

# **HOUSE RENTAL SYSTEM**

## **A PROJECT REPORT**

*Submitted by*

**Roshith S (2303811724322091)**

*in partial fulfillment of requirements for the award of the course*

**CGB1201 – JAVA PROGRAMMING**

*in*

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by  
AICTE, New Delhi)

**SAMAYAPURAM – 621 112**

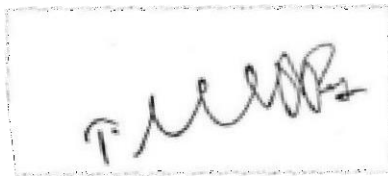
**DECEMBER, 2024**

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)**

**SAMAYAPURAM – 621 112**

**BONAFIDE CERTIFICATE**

Certified that this project report on “**HOUSE RENTAL SYSTEM**” is the bonafide work of **ROSHITH S(2303811724322091)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



**Signature**

Dr. T. AVUDAIAPPAN M.E., Ph.D.,

**HEAD OF THE DEPARTMENT,**

Department of Artificial Intelligence,  
K. Ramakrishnan College of Technology,  
Samayapuram, Trichy -621 112.



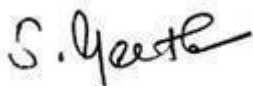
**Signature**

Mrs. S. GEETHA M.E.,

**SUPERVISOR,**

Department of Artificial Intelligence,  
K. Ramakrishnan College of Technology,  
Samayapuram, Trichy -621 112.

Submitted for the viva-voce examination held on 3.12.24



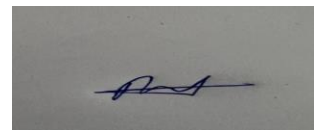
**INTERNAL EXAMINER**



**EXTERNAL EXAMINER**

## DECLARATION

I declare that the project report on “**HOUSE RENTAL SYSTEM**” is the result of original work done by me and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING**.

A rectangular box containing a handwritten signature in blue ink, which appears to be 'Roshith S'.

**ROSHITH S**

**Place:** Samayapuram

**Date:** 3/12/2024

## ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **“K. Ramakrishnan College of Technology (Autonomous)”**, for providing us with the opportunity to do this project.

I extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN, B.E.**, for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.**, Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D.**, Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E.**, Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

## **VISION OF THE INSTITUTION**

To serve the society by offering top-notch technical education on par with global standards.

## **MISSION OF THE INSTITUTION**

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

## **VISION AND MISSION OF THE DEPARTMENT**

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conducive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

## **PROGRAM EDUCATIONAL OBJECTIVES (PEOS)**

**PEO 1:** Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

**PEO 2:** Provide industry-specific solutions for the society with effective communication and ethics.

**PEO 3:** Hone their professional skills through research and lifelong learning initiatives.

## **PROGRAM OUTCOMES**

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

### **PROGRAM SPECIFIC OUTCOMES (PSOs)**

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

## **ABSTRACT**

A House Rental System in Java is a software solution designed to streamline the process of renting and managing residential properties. The system enables property owners to list available houses with details like location, rent, and amenities, while tenants can search, view, and book properties that meet their preferences. Core functionalities include user authentication, property search filters, booking management, and rental agreements. Built with Java, the system employs Object-Oriented Programming principles, ensuring modularity and scalability. It integrates with databases for secure storage of user data, property details, and transaction records, offering a seamless and efficient platform for both landlords and tenants.



## TABLE OF CONTENTS

| <b>CHAPTER No</b> | <b>TITLE</b>  | <b>PAGE No</b> |
|-------------------|---|----------------|
|                   | <b>ABSTRACT</b>                                     |                |
| <b>1</b>          | <b>INTRODUCTION</b>                                 | <b>1</b>       |
|                   | 1.1 INTRODUCTION                                    | 1              |
|                   | 1.2 OBJECTIVE                                       | 1              |
| <b>2</b>          | <b>PROJECT METHODOLOGY</b>                          | <b>2</b>       |
|                   | 2.1 PROPOSED WORK                                   | 2              |
|                   | 2.2 BLOCK DIAGRAM                                   | 4              |
| <b>3</b>          | <b>JAVA PROGRAMMING CONCEPTS</b>                    | <b>5</b>       |
|                   | 3.1 CORE FUNCTIONALITY MODULE                       | 5              |
|                   | 3.2 USER INTERACTION AND ERROR<br>MANAGEMENT MODULE | 5              |
| <b>4</b>          | <b>MODULE DESCRIPTION</b>                           | <b>6</b>       |
|                   | 4.1 PROPERTY MODULE:                                | 6              |
|                   | 4.2 LANDLORD MODULE:                                | 6              |
|                   | 4.3 TENTANT MODULE:                                 | 6              |
|                   | 4.4 ADMIN MODULE:                                   | 6              |
| <b>5</b>          | <b>CONCLUSION</b>                                   | <b>7</b>       |
|                   | <b>REFERENCES</b>                                   | <b>8</b>       |
|                   | <b>APPENDICES</b>                                   |                |
|                   | Appendix A – Source code                            | 10             |
|                   | Appendix B – Screen shots                           | 16             |

