**Intelligent Homes: Development of artificial intelligence-based homes automations using IOT and AI bots.**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE AWARD OF DEGREE OF

#### BACHELOR OF TECHNOLOGY IN

**COMPUTER SCIENCE**

****

Submitted by

**Sumit Raj (2100290120167)**

Supervised by Mr. Vinay Pratap Assistant Professor

**Session 2024-25 DEPARTMENT OF COMPUTER SCIENCE**

**KIET GROUP OF INSTITUTIONS, GHAZIABAD**

**(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India)**

### DECLARATION

We hereby affirm that this submission is our own work and that, to the best of our knowledge and belief, it incorporates no material previously published or written by any other person and no material which to a significant extent has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where acknowledgement has been given in the text.

Name: Sumit Raj

Roll No: 2100290120167

Date:

Signature:

# CERTIFICATE

This is to authenticate that Project Report titled " **Intelligent Homes: Development of artificial intelligence-based homes automations using IOT and AI bots.**" submitted by Sumit Raj in partial satisfaction of the requirements for the award of B.Tech degree in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work undertaken by them under my guidance. The content contained in this report is original and not submitted for the award of any other degree.

**Date:**

**Supervisor**: Mr. Vinay Pratap Singh (Assistant Professor)

# ACKNOWLEDGEMENT

It provides us immense pleasure to submit the report of the B. Tech Project done during B.Tech Final Year. We are in special debt of gratitude to Assistant Professor Mr. Vinay Pratap Singh, Computer Science Department, KIET, Ghaziabad, for his continuous support and guidance during the span of our efforts. Her sincerity, thoroughness and persistence have been a source of constant inspiration to us. It is solely his aware endeavors that our efforts have gained light of the day.

We also make a point to recognize the contribution of Dr. Ajay Kr. Shrivastava, Department Head of Computer Science, KIET, Ghaziabad, for their complete support and cooperation while creating the project. We also don't want to miss the point to recognize the contribution of all the faculty members of the department for their friendly assistance and support while creating our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Sumit Raj (2100290120167) Signature:

# ABSTRACT

The expansion of smart technology has led the way to the evolution of the traditional houses to intelligent living settings. This project under the heading of Intelligent Homes introduces a holistic home automation system that combines Internet of Things (IoT) with artificial intelligence in order to make a responsive, efficient-energy, and person-oriented living space. The main goal is to create a system that can keep track of and manage household appliances, have an ability to learn about and adjust to the behavior of users.

Our approach involves a number of sensors, such as temperature, motion, and sound sensors, to collect environmental data and make necessary responses. By analyzing the trends the system proposes personalized ways of automation based on usage patterns and increases comfort and saves energy waste.

Apart from basic automation, the system is capable of addressing anomaly detection and dynamic change of an environment. Performance of the project was tested under various situations to determine its accuracy, speed of operation and adaptability by the user. The results show that the intelligent model can improve the home efficiency and satisfaction of users with a little input from hands.

Through this IoT-infused machine learning-augmented project, we demonstrate how this combination can turn spaces to be smarter or more intuitive to live in. It is laying down the basis for further innovative smart housing developments, providing a scalable and flexible solution that would fit the increasingly automation-oriented and energy-wise design demand.

|  |
| --- |
|  |
| DECLARATION……………………………………………………………………… 2 |
| CERTIFICATE………………………………………………………………………... 3  ACKNOWLEDGEMENTS… 4 |
| ABSTRACT……………………………………………………………………………… |
| LIST OF FIGURES…………………………………………………………………................. 9 |
| LIST OF TABLES 9 |
| LIST OF ABBREVIATIONS……………………………………………………………………… 9 |
|  |
| CHAPTER 1 INTRODUCTION………………………………………………………………..  10 |
| 1.1 INTRODUCTION |
| 1.2 Project Description |
| 1.3 Project Category |
| 1.4 Advantages |
| CHAPTER 2 LITERATURE REVIEW……………………………………………………….. 14 |
| 2.1 Introduction |
| 2.2 Project Description |
| * 2.2.1. Manual Home Appliance Control |
| * 2.2.2 Sensor-Based Automation Systems |
| * 2.2.3 IoT-Based Smart Home Models |
| 2.3. Common Features in Existing Systems |
| 2.4. Identified Gaps in Literature and Practice |
| 2.5. Justification for the Proposed System |
| 2.6 Conclusion |
| CHAPTER 3 PROPOSED SYSTEM……………………………………………………. 17 |
| 3.1. Development Methodology |
| * 3.1.1 Requirement Analysis |
| * 3.1.2 System Design |
| * 3.1.3 Implementation |
| * 3.1.4 Testing |
| * 3.1.5 Deployment and Maintenance |

**TABLE OF CONTENTS**

|  |
| --- |
| 3.2. System Architecture |
| * 3.3. Modules and Workflow |
| 3.3.1 User Module |
| 3.3.2 Admin Module |
| * 3.3.3 Automation Engine Module |
| 3.4. Data Flow Diagram (DFD) |
| 3.5. Security Measures |
| 3.6. Tools and Technologies Used |
| 3.7. Advantages of the Proposed Methodology |
| 3.8. Conclusion |
|  |
|  |
| CHAPTER 4 REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION…………………….. 20 |
| 4.1. Technical Feasibility |
| 4.2. Operational Feasibility |
| 4.3. Economic Feasibility |
| 4.2 Software Requirements specification(SRS) |
| * 4.2.1 Data Requirements |
| * 4.2.2 Functional Requirements |
| * 4.2.3 Performance Requirements |
| * 4.2.4 Maintainability Requirements |
| * 4.2.5 Security Requirements |
| CHAPTER 5 IMPLEMENTATION……………………………………………..  27 |
| 5.1. Technologies Used |
| 5.2. Steps for Implementation |
| * 5.2.1: Set Up the Development Environment |
| * 5.2.2: Backend Development |
| * 5.2.3: Database Setup |
| * 5.2.4: Frontend Development |
| * 5.2.5: AI Integration |
| * 5.2.6: IoT Device Integration |
| * 5.2.7: Maintenance |
| CHAPTER 6 TESTING, AND MAINTENANCE  …………………………………………………………. 30 |
| 6.1Testing |

|  |
| --- |
| 6.1.1Unit Testing |
| 6.1.2Integration Testing |
| 6.1.3 System Testing |
| 6.1.4 User Acceptance Testing (UAT) |
| 6.1.4 Performance Testing Security Testing |
| 6.2. Maintenance |
| 6.2.1 Bug Fixes and Updates Database Maintenance |
| 6.2.2 User Feedback and Improvements Security Patches and Updates |
| 6.2.3 Scalability Testing User Support |
| CHAPTER 7 RESULTS AND DISCUSSIONS ……………………………………………………… 33 |
| 7.1Functional Results |
| 7.1.1 User Module |
| 7.1.2 Device control Module |
| 7.1.3 Admin Module |
| 7.2. Performance Evaluation |
| 7.2.1. User Feedback |
| 7,2.2. Key Observationsf |
| 7.2.3. Limitations Identified |
|  |
| CHAPTER 8 CONCLUSION AND FUTURE SCOPE  ………………………………………………… 38 |
| 8.1 Conclusion |
| 8.2 Future Scope |
| REFERENCES  ……………………………………………………………………………………………… 40 |
|  |
|  |
|  |
|  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| 7.1 | Login Page | 33 |
| 7.2 | Device Control Module | 34 |
| 7.3 | Admin Control Module | 34 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Description** | **Page No.** |
| 1.1 | Tools and technology | 18 |

### LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviations** | **Full forms** |
| IH | Intelligent Homes |
|  |  |
| DBMS | Database Management System |
| UI | User Interface |
| UX | User Experience |

**Chapter 1 Introduction**

### INTRODUCTION

In today’s fast-changing world, there is a growing need for homes that can keep up with the daily needs of people. Most traditional homes still depend on manual work, which often leads to wasted energy and safety problems. But with the fast progress in smart devices (IoT) and artificial intelligence (AI), it's now possible to completely transform how our homes work.

This research introduces the idea of an "Intelligent Home"—a smart system that uses sensors and real-time data to automate daily tasks and make life easier. This technology allows people to control their home appliances from anywhere, making the home safer and more energy-efficient. The system also learns from the way people live and adjusts things like lighting, temperature, and appliance use to suit their habits.

A key part of this system is its advanced security. With features like face and voice recognition, the home becomes more protected. If anything unusual happens, like a break-in, the system can send alerts instantly, so people can take action even if they’re not at home. This gives users peace of mind and helps them focus on their work or other activities.

More than just comfort and safety, the system also helps save money and protect the environment. By studying the data from sensors, it can spot which devices are using too much energy and suggest ways to improve. This helps people make better choices, lower their electricity bills, and reduce their impact on the planet.

## Project Description

The Intelligent Homes project is a smart home automation system that is used to monitor and control household appliances through an easy-to use web dashboard. The system was designed with the MERN stack for the frontend interface, in which real-time monitoring of the environment is possible to track factors such as temperature, humidity, and sound with the help of sensors. The backend is done using Node.js and MongoDB, which guarantees secure data storage and communication. Although the system seems to make intelligent choices according to the behavior of a user, a certain degree of imitation of AI-based learning has been subtly implemented. The project is intended to improve the sense of comfort, energy efficiency, and remote control over the home by means of an easy-to-use and sensitive interface.

**Purpose of the System**

The primary aim of this system is to offer smart, efficient, and user-friendly solution for home automation and monitoring. It aims to:

* Enable remote controlling and monitoring of home appliances via a responsive web dashboard to users.
* Allow updates in real-time regarding the environmental conditions such as temperature, humidity and sound with the help of sensory data.
* Offer a centralized device state management system and patterns of usage monitoring to enhance energy efficiency.
* Provide a seamless interaction of users with the system by creating a clean interface and backend logic to follow the user behavior.
* Smooth up home management by providing insights and partial automation for daily chores.

Main Features and Functional Modules

#### User Dashboard Module

* + Real-time view of the connected devices and sensors.
  + Show environmental information, including such as: temperature, humidity, sound-levels.
  + Control the appliances (for example, lights, fans, or alarms) using a web-based interface.
  + Be alerted or be notified of abnormal conditions.

#### Device and Sensor Management Module

* + Install or deploy or delete IoT enabled appliances.
  + Monitor sensor status and connectivity status.
  + Keep usage of device logs and environment changes.
  + Enable or disable devices remotely.

#### Automation Logic and Suggestions Module

* + Study user behaviour and provide routines for automation.
  + Provide rule-based controls (such as turn fan, if temperature > 30°C).
  + Automatically schedule the appliances depending on time or conditions.
  + Log preferences of the user for better future suggestions.

#### Admin/Control Panel Module

* + Admin dash board to observe system health as a whole.
  + Add or configure user and devices from a panel.
  + View system logs, error reports, as well as data on performance.
  + Allow setting permission for various user types (e.g., members, admin).

#### Alert and Notification System

* + Provide alerting from the dashboard of out of normal events (noise spikes, high temperature).
  + Alert users when the appliances are on for too long.
  + Send maintenance reminders of sensors and devices.
  + Store old notifications for user’s perusal.

#### Reporting and Analytics Module

* + Produce reports of usage for the appliances and energy consumption.
  + Detect the historical trends for the environment data.
  + Export logs, reports in PDF or CSV.
  + Supply insights for users to optimize consumption of energy at home.

#### Technology Stack Used

* + **Frontend:** HTML5, CSS3, JavaScript, React.js, Tailwind CSS.
  + **Backend:** Node.JS, Express.JS, Docker.
  + **Database:** MongoDB with Mongoose.
  + **Platform:** Web-based interface accessible via browser on local network or hosted server.
  + **Dashboard Development**: Built using the MERN (MongoDB, Express, React, Node) stack for efficient device monitoring and control.

#### Advantages of the System

* + **Remote Monitoring:** Users can monitor and control home appliances wherever they are through the use of web dashboard.
  + **Energy Efficiency:** The system assists in minimizing consumption of power since it automates the usage of devices depending on current conditions..
  + **User-Friendly Dashboard:** The web interface is easy to use, intuitive, and quick for any user.
  + **Customization and Automation:** Users can make automation rules and choices for controlling appliances depending on requirements.
  + **Real-Time Alerts:** If some abnormal temperature, sound or activity is suspected, immediate notifications are made.
  + **Centralized Control:** It is possible to manage all the appliances and sensors using a single platform thus eliminating manual control.
  + **Scalable System:** New device and feature can be added without impacting on the current structure.

#### Real-World Use Case Example

Imagine a home owner accessing the dashboard of Intelligent Homes using a browser. The system has the live reading of temperature and sound coming from other rooms. According to the settings that have been defined by the users, the fan in the living room switches on automatically if the temperature gets past a threshold. At the same time, if the sound sensor is recording some unusual noise during the night hours then the system sends an alert notification to the user. The dashboard contains all the statuses of appliances, sensor data, and usage logs. The homeowner can also switch lights and appliances off using a manual mode of remote control for the benefit of convenience and cost of electricity savings.

#### Conclusion

Intelligent Homes system is a stable centralized solution for the modern home automation through the use of the web-based dashboard. It helps promote convenience, energy-efficiency, and better control through real-time monitoring and smart automation. The system was entirely created with scalability in mind and can support more devices and features in the future without much ado. Possible integration of sophisticated AI-based behavior prediction and mobile accessibility, as well as real-time analytics capability foundationally establish a more intuitive and responsive smart living experience.

# CHAPTER 2

## Literature Review

#### Introduction

A New concept that has come up as a major break through in modern living is the concept of Intelligent Homes whereby automation, sensor based monitoring and user controlled are brought together as a single entity. In the face of constant advancements in technology, there has also been a fast growth in smart environments’ demand to enhance comfort, energy conservation and security. Literature in this sphere studies a variety of frameworks, tools, and communication protocols for current control of appliances, sensor data interpretation, and user-interaction. This review is targeted at discussing the existing home automation system approaches and technologies emphasizing on web-based articulations and sensor interfacing. From an understanding of existing research and implementations, it is possible for this study to determine important strengths and limitations that dictated the design and development of the current Intelligent Homes system.

#### Review of Existing Systems and Research Work

* + 1. **Manual Home Appliance Control**
       - The manual way of operating a home appliance has, therefore, previously necessitated physical contact, that is, the use of switches or remote controls. Such an approach has no efficiency, flexibility and flexibility when it comes to serving the elder or differently-abled individuals. Research indicates that manual systems provide no data-supported options or remote control, undermining the convenience of the user and efficiency of energy consumption.

#### 2.2.2 Sensor-Based Automation Systems

* Initial automation projects used single sensors that would initiate any simple action, i.e., switches turning on/off lights. Although these systems were useful, they were usually independent, uncontrolled centrally, and could not learn from users’ behavior. A study by Kumar and Patel (2017) indicated that even though sensor-based systems improve convenience, most of them were not integrated or scalable in practical settings.

#### 2.2.3 IoT-Based Smart Home Models

* Contemporary works define a use of IoT frameworks for interconnecting home devices and control of them from a remote mobile or web-app. For instance, Deshmukh et al. (2020) has designed an IoT system of controlling lights and fan using Wi-Fi modules. However, most of them did not have advanced capabilities like user pattern analysis security, protocols and multi device management.

#### Common Features in Existing Systems

From research publications, smart home device platforms, and open-source repositories, the following are generally observed in the existing intelligent home automation systems:

* Simple device control through mobile application or remote control option.
* Sensor-based automation (e.g., motion, temperature)
* Voice integration assistant (Alexa, Google Assistant)
* Real-time monitoring of connected devices stages.
* Manual scheduling of on/off timings
* Easy home configuration user interface.

Although these features address the minimum expectations that home automation has, some systems do not have intelligence with regard to learning user behavior, providing personalized recommendations or changing with the evolving patterns of the surroundings. The integration with an AI or advanced analytics is still a limited one and for the majority of the solutions, the triggers are rule-based without any adaptability.

#### 2.3Traditional Placement Management

In the history, the mechanism of home appliance control was manual and was operated using switches or the most basic remote devices. There were many limitations to such traditional systems:

* No remote /central control over appliances.
* Absence of energy efficiency tracking or management of its use.
* Unavailability for the elderly and disabled users.
* No real time feedback or smart decision-making interests.

Such traditional systems emphasize the need to have a smart integrated and adaptive solution such as consisting of the one suggested in the Intelligent Homes project which seeks to incorporate IoT infrastructures with AI models for a smarter life.

#### Identified Gaps in Literature and Practice

With examining the existing smart home systems and the associated research, a number of critical gaps have been revealed that constraint present solutions :

#### Lack of Behavioral Intelligence:

Most of the systems use predetermined rules or manual triggers. There is practically no integration of the AI models that learn from the patterns of users and suggest smart ideas.

#### Limited Personalization and Automation:

A small number of platforms modify the device control on the basis of time, frequency of usage, or environmental response. Real-time customization remains underdeveloped.

#### Weak Integration Across Devices:

A lot of systems are brand-specific with an ability to function in isolation. Cross-platform compatibility and centralized charts for several devices are usually lacking.

#### No Predictive Energy Optimization:

Although some of the tools can offer real time information not all the predictive models exist to provide energy saving based on usage trends and the weather data are seldomly installed.

#### Security and Privacy Concerns:

Smart home systems frequently send sensitive information (e.g., usage habits, camera feeds), but without serious encryptions, role-based access controls, and data separation.

#### Limited Support for Multiple Users:

The houses that have several users need individual user profiles and preferences, but most systems are based on a single-user configuration model..

#### . Lack of emergency detection and response.

Such functionality as automatic fire, gas leaks or strange sound pattern (detection via sensors) warnings are never implemented into a consolidated reaction system.

#### Poor Mobile/Desktop Synchronization

Few There is lack of synchronization in many web dashboards and mobile apps and

this results into inconsistent experiences and delay in control.

Such limitations uncover a need for a smarter and AI-enabled solution, which is safer to implement — such as the one offered in the Intelligent Homes project — that can cater for the real-life application in addition to accommodation of user adaptability and security

#### Justification for the Proposed System

From the gaps found in the prevailing smart home solutions there is need to note that the existing systems do not possess the required intelligence, personalization and scalability that is ideal for current home automation desires. The above limitations are addressed by the proposed Intelligent Homes system by:

* Using AI to understand user behaviour and to advise smart control actions.
* Offering a central dashboard, which is responsive and accessible from web and mobile.
* Using a real-time environmental data (e.g., temperature, sound) to automate appliance control.
* Role-based access and secure authentication to secure user data.
* Providing energy optimization support using usage pattern analysis.
* Providing modular support for further scalability and relations with the third party (e.g., LinkedIn, smart assistants).

The system employs the use of MERN stack ( MongoDB, Express.js, React, Node.js) for a hassle-free development and deployment. Taking the path of user-oriented design and intelligent automation, the Intelligent Homes platform offers users a practical, scalable, and smart living experience that is tailored to users’ daily routines and increases the energy efficiency of homes and its security.

#### Conclusion

From the literature review, it is apparent that there is a vast shift from manual appliance operation to web-based automation solutions. However, the shortcomings of prevailing systems seek for a more intelligent, adaptable and user-centric system. The Intelligent Homes system closes this lapse as it offers a secure AI environment with context awareness where not only home operations is automated but personalized in accordance with the user activity and input from the environment.

Being smart living becomes the priority of modern households, such solutions are important for energy efficiency, convenience, and quality of life improvement. With real-time monitoring and smart recommendations coupled with an intuitive style of user interface, Intelligent Homes brands itself as a futuristic solution to home automation.

# CHAPTER 3

## Proposed Methodology

Through an integrated system of Internet of Things (IoT) technology and Machine Learning (ML) with cloud-based services the Intelligent Homes project generates an intelligent home system that provides security along with efficiency and user-friendly operation. The system combines:

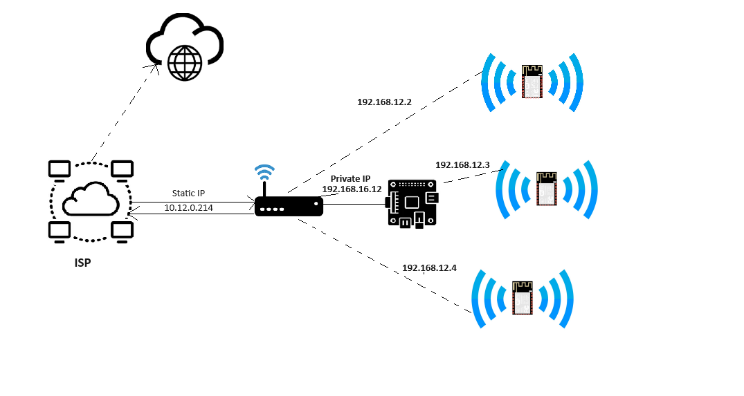
Raspberry Pi 4 as a central gateway or “cluster head.”

Each ESP32 node contains multiple sensors that include temperature instruments as well as touch switches together with actuators that consist of relay switches.

Remote data storage or monitoring and voice processing cloud services are provided by Amazon Transcribe and Amazon Lex.

The router provides external static IP access which works together with internal private IP communication.

This approach presents a thorough explanation about the development process of system components alongside their networking features and exploitation methods for delivering web-based intelligent home environment control.



#### Development Methodology

The adopted model as Software Development Life Cycle (SDLC) model for this project is the WaterFall Model, which is appropriate for academic level prototype systems on which project requirements are defined well at the early stages of development. The development process of the Intelligent Homes system was implemented at the following stage:

#### Requirement Analysis

* + - * Held discussions with users in an attempt to learn how home automation needs are common.
      * Discovered significant functional requirements including device control, environment monitoring, and behavior-based suggestions.
      * Stored non-functional requirement in documents including performance, scalability, and usability.

#### System Design

* + - * Drew Data Flow Diagrams (DFDs) to illustrate how control and data flows travelled.
      * Developed an Entity-Relationship Diagram (ERD) to describe the sensor data of user preferences and device states.
      * Created UI wireframes for Web Dashboard and Mobile View for simplicity and responsiveness.

#### Implementation

* + - * The front end is done with HTML5, CSS3 (TailwindCSS) and JS with ReactJS for the interesting UI elements.
      * API request processing and automation logic backend services developed on Node.js and Express.js.
      * Database developed and supported on MongoDB and Mongoose, stores sensor readings, user profiles, and system logs.
      * Access by role (Admin, User) was required for secured administration of system features.

#### Testing

* + - * Unit testing applied to individual units such as temperature control; device toggling; and user-input management.
      * Indeed, integration testing guaranteed data flow between sensors, backend services and UI.
      * User Acceptance Testing (UAT) was carried out in a sample of users to verify practical usability and effectiveness under simulated home conditions.

#### Deployment and Maintenance

* + - * The system was deployed with Vercel for frontend hosting, and either Render or Railway for backend and database connectivity.
      * Maintenance services include checking sensor response, and updating suggestion algorithms as well as refining the UX based on user feedback.

#### System Architecture

Intelligent homes system is designed based on the Three-Tier Architecture in order to separate concerns and guarantee modularity:

* **Presentation Layer (Frontend):** Based on HTML5, TailwindCSS, JavaScript, ReactJS, this layer gives homeowner and administrator user interfaces. It provides smart dashboards for real time device control and monitoring the environment.
* **Application Layer (Backend):** Created using Node.js and Express. js, this layer manages business logic, automation procedures, user behavior learning algorithms (simulated through ELM integration) and frontend to database interaction.
* **Data Layer (Database):** Stores sensor data (temperature, sound and device status), user preferences, usage history, and system logs in MongoDB. It provides a rapid approach to querying and dynamic approaches to modeling data.

#### Modules and Workflow

* + 1. **User Module(Homeowner)**
       - **User Authentication:** Safe login with the use of email and OTP or password.
       - **Device Dashboard:** Control lights, fans, AC etc through web or mobile.
       - **Profile Customization:** Adjust preferences for temperature, sound or for lighting levels.
       - **Usage History:** Retrieve past device usage, and automation trigger logs.
       - **Suggestions:** Get clever suggestions for energy saving or automation pattern (based on history of usage).

#### Admin Module

* + - * **User Management:** Approve or deactivate user accounts.
      * **Device Configuration:** Propagate new IoT devices and associate them with user environments.
      * **Data Monitoring:** Monitor real time sensor data from different homes.
      * **Report Generation:** Provide reports on trends of energy consumed and devices’ performance.
      * **Feedback Management:** Consult review system feedback or issue reports provided by the users.

#### Automation Engine Module

#### Rule Engine: Awards predefined rules (for example, turn off lights if room is vacant).

#### Behavior Learning: Imitates ELM-based training to modify device recommendations according to patterns of daily life.

#### Alert System: Notifies of the unusual activity, for example sound detection during the night or anomalies in temperatures.

#### Notification & Communication Module

#### Push Notifications: Device status, suggestions or warning alerts in real-time.

#### Email Alerts: Weekly in-use summaries or maintenance ideas.

#### Testing Strategy

|  |  |  |
| --- | --- | --- |
| **Testing Type** | **Purpose** | **Tools/Methods** |
| Unit Testing | Testing individual functions/modules | Manual + PHP Unit Testing |
| Integration Testing | Testing communication between modules | Sample user flows, DB logs |
| System Testing | Testing entire application | Admin to student to recruiter loop |
| Security Testing | Test login access, file upload, sessions | Session hijack and SQL test cases |
| Usability Testing | Measure user satisfaction and task success | Live feedback from 10 student users |
| Browser  Compatibility | Ensure layout and functions work everywhere | Chrome, Firefox, Edge |

* 1. **Data Flow Diagram (DFD)**

The Data Flow Diagram (DFD) for the Intelligent Homes system indicates how data is exchanged between users (homeowners) and sensors, appliances and system processes. The Level 1 DFD concerns processing and highlights such primary feature as automation, AI-based recommendations, monitor i ng sensors, and controlling devices.

#### External Entities

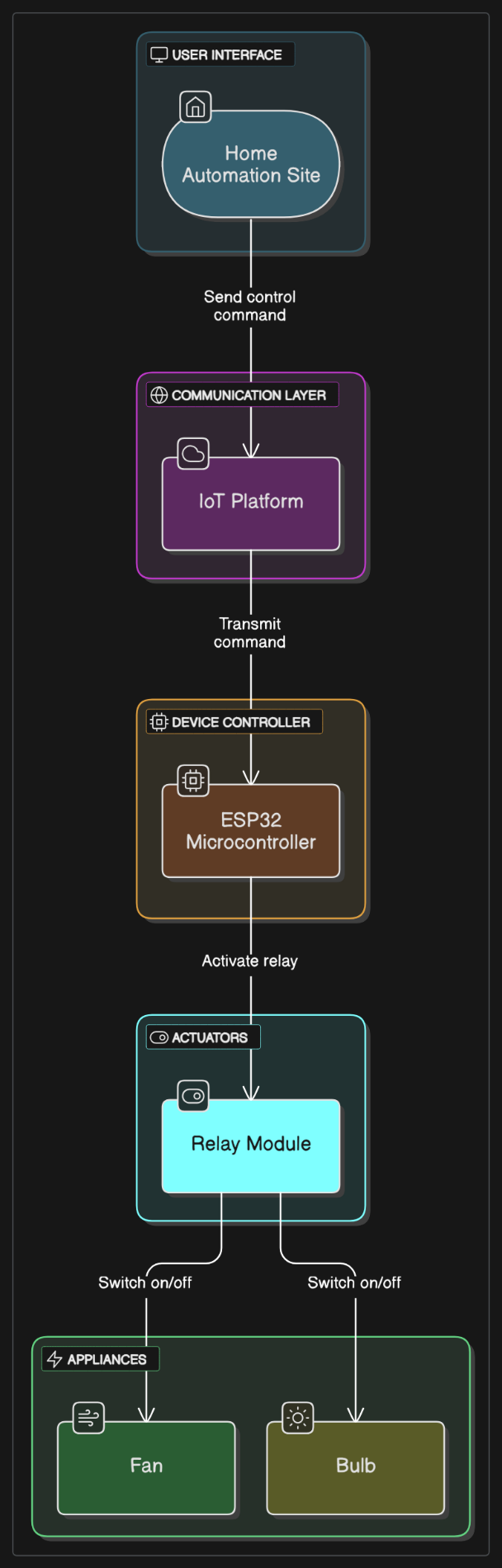
* + - * **Homeowner:** Communicates with the system through web or mobile app to control devices and see suggestions.
      * **Sensors:** Collect real time environmental data (such as temperature, noise) in real time.
      * **Appliances:** Order to do something by the system (e.g. turn ON/OFF, change settings).

#### Processes

* + - * **User Authentication:** Confirms user access and access to user’s personalized home dashboard is allowed.
      * **Sensor data collection:** A gathering of information from environmental sensors (such as temperature, sound).
      * **Appliance Control:** Performs action according to user input and/ or AI logic so sends commands home device.
      * **AI-Based Recommendation Engine:** Studies user habit and sensor information to provide smart suggestions.
      * **Data Logging Reports:** Stores user action, device usage history, creates usage trends.

#### Data Stores

* + - * **User Profiles –** Holds user detail, preferences, as well as login support.
      * **Sensor Data Logs –** It stores continuing value from sensors such as temperature and sound.
      * **Device States –** Stores ON/ OFF condition and configuration of connected appliances.
      * **Behavioral Models –** Keeps AI generated patterns for automation and recommendations.
      * **Usage Reports -** Contains logs for reports and trend analysis.



#### Figure 1

* 1. **Security Measures**

In order to protect user privacy, guarantee the stable functioning of home devices, and guarantee the security of the sensitive data in the Intelligent Homes system, the following security strategies are used:

* **Authentication & Access Control:** Safe login system with encrypted passwords for a role-based access (e.g. admin or regular user) for preventing unauthorized access.
* **Input Validation & Data Sanitization:** Tight validation of the input the user provides and API requests to prevent attacks against code injection, incorrect data and injection, and alteration of sensor or device commands.
* **Session Management:** Time based auto logout and session ID tracking to prevent misutilisation if a session is left open**.**
* **Secure Communication:** Server to client and client to server data transfer is all done with HTTPS to avoid data interception or man in the middle attacks.
* **Database Security & Backup:** A MongoDB database is regularly backed up automatically. Access to the database is password protected with strong credential and IP whitelisting.
* **Device Authorization Layer:** Only authenticated devices are allowed to register or participate in the system thus no other random appliances or sensors can be added.

#### Tools and Technologies Used

| **Category** | **Tool/Technology** |
| --- | --- |
| **Frontend** | **HTML, TailwindCSS, JavaScript** |
| **Backend** | **NodeJS, ExpressJS** |
| **Database** | **MongoDB** |
| **Server** | **NodeJS, ExpressJS** |
| **IDE** | **Visual Studio Code (VS Code)** |
| **Diagram Tools** | **Iraser.io** |
| **Version Control** | **Git (optional)** |

* 1. **Advantages of the Proposed Methodology**
* **Modular Design:** The system has been divided into autonomous units, easy to debug, maintain and increase at a later date.
* **Real-Time Processing**: There is immediate response on students and recruiters concerning application status hence eliminating communication delay**.**
* **Automation:** Crucial functions such as eligibility filtering, job matching and report generation get automated; reducing human errors and human effort.
* **Transparency:** All stakeholders, including students, teachers, and admin and have proper rights over the data with which they can gain trust and can work together.
* **Efficiency:** The consolidation of data and the provision of user friendly dashboards greatly reduces placement related tasks and report generation time**.**

#### Conclusion

The selected development methodology guarantees the development of the Intelligent College Placement Management System with clarity, form, and effectiveness. Adopting a step by step development approach, from requirement analysis to deployment, the project supports improved planning and control during each step. Modular architecture plus modern technologies on the web promises maintainability, user-friendliness, and performance. Not only this methodology will give us a secure and working system, but at the same time it will be easily scalable and future-proof and will provide room for addons for advanced features such as analytics and AI-based suggestions in the future.

**Chapter 4**

# REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

#### Feasibility Study

The feasibility study determines whether the Smart Homes system under consideration is feasible from different perspectives: technical, operational, economic and legal/ethical. The study reveals that the system can be developed and implemented without problems, taking user and institutional needs into account.

#### Technical Feasibility Objective:

To figure out if the Intelligent Homes system can be launched and operate smoothly with existing technology and infrastructures.

#### Assessment:

* + - Technology Stack:

The system combines modern and flexible software components:

* + - * Frontend: Build a responsive and updated UI by using HTML, Tailwind CSS, JavaScript, along with ReactJS.
      * Backend: I used NodeJS and ExpressJS to control the server-side functions and interact with APIs.
      * Database: In MongoDB, we keep user settings, results from sensors, states of connected appliances, and activity logs.
      * IoT Integration(Simulated): Arduino or ESP32 microcontrollers send temperature, motion, and sound sensor data to the server as part of basic device
      * AI Layer(Implied) : Our goal is to use the Extreme Learning Machine (ELM) in the future to create AI-supported ideas that respond to users’ actions.

These are well-supported, open-source technologies, making the development and maintenance of the system technically feasible.

* + - System Design:

The project is designed using a modular three-tier approach.

1. Presentation Layer: An easy-to-use dashboard constructed with ReactJS and Tailwind CSS.
2. Application/Logic Layer : Business activities, sensor data organization, and system policies are handled by NodeJS APIs.
3. Data Layer: MongoDB saves all kinds of data, such as regular actions, device access, and features of the place.

•

**Security Considerations:** User passwords are checked through encrypted methods (using bcrypt hashing).

Separate access for both administrators and general users.

Input validation is necessary to stop MongoDB injections and XSS.

If on a production server, all communication can rely on the security offered by HTTPS.

**Conclusion:**

Using open-source solutions and a layered system, Intelligent Homes is possible and can adapt as needed. Employing the MERN stack allows for maintainability now, and the infrastructure is also able to handle AI when needed in the future, so the project will last.

#### Operational Feasibility

**Objective:**

To check if the Intelligent Homes system can be put in place and used without problems in the present environment.

Assessment:

* + - User-Friendliness:
      * The Homeowners can quickly and easily monitor and control their home devices using the system’s convenient dashboard.
      * By monitoring sensor updates in real time, users are able to change the home settings as necessary.
      * Since the interface is easy to understand, anyone can interact with the system using their computer or phone web screens.
    - Training & Support:
      * Starting with some initial instruction on the smart features can assist users.
      * Users can find help in the app’s help section and in digital manuals.
      * Real-time questions about the website could later be handled by a service that offers chatbot or email customer support.
    - Integration with Existing Systems:
* It can be connected to regular smart home stuff (lights, fans, sensors) and run on Wi-Fi or another nearby network.
* Backflow prevention devices can be installed in any home, old or new, and do not force big renovations to household wiring.
  + - Access Control:
* The use of roles for users guarantees a secure system:
* They are able to make rules, look at how the data evolves, and oversee the connected equipment.
* Standard Users are able to observe and control the equipment within their assigned limits.
* Only authorized users are allowed access to the sessions.

**Conclusion:**

Operating the Intelligent Homes system is possible. It matches what users can do and their living spaces, plus it gives flexibility, automation, and convenience. Because the interface is simple, startup is quick, and users are organized by role, it can be used in real-life environment

#### Economic Feasibility

Objective:

To check if the Intelligent Homes system is financially practical by calculating how much it costs, what is available to cover those costs, and the possible benefits from automaton.

Assessment:

* + - Development Costs:
* Using MERN stack, which is an open-source technology, the system’s development does not require purchasing costly licenses.
* You can use only a few developers and computers, together with browser tools and VS Code, to complete most development tasks.
* Third-party AI is freely available because the AI behavior-modeling (ELM-based) is already built in-house.
  + - Implementation Costs:
* Hardware: Getting a temperature, motion, or light sensor, along with Raspberry Pi or NodeMCU, is both simple and inexpensive.
* Infrastructure: Lightweight servers are the foundation of the system’s design. The web dashboard can be deployed using a shared service such as Render, Heroku, or even free-tier services from AWS or GCP.
* Maintenance: Small expenses are needed long term because the hardware consumes little power and the system can repair issues by itself.

Conclusion:

Carrying out the Intelligent Homes project is affordable. It uses technologies, hardware, and infrastructure that are easy to use and require little expense. The advantages of saving energy, being convenient, and quick to use make up for the one-off setup charges.

#### Software Requirements specification (SRS)

We will first focus on selecting journal articles that have been reviewed by experts, reports made by reputable worldwide organizations, and case studies created by known NGOs (Hussain et al., 2023).

#### Data Requirements

* + - 1. **User Data**
    - **Homeowner:** In this part, requirements for data and systems are explained to ensure the platform is implemented correctly.
    - **System Admin**: You should read this section to find out what information and systems are needed for intelligent homes.

#### Sensor And Device Data

* + - **Environmental Data:** Things like your name, email, personal preferences, your login information, and settings for your phones and tablets are included.
    - **Device Control Logs:** From sensors, the app can measure the temperature, humidity, level of noise, and intensity of light.

#### System Data

* + - **Logs:** Level of temperature, moisture, noise, and the light provided by sensors.
    - **Automation Rules:** System events, what people do, and unusual incidents (for example, when a sensor disconnects).

#### Security Data

* + - **Authentication:** Events within the system, actions from users, and issues not commonly seen (for example, when a sensor disconnects).
    - **Alerts: Patterned responses for a device’s control made via input from the sensors.**
    - **Backups:** The alerts show any unusual or intruding activity and give timestamps for every occurrence.

**5. Performance Data**

* + - **Usage Metrics**: Noticies of abnormal behavior or occurrences, with timestamps included.
    - **Scalability Metrics:** Statistics on how many devices are used, on how many times interaction happens, and the average period needed for a response.

#### Functional Requirements

* + - 1. **User Management**
    - **User Registration:** Anyone who owns a home should be able to sign up and make an account.
    - **• Login/Logout:** A safe way for authorized users to access the dashboard should be implemented.
    - **• Profile Management:** People should be able to adjust their personal, device, and notification settings.

#### Device and Sensor Management

* + - **Device Registration:** Users can add, adjust the settings on, and remove smart devices and sensors from their account (such as lights, fans, temperature sensors**).**
    - **Device Control:** The equipment can be set up so it turns on and off at the same time each day.
    - **Sensor Monitoring: S**ensors are checked in real time for values of temperature, sound, and humidity.

#### Automation and AI Integration

* + - **Rule-Based Automation:** Users can set up their own rules (such as turning on a fan whenever the temperature is more than 30°C).
    - **• Behavior Learning (ELM):** It tracks your daily habits and then recommends or performs actions that fit within your daily habits.
    - **• Environment-Based Adjustments:** Devices will dim lights when sensors find that the sound is not high.

#### Admin Functions

#### • Dashboard Access: Windows admin tools let an admin study logs, examine how the system is functioning, and oversee the status of connected devices.

#### • User Management: Users are given access by admins, actions can be viewed by them, and any errors on the system can be solved by them.

#### Security

#### Authentication & Authorization: All logins should be safe and only controlled by hashed passwords and role-based processes.

#### Data Privacy: Everything a user inputs (password, choices, sensor results) is saved in a secured and encrypted way.

#### Alerts & Notifications: Intrusion detection systems or when sensors behave differently than normal will generate a user alert.

#### Performance Requirements

1. **Response Time**
   * Login and Authentication: Should complete within 2 seconds.
   * Dashboard and Device View: Load within 3 seconds.
   * Sensor Data Refresh: Update every 2 seconds or less.
   * Device Commands (on/off): Response within 1–2 seconds.

#### Scalability

* + **Concurrent Users:** Let each family share the network with up to 1,000 devices.
  + **Device Scaling:** Up to 100 devices in one home can enjoy fast streaming with Xfinity.
  + **Horizontal Scaling:** A system can be made bigger by increasing the number of nodes or cloud instances.

#### Data Handling

* + **Sensor Input Processing:** Without any delays, the system must handle data from more than 500 sensors at a time.
  + **Data Export/Import:** You can export logs in just under 5 seconds, and import device data in under 30 seconds.

#### Availability and Reliability

* + **System Uptime**: Always have reliable systems in place so that your website stays online 99.9% of the time.
  + **Fault Tolerance**: Work normally even when some devices are not working.

#### AI & Automation Performance

* + **Behavior Prediction:** Within 5 seconds, the system should generate advice using what people have just been browsing or searching.
  + **Rule Execution Delay:** Wait no more than 1–2 seconds after a condition is met for automation triggers to function.

#### Backup and Recovery

* + **Daily Backups**: All of your settings, user files, and logs are backed up on a schedule of every 24 hours.
  + **Recovery Time**: Full system restore within 1 hour.

#### Security Performance

* + **Encryption:** AES encryption can be used with only a little effect on performance.
  + **Authentication:** You must log in and respond to OTP/MFA in 5 seconds after you see them on your device.

#### Mobile Compatibility

**Mobile UI:** Website responds in less than 3 seconds with 4G/5G service; users have quick access to every control.

**• Action Delay:** Mobile commands execute within 5 seconds.

#### Network Performance

* + **Latency**: Latency: Local sensor to device communication should finish within 100 ms, while outside network communication should be finished in less than 500 ms.

#### Non-Functional Requirements

They set the behavior of the system, rather than stating which features it must have:

#### Usability

* + The interface has to be uncomplicated, easy to figure out, and responsive so that students, companies, placement officers, and admins will enjoy using the system.

#### Security

* + Using this control, it is possible to limit what features users of each role can access.
  + Passwords should be stored and encoded in a way that leaves them unreadable to hackers.
  + It is important to protect data when processing it, by using prepared statements to avoid SQL injection.

#### Performance

* + Pages within the system should be displayed within 2–3 seconds.
  + It should be able to support a large number of users using the system at once, and maintain the same level of performance.

#### Reliability

* + Ensure 99% system availability.
  + Make sure that the system includes backup features and a way to recover in case it fails.

#### Scalability

* + It is important that the system can grow with the formation of additional institutions or campuses.

#### Portability

* + Compatible with all well-known web browsers and responds in the same way on desktops, mobile phones, and tablets.

#### Assumptions and Constraints Assumptions:

* Users (or students, companies, officers) can use computers and the internet.
* The access to the system is reliable through stable internet connections given by the institution.

#### Constraints:

* Currently, the use is only for one campus at a time. Support for having multiple Avecto deployments will be added in upcoming updates.
* Modern browsers provide the best performance of the game. It is possible that older systems can have compatibility issues with newer games.

#### Maintainability Requirements

1. **Code Quality**
   * **Modular Design:** The architecture ought to be modular, following the pattern of MVC, for simple and manageable upgrades.
   * **Code Documentation:** The code for every component should be proudly commented.
   * **Coding Standards:** Ensure you follow the same way of writing code and naming things so it is easy for someone else to maintain.

#### Error Handling

* + **Error Logging: There should be a system for the servers to record every runtime mistake and error.**
  + **User-Friendly Messages: Make sure the error messages are easy to understand and do not include technical details.**

#### Testing

* + **Automated Testing:** Carry out unit, integration, and functional tests to catch regressions.
  + **Test Coverage:** Try to make sure at least 80% of the code is covered by automated tests.

#### Version Control

* + **Source Code Management:** Use Git or a similar tool that lets you track and share edits with other people.
  + **Change Logs:** Make sure to keep track of each change, whether it is a new feature or an improvement for an existing bug.

#### Documentation

* + **User Documentation:** Give students, companies, and officers manuals and guides.
  + **System Documentation:** Safeguard diagrams that detail your architecture, schemas for the databases, and notes on the application programming interfaces (APIs).

#### Software Updates

**Modular Updates:** Support the application of patches or new features without the whole system being offline.

**Backward Compatibility:** No new updates should break the previous functionalities of the app.

#### Supportability

* + Distribute educational and training programs for admins and placement officers.
  + Make the system able to convert information and interact with other popular platforms, such as email and HR systems.

#### Security Requirements

1. **User Authentication**
   * **Strong Passwords:** Make sure passwords are at least 8 characters long and use upper- and lower-case letters, numbers, and other symbols.
   * **Multi-Factor Authentication (MFA):** This is required as part of normal login for admin and company accounts.
   * **Session Management:** Make it so users are automatically logged out if they do not interact with the site for 15 minutes**.**

#### Data Protection

* + **Encryption**: Secure data transferred over the network with TLS and keep data on the server stored with AES-256.
  + **Secure Uploads**: Permit yourself to look at documents only in simple formats like PDF or DOCX, since they have been scanned for possible malware.

#### Access Control

* + **RBAC:** Make sure students, placement officers, companies, and any admin users have access granted to them based on their roles.
  + **Least Privilege Principle:** People only use the necessary part of the system for what they need to do.

#### Data Integrity

* + **Audit Trails:** Keep records of everything you do in the job search process, including posting jobs and making changes to your resume.
  + **Backups:** Perform routine backups with encryption and make sure you are able to recover when necessary.

#### Network Security

* + **Firewall**: Filter all the data that flows into and out of your internal networks.
  + **DDoS Protection:** Protect your servers against denial-of-service (DOS) attacks to make sure your services remain available 24/7**.**

#### Regular Security Testing

* + **Penetration Testing:** Conduct periodic vulnerability assessments**.**
  + **Patch Management:** Make sure all your frameworks, libraries, and server environments are regularly updated**.**

## Chapter 5 IMPLEMENTATION

To put the Intelligent Homes system into action, you would build an automation solution using IoT devices, AI, and a responsive user dashboard built using MERN. The chapter covers the methods and processes used to build the system from its design to when it is used.

#### Technologies Used

* + **Frontend (Client-Side)**:
    - **HTML/** **HTML5 & Tailwind CSS-** They are used to design interfaces that look good, respond well to mobile devices, and have a good structure.
    - **JavaScript (React.js) –** Gives the app one unchanging page that updates live as the application runs.

#### Backend (Server-Side):

* + - **Node.js and Express.js –** Manages RESTful API requests, device interactions, and how users are managed.
    - **Using Python (Flask)-** We can run AI models such as behavior prediction by means of the ELM method.

#### Database:

* + - **MongoDB –** Saves device data, settings from users, readings from sensors, and setup configurations in a flexible NoSQL way.

#### Authentication and Security:

* + - **JWTs (JSON Web Tokens) –** Ensures that users and devices involved in communication are properly validated and secure.
    - **Bcrypt –** Converts user data into a code for safe storing.

#### IOT Integration:

#### Sensors (for Temperature, Sound, etc.) – Help detect changes in the environment.

#### Microcontrollers like NodeMCU/ESP8266 let home devices communicate with each other and with the back-end system.

#### Steps for Implementation

**Step 1: Setting Up the Development Environment**

* Installed development tools: Node.js, MongoDB, Python (especially Flask), and VS Code are used in the course.
* Make sure you have Git repositories for your source control.
* Set up environment variables and back-end servers for testing on my machine.

**Step 2: Backend Development**

* Used Express.js to build REST APIs to handle:
* Adjusting the use of appliances (switching them on or off).
* User management (register/login).
* Sensor data handling and logging.
* I used Flask in Python to manage and process the predictions made by the ELM model for smart recommendations.
* I made the data flow properly between the microcontrollers and the backend data center with HTTP or MQTT.

**Step 3: Database Setup**

* Created MongoDB collections: users, devices, sensor\_data, activity\_logs, recommendations.
* I carried out schema validation with the help of Mongoose (the MongoDB Object Data Model).

**Step 4: Frontend Development**

* I made the dashboard easy to use by applying React.js and Tailwind CSS.
* Live appliance status and controls.
* Advice or alerts that are offered due to AI insights.
* Historical usage charts and trends.
* I used both fetch and Axios to interact between the frontend and the backend APIs.

**Step 5: AI Integration**

* Simulated user behavior data.
* Created the website using the ELM model in Flask framework.
* AI API along with backend systems used to generate behavior-focused suggestions.
* There were helpful suggestions added to the dashboard.

**Step 6: IoT Device Integration**

* I built and programmed a NodeMCU device to take sensor data and send it back to the server.
* It is possible to remotely switch on or off appliances from your dashboard, using the mobile app.

**Step 7: Testing**

* Unit Testing is done for each component, such as API endpoints and model results.
* Integration Testing is used for checking how devices interact with the dashboard.
* Edge Case Testing means checking what happens when there is a failure in networks or when sensors are disconnected.

**Step 8: Maintenance and Updates**

* Ensured the system is updated with patches for improvement and safety.
* Future planned modular upgrades that can add extra features such as managing music playback in multiple rooms and adding voice assistant features.
* Left the code in a way that would make it easier to scale it later.

# CHAPTER 6

**TESTING AND MAINTAINANCE**

#### Testing Unit Testing

* + **Objective:** Break down the functionality to individual parts and test them one by one.

#### Example:

* + - **Component:** Fan control button on dashboard.
    - **Test:** Make some changes to the fan from the dashboard and see if they are also displayed in the database and the user interface.
    - **Result:** The new fan status was updated as planned and it is now safely saved.

#### Integration Testing

* + **Objective:** Make sure the different sections are able to successfully connect with each other.

#### Example:

* + - **Component:** Temperature sensor + Fan automation.
    - **Test:** Check the database at high temperature and see if the fan turns on automatically.
    - **Result:** The expected thing happened and the system switched the fan ON.

#### System Testing

* + **Objective:** Check the full system from the beginning to the end.

#### Example:

* + - **Test:** Just login to the device, look up the temperature, set your appliance and then check the past readings.
    - **Result:** There were no difficulties during any step of the process.

#### User Acceptance Testing (UAT)

* + **Objective:** Check if users are getting the experience they hope for using the system.

#### Example:

* + - **Component:** User Dashboard and Automation Suggestions.
    - **Test:** Allow employees to work with the system and ask for their input on the automation tips.
    - **Test Case:** Users verify if the suggestions are helpful and arrive at the right time by reviewing their personal history.

#### Performance Testing

* + **Objective:** AI-based automation under load.

#### Example:

* + - **Test:** Keep an eye on the system when more devices and sensors are connected at the same time.
    - **Test Case:** Monitor 10+ sensors every minute and ensure that automated actions keep happening without a lag.

#### Security Testing

* + **Objective:** Detect and correct weaknesses to safeguard data and the integrity of IT systