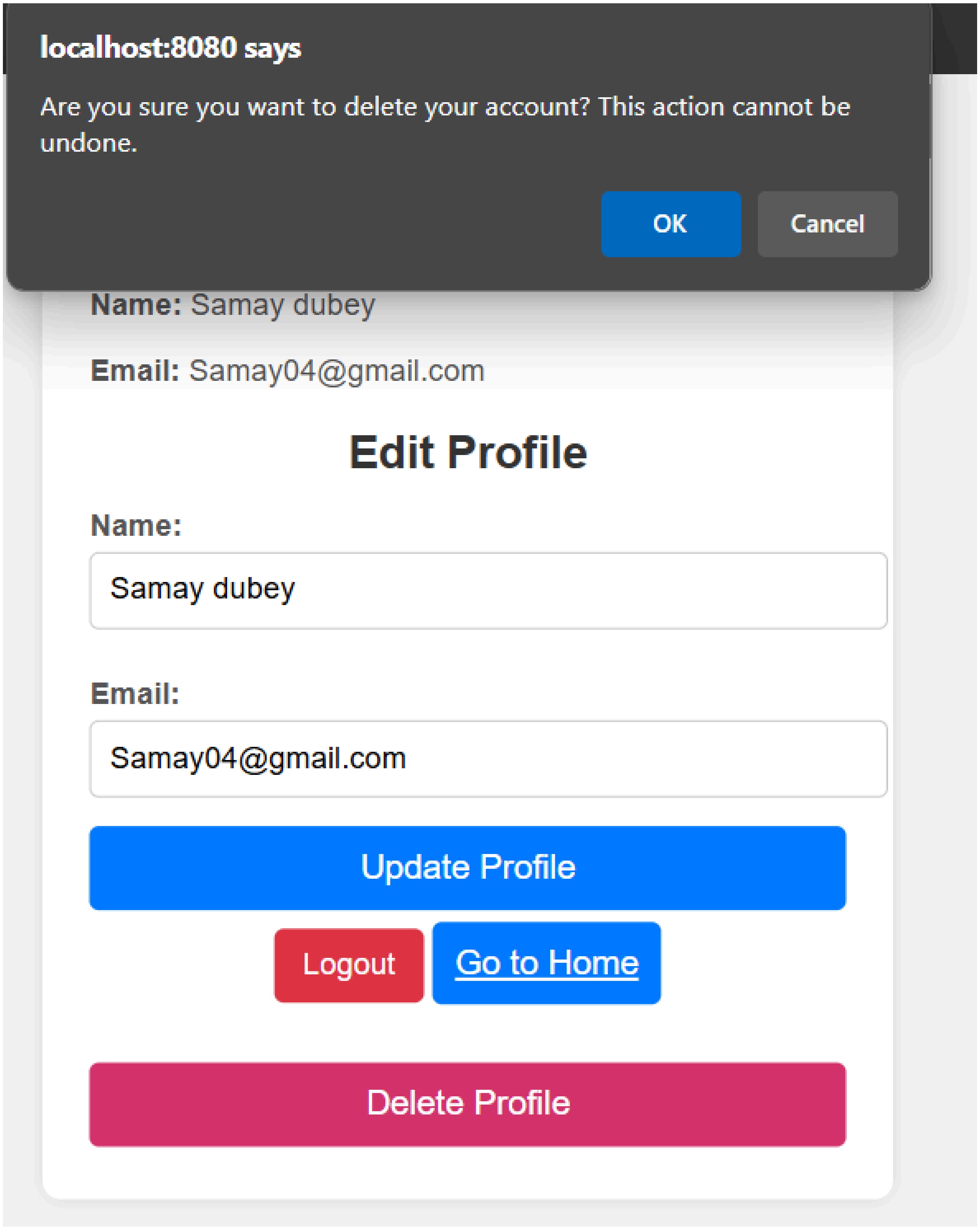
1BHK

Location: Panvel

Building: Indiabulls greens

Price: $7600000

Property Type: buy



localhost:8080 says

Are you sure you want to delete your account? This action cannot be undone.

Name: Samay dubey

Email: Samay04@gmail.com

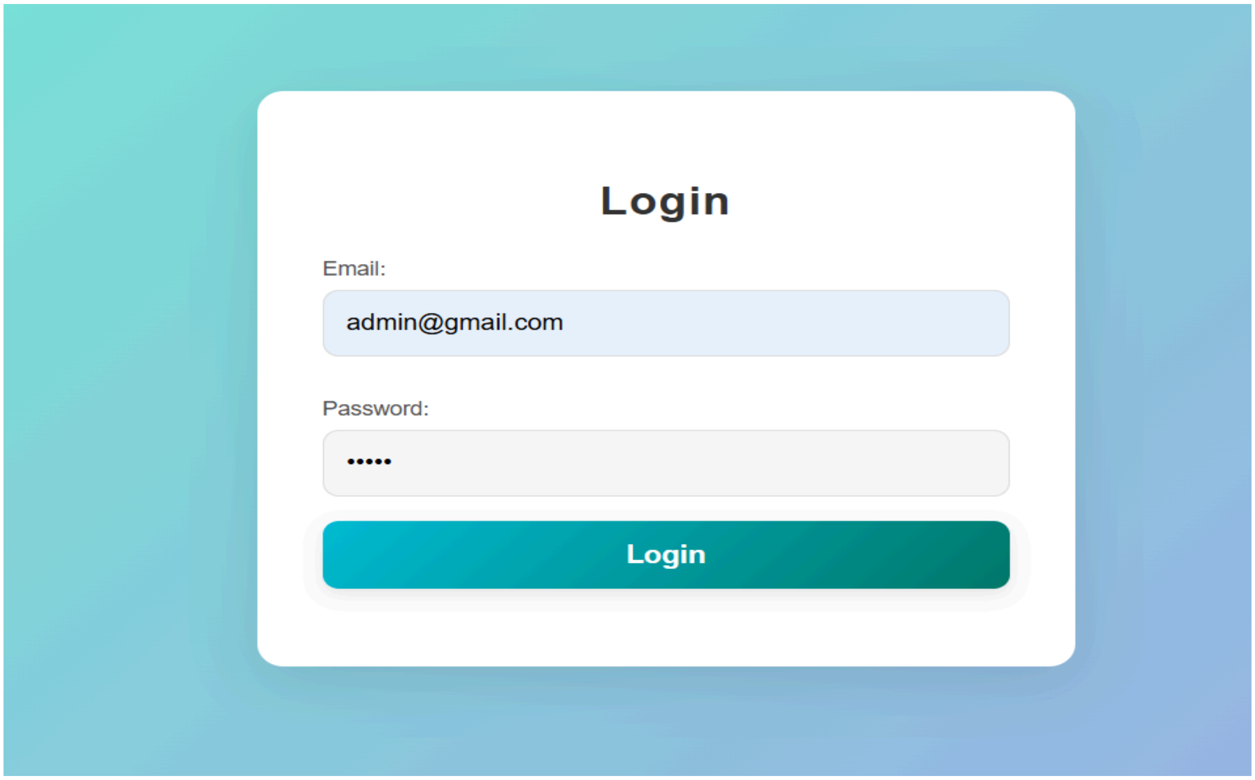
Edit Profile

Name:

Samay dubey

Email:

Samay04@gmail.com



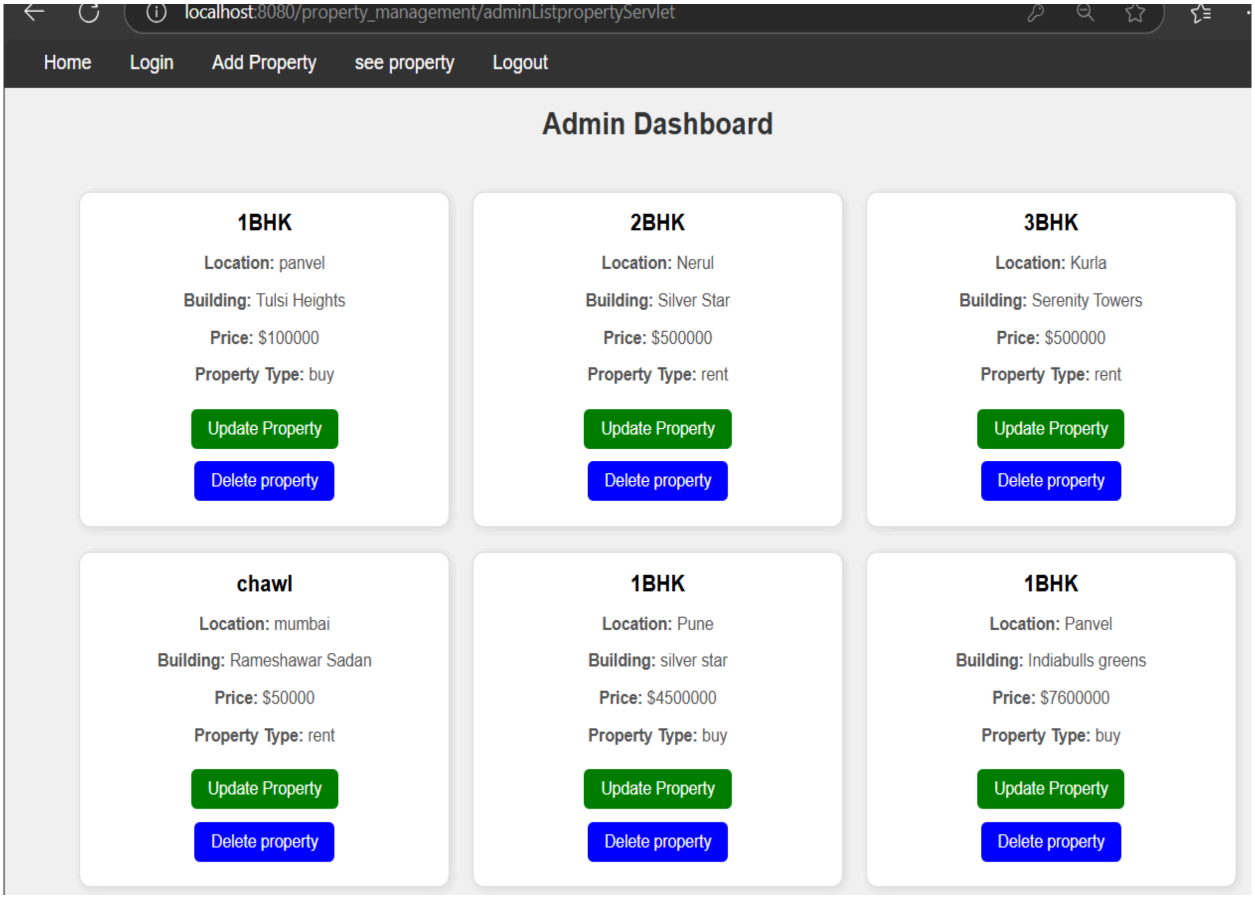
Login

Email:

admin@gmail.com

Password:

•••••



localhost:8080/property_management/adminListpropertyServlet

Admin Dashboard

1BHK

Location: panvel

Building: Tulsi Heights

Price: $100000

Property Type: buy

2BHK

Location: Nerul

Building: Silver Star

Price: $500000

Property Type: rent

3BHK

Location: Kurla

Building: Serenity Towers

Price: $500000

Property Type: rent

chawl

Location: mumbai

Building: Rameshawar Sadan

Price: $50000

Property Type: rent

1BHK

Location: Pune

Building: silver star

Price: $4500000

Property Type: buy

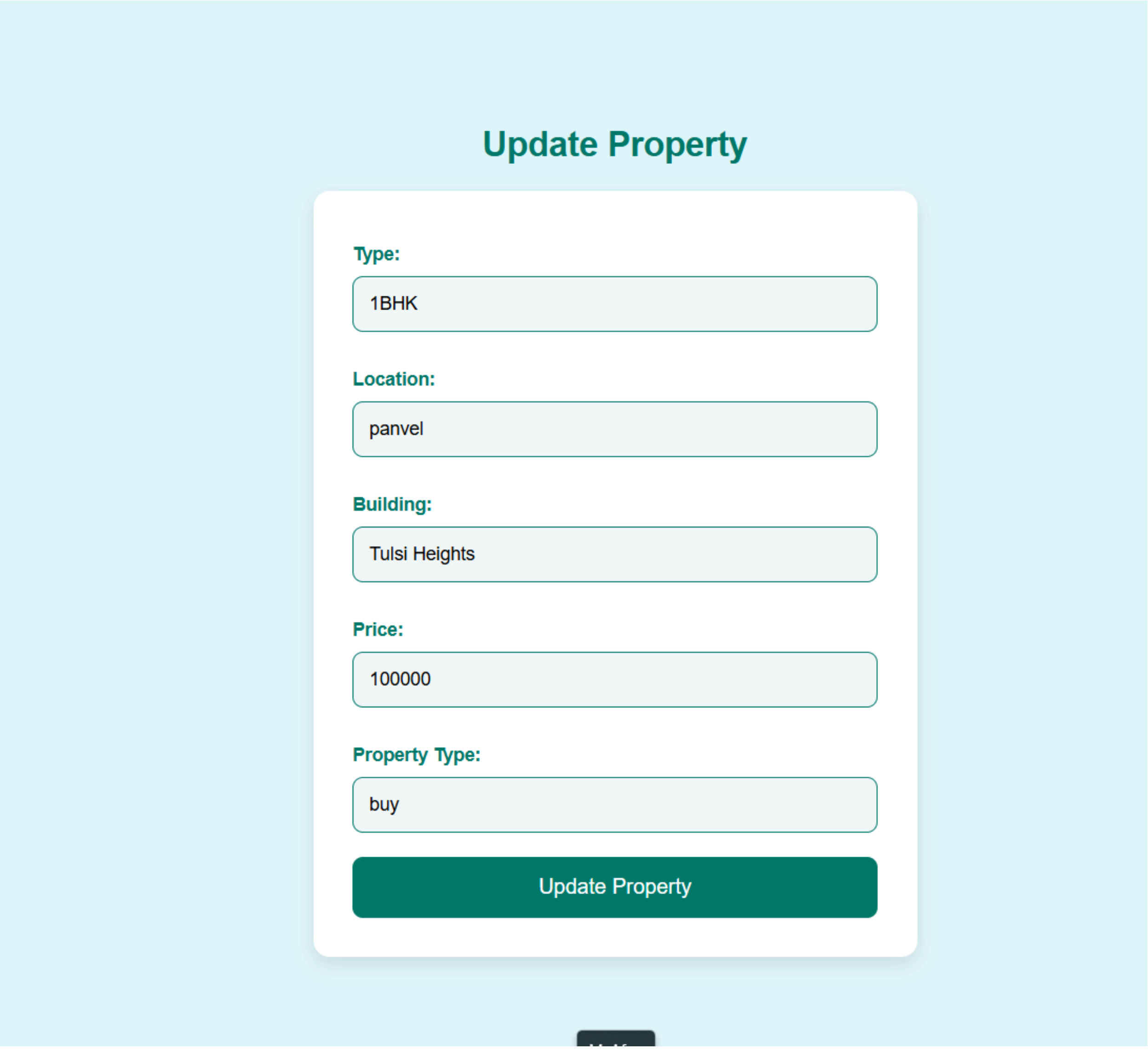
1BHK

Location: Panvel

Building: Indiabulls greens

Price: $7600000

Property Type: buy



Update Property

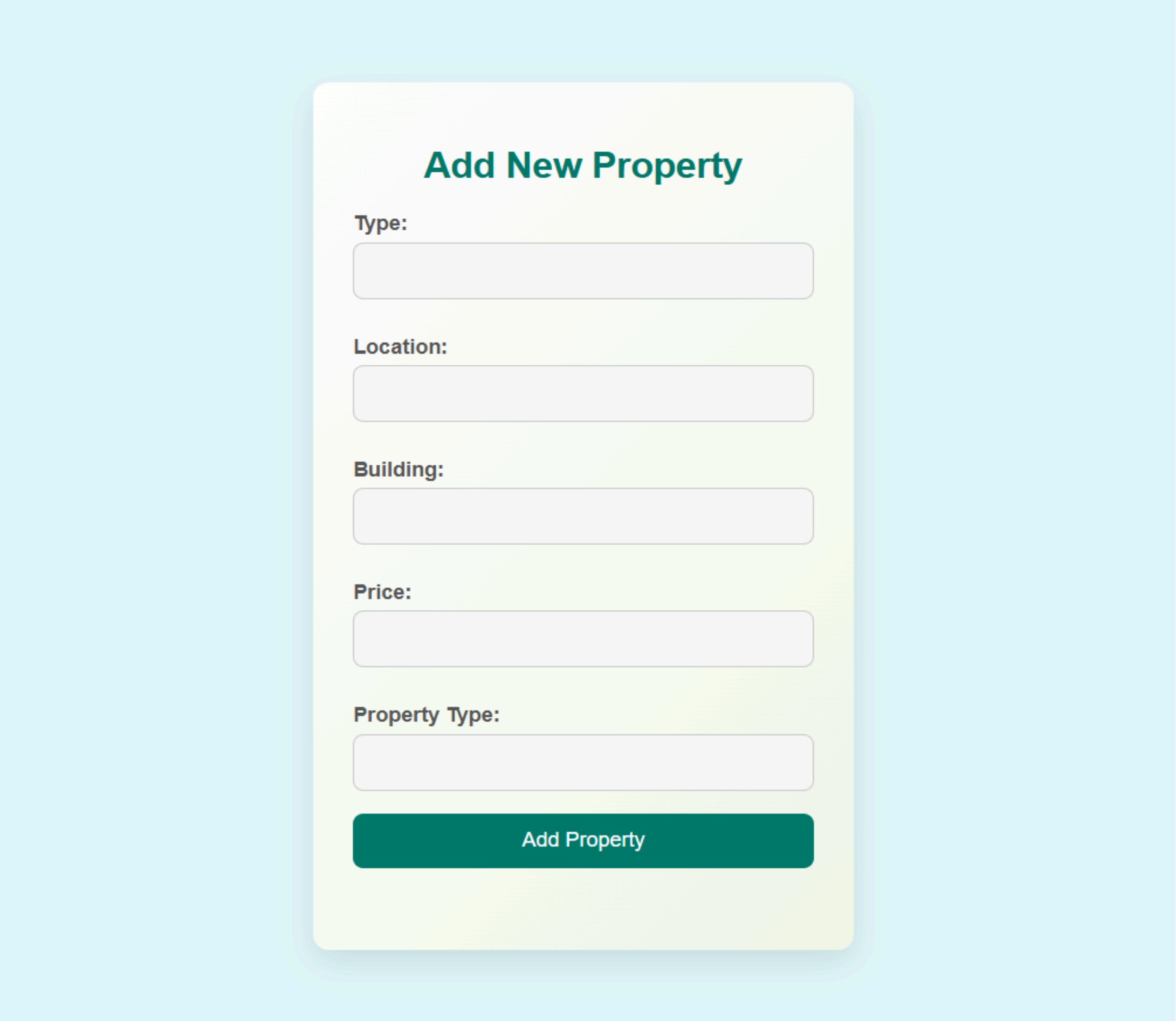
Type: 1BHK

Location: panvel

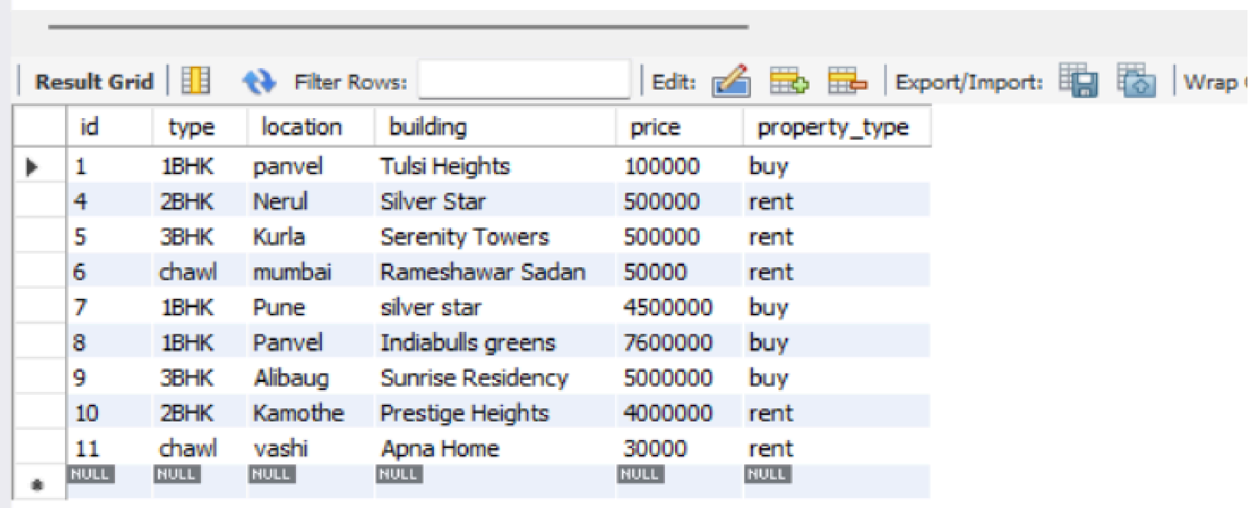
Building: Tulsi Heights

Price: 100000

Property Type: buy

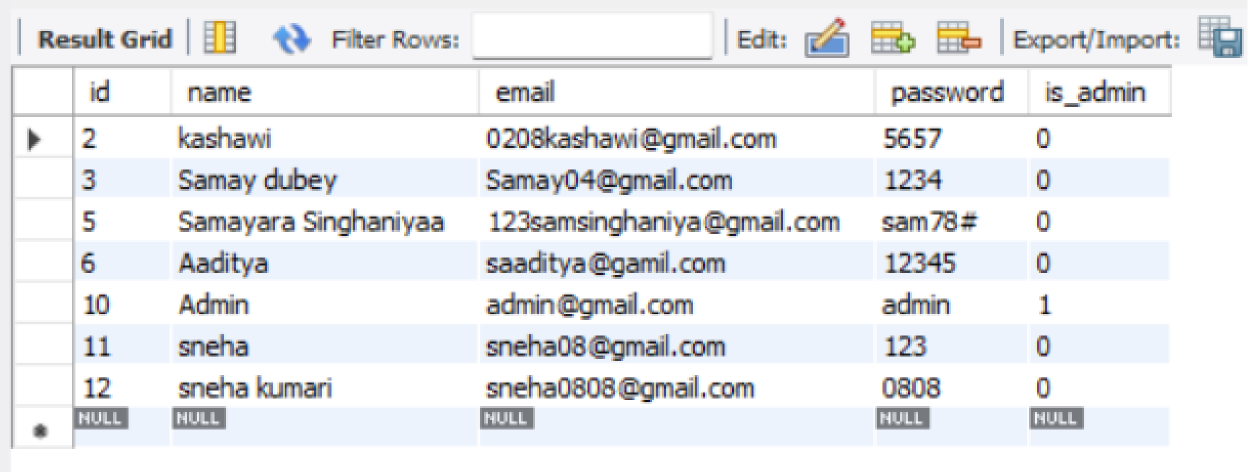


<u>properties table</u>



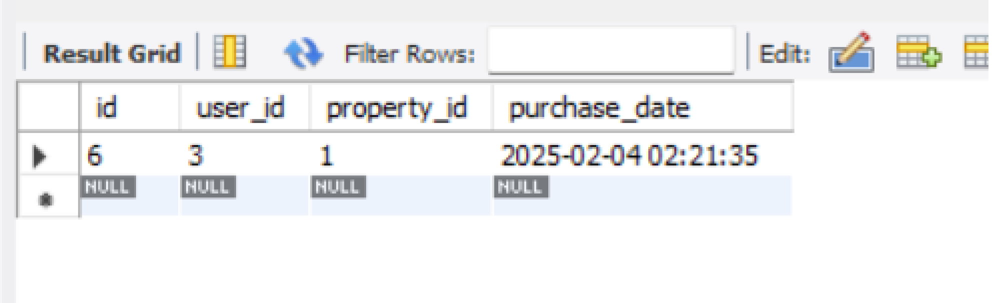
id	type	location	building	price	property_type
1	1BHK	panvel	Tulsi Heights	100000	buy
4	2BHK	Nerul	Silver Star	500000	rent
5	3BHK	Kurla	Serenity Towers	500000	rent
6	chawl	mumbai	Rameshawar Sadan	50000	rent
7	1BHK	Pune	silver star	4500000	buy
8	1BHK	Panvel	Indiabulls greens	7600000	buy
9	3BHK	Alibaug	Sunrise Residency	5000000	buy
10	2BHK	Kamothe	Prestige Heights	4000000	rent
11	chawl	vashi	Apna Home	30000	rent
NULL	NULL	NULL	NULL	NULL	NULL

users table



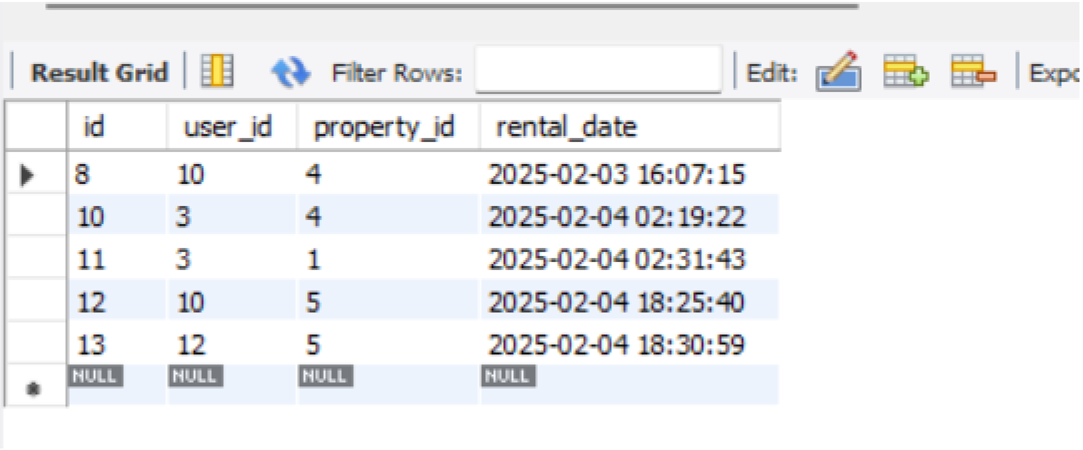
	id	name	email	password	is_admin
▶	2	kashawi	0208kashawi@gmail.com	5657	0
	3	Samay dubey	Samay04@gmail.com	1234	0
	5	Samayara Singhaniyaa	123samsinghaniya@gmail.com	sam78#	0
	6	Aaditya	saaditya@gamil.com	12345	0
	10	Admin	admin@gmail.com	admin	1
	11	sneha	sneha08@gmail.com	123	0
	12	sneha kumari	sneha0808@gmail.com	0808	0
*	NULL	NULL	NULL	NULL	NULL

Purchases table



	id	user_id	property_id	purchase_date
▶	6	3	1	2025-02-04 02:21:35
*	NULL	NULL	NULL	NULL

rentals table



	id	user_id	property_id	rental_date
▶	8	10	4	2025-02-03 16:07:15
	10	3	4	2025-02-04 02:19:22
	11	3	1	2025-02-04 02:31:43
	12	10	5	2025-02-04 18:25:40
	13	12	5	2025-02-04 18:30:59
*	NULL	NULL	NULL	NULL

SWOT ANALYSIS

STRENGTH :

1. Users can browse and book properties from anywhere without physical visits.
2. Reduces the time and effort involved in traditional property dealings.
3. Helps users find properties that meet specific requirements.

WEAKNESSES :

1. Some properties may not match online descriptions, that leads to trust issues.
3. Users with limited internet access may face challenges.

OPPORTUNITY :

1. In future different payment options can be added like online banking and card transactions.
2. More accurate representation of the property.

THREATS :

1. Establishing property rental websites pose a challenge because of competition from Existing Platforms.
2. Negative experiences of users can affect the platform's credibility.
3. Property prices and market demand can be unpredictable.

CONCLUSION :

The **Online Property Rental and Buying System** provides a solution for individuals looking to rent or buy properties conveniently. It reduces the challenges of traditional real estate transactions by offering a structured and automated platform for property listings, searches, and management.

By implementing strong security measures, user verification processes, and seamless property management features, the system ensures a trustworthy and efficient experience for both property owners and buyers. With the potential for future expansion and technological upgrades.