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION**

The **House Rental System** is a Java-based application that simplifies property rentals for both landlords and tenants. Landlords can manage property listings, agreements, and payments, while tenants can search for properties, apply online, and receive notifications. The system includes features like payment tracking, agreement generation, and timely alerts for seamless, transparent operations.

### **1.2 OBJECTIVE**

The objective of the House Rental System is to streamline and simplify the property rental process for both landlords and tenants. It aims to enhance efficiency, transparency, and convenience by providing a platform where landlords can manage property listings, agreements, and payments, while tenants can easily search for properties, apply online, and receive timely notifications. The system is designed to automate key processes, ensuring smooth and hassle-free transactions, and fostering trust between landlords and tenants.

## **CHAPTER 2**

### **PROJECT METHODOLOGY**

#### **2.1 PROPOSED WORK**

##### **1) User Management:**

1. Implement role-based access for landlords and tenants.
2. Secure login and registration functionality.

##### **2) Property Management:**

1. Allow landlords to add, update, and delete property listings.
2. Include property details such as location, rent, and amenities.

##### **3) Search and Filter:**

1. Enable tenants to search for properties using filters like price, location, and type.

##### **4) Rental Agreements:**

1. Provide an interface for generating and managing rental agreements.

##### **5) Payment System:**

1. Integrate payment tracking for rent collection.
2. Notify users about due dates and payment confirmations.

#### **6) Database Integration:**

1. Use a relational database to store user, property, and transaction details securely.

#### **7) User Notifications:**

1. Send alerts for new listings, payment reminders, or agreement updates.

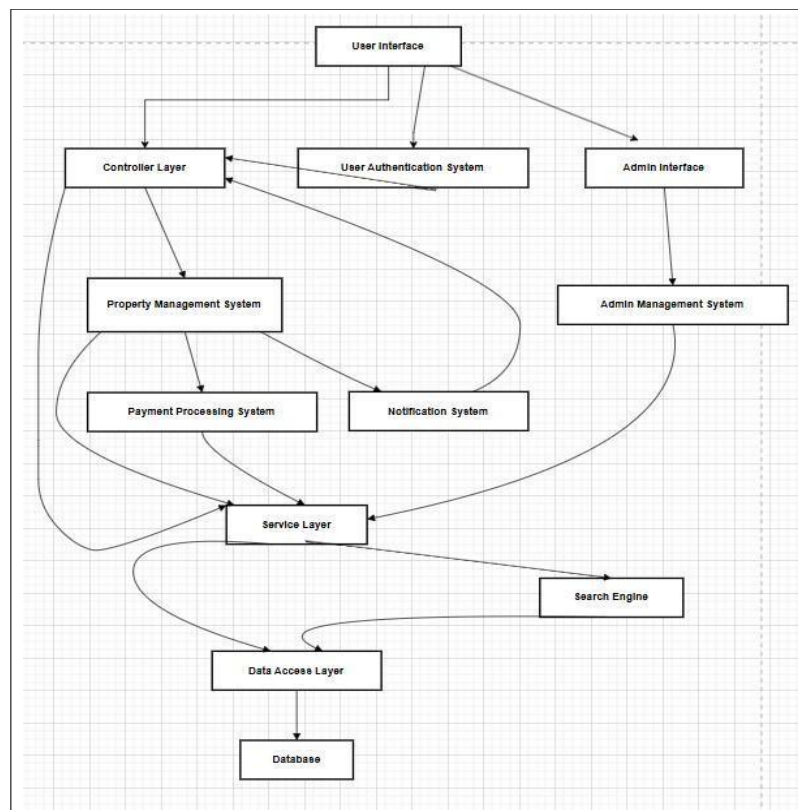
#### **8) Report Generation:**

1. Allow landlords to generate reports on rentals and earnings.

#### **9) Scalability:**

1. Plan for future integration of advanced features like AI-based property recommendations.

## 2.2 BLOCK DIAGRAM



## **CHAPTER 3**

### **JAVA PROGRAMMING CONCEPTS**

#### **3.1. CORE FUNCTIONALITY MODULE**

Object-Oriented Programming (OOP): Models real-world entities such as properties, tenants, and landlords as objects with attributes (e.g., rent, name) and behaviors (e.g., book property, pay rent).

Encapsulation: Protects sensitive data like user details and financial records by hiding internal workings and accessing them securely through methods.

Arrays/Lists: Manages collections of data such as property listings, tenant records, and payment histories for efficient storage and retrieval.

File Handling: Enables persistent storage by saving critical information (e.g., tenant details, property listings) into files, ensuring data continuity.

#### **3.2. USER INTERACTION AND ERROR MANAGEMENT MODULE**

GUI (Graphical User Interface): Provides a user-friendly interface using tools like Swing or JavaFX, allowing users to interact with the system seamlessly. Functions include viewing property details, adding new properties, and managing tenant information.

Exception Handling: Ensures smooth operation by gracefully handling runtime errors like invalid user input or missing files, thereby preventing system crashes and enhancing user experience.

## **CHAPTER 4**

### **MODULE DESCRIPTION**

#### **4.1 PROPERTY MODULE:**

The Property Module in the House Rental System enables landlords to efficiently manage property listings, including details like location, price, and availability. Tenants can search for properties based on criteria such as price and amenities. The module supports tenant applications, property maintenance requests, and automatic updates on property availability.

#### **4.2 LANDLORD MODULE:**

The Landlord Module in the House Rental System allows landlords to manage property listings, review tenant applications, and track rental payments. It provides tools to generate rental agreements, receive notifications for payments and maintenance, and access reports on rental income and property performance.

#### **4.3 TENANT MODULE:**

The Tenant Module in the House Rental System enables tenants to search for properties, submit applications online, view property details, and receive notifications for application status and payment reminders, simplifying the rental process and enhancing convenience.

#### **4.4 ADMIN MODULE:**

The Admin Module in the House Rental System enables administrators to manage user accounts, monitor property listings, and track transactions. It offers control over system operations, generates reports, manages permissions, and ensures smooth, secure platform performance and maintenance.

## **CHAPTER 5**

### **CONCLUSION**

The "House Rental System" simplifies and automates the rental process, bridging the gap between landlords and tenants. By offering features like property management, rental agreements, and payment tracking, it ensures efficiency, transparency, and ease of use. This system provides a reliable solution for modernizing property rentals.



## REFERENCES:

### Books:

- 1) *"Java: The Complete Reference"* by Herbert Schildt – for understanding core Java concepts.
- 2) *"Head First Java"* by Kathy Sierra and Bert Bates – for beginner-friendly Java programming.

### · Online Tutorials:

- 1) GeeksforGeeks - Java – Offers tutorials on Java programming and project-based learning.
- 2) TutorialsPoint - Java – Provides detailed explanations and examples for Java development.

### · GitHub Repositories:

- 1) Search for Java-based rental system projects on GitHub for sample code and project structure.
- 2) Example search: [GitHub House Rental System Java](#)

·  
·

- **Database References:**

1) "SQL For Beginners" – For understanding how to integrate databases in your project.

2) [MySQL Documentation](#) – For integrating MySQL in your system.

- **Frameworks/Tools:**

1) [Spring Framework](#) – Useful for building backend services.

2) [JavaFX](#) – For designing the user interface in desktop-based rental systems

## APPENDICES

### APPENDIX A – SOURCE CODE

```
import java.awt.*;
import
java.util.ArrayList;
import
javax.swing.*;
import javax.swing.table.DefaultTableModel;

// Main Class
public class HouseRentalSystem {

    private static ArrayList<House> houseList = new ArrayList<>(); //
    Stores house details public static void main(String[] args) {
        SwingUtilities.invokeLater() -> new LoginPage();
    }

    // Login Page
    static class LoginPage extends
    JFrame { LoginPage() {
        setTitle("Logi
        n");
        setSize(300,
        150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Layout and Components
        setLayout(new
        GridLayout(3, 2));
        JLabel lblUsername = new
        JLabel("Username:"); JTextField
        txtUsername = new JTextField();
        JLabel lblPassword = new
        JLabel("Password:");
        JPasswordField txtPassword = new
        JPasswordField(); JButton btnLogin = new
        JButton("Login");

        // Adding Components
```

```

add(lblUsername);
add(txtUsername);
add(lblPassword);
add(txtPassword);
add(new JLabel()); //
Empty cell add(btnLogin);

// Login Action
btnLogin.addActionListener
r(e -> {
    String username = txtUsername.getText();
    String password = new String(txtPassword.getPassword());
    if (username.equals("admin") &&
        password.equals("admin")) { dispose();
        new MainMenu();
    } else {
        JOptionPane.showMessageDialog(this, "Invalid credentials!");
    }
});

setVisible(true);
}
}

// Main Menu
static class MainMenu extends
JFrame { MainMenu() {
    setTitle("Main
Menu");
    setSize(300, 200);
    setDefaultCloseOperation(JFrame.EXIT_ON_C
LOSE); setLocationRelativeTo(null);

// Layout and Buttons
setLayout(new
GridLayout(3, 1));
JButton btnAddHouse = new JButton("Add
House"); JButton btnViewHouses = new
JButton("View Houses"); JButton btnExit =
new JButton("Exit");

// Adding Buttons
add(btnAddHouse
);
add(btnViewHous
es); add(btnExit);

```

```

// Button Actions
btnAddHouse.addActionListener(e -> new AddHouse());
btnViewHouses.addActionListener(e -> new ViewHouses());
btnExit.addActionListener(e -> System.exit(0));

setVisible(true);
}
}
// Add House Page
static class AddHouse extends
JFrame { AddHouse() {
    setTitle("Add
    House");
    setSize(400, 250);
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setLocationRelativeTo(null);

// Layout and Components
setLayout(new
GridLayout(4, 2));
JLabel lblAddress = new
JLabel("Address:"); JTextField
txtAddress = new JTextField();
JLabel lblRent = new
JLabel("Rent:"); JTextField txtRent
= new JTextField();
JLabel lblOwner = new
JLabel("Owner Name:"); JTextField
txtOwner = new JTextField(); JButton
btnAdd = new JButton("Add House");

// Adding Components
add(lblAddress);
add(txtAddress);

```

```

add(lblRent);
add(txtRent);
add(lblOwner);
add(txtOwner);
add(new JLabel()); // Empty cell
add(btnAdd);
// Add House Action
btnAdd.addActionListener(e -> {
String address = txtAddress.getText();
String rent = txtRent.getText();
String owner = txtOwner.getText();

if (address.isEmpty() || rent.isEmpty() || owner.isEmpty()) {
JOptionPane.showMessageDialog(this, "Please fill all fields!");
} else {
houseList.add(new House(address, rent, owner));
JOptionPane.showMessageDialog(this, "House added successfully!"); dispose();
}
});

setVisible(true);
}
}

// View Houses Page with JTable
static class ViewHouses extends
JFrame { ViewHouses() {
setTitle("V
iew Houses");
setSize(500,
300);
setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
setLocationRelativeTo(null);

// Table Model and JTable
String[] columnNames = {"Address", "Rent",
"Owner"}; DefaultTableModel model = new
DefaultTableModel(columnNames, 0); JTable table =
new JTable(model);

// Populate Table with House Data
for (House house : houseList) {

```

```

        model.addRow(new Object[]{house.getAddress(), house.getRent(),
house.getOwner()});
    }

    // Scroll Pane for Table
    JScrollPane scrollPane = new
    JScrollPane(table); add(scrollPane);

    setVisible(true);
    }
    }

    // House Class for Data
    Storage static class
    House {
        private String
        address; private
        String rent;
        private String
        owner;

        public House(String address, String rent,
            String owner) {

            this.address = address;

            this.rent = rent;

            this.owner = owner;
        }

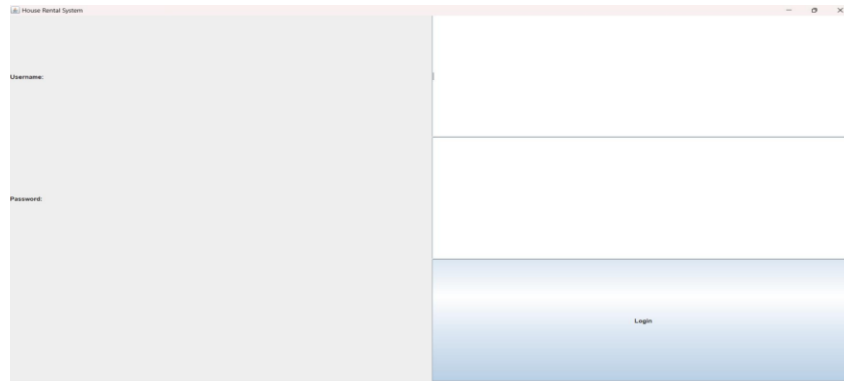
        public String
        getAddress() {
            return address;
        }
        public String
        getRent() {
            return rent;
        }

        public String
        getOwner() {
            return owner;
        }
    }
}

```

## APPENDIX B - SCREENSHOTS

### Login



Login with username and password.

### Creating Module:



Adding new house for rent or visiting their details, viewing house details.

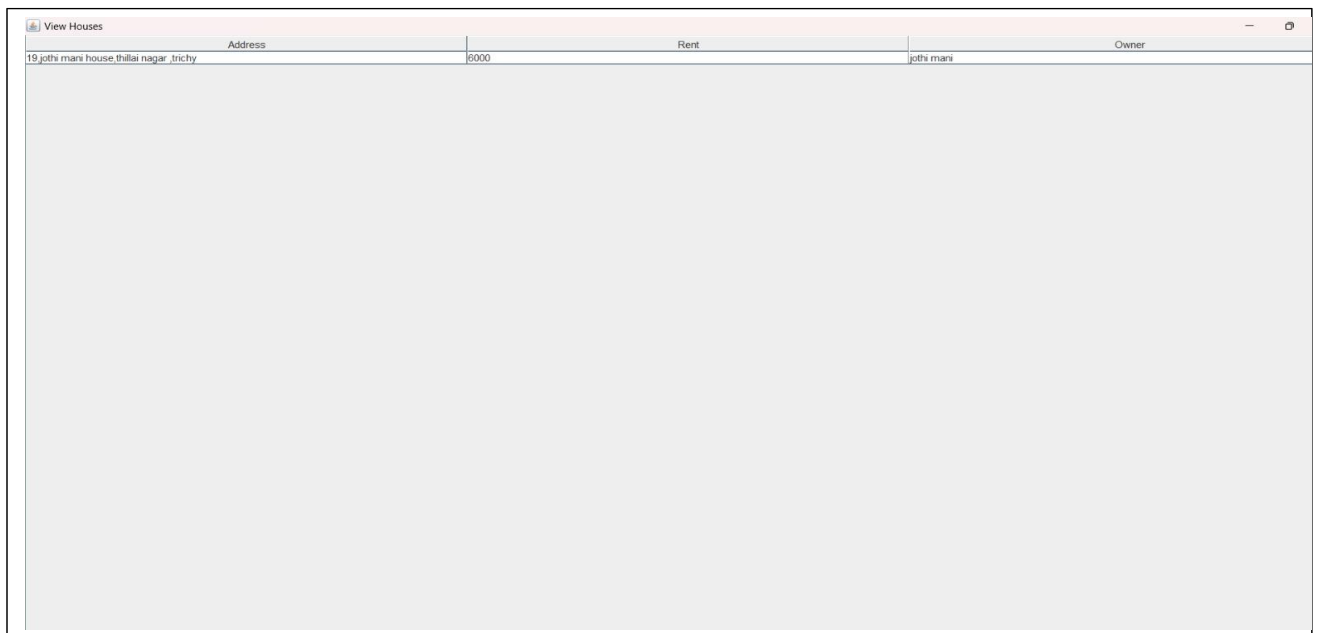


## Details Module:



A screenshot of a web application window titled "House Rental System". The window is divided into two main sections. The left section is a light gray area containing three labels: "Address:", "Rent:", and "Owner Name:". The right section is a white area with three empty input fields stacked vertically. At the bottom right of the white area is a blue button labeled "Save".

## Adding house details



A screenshot of a web application window titled "View Houses". The window displays a table with the following data:

| Address                                   | Rent | Owner      |
|---|------|------------|
| 19,jothi mani house,thillai nagar ,trichy | 9000 | jothi mani |

The table is displayed in a light gray area. The window title is "View Houses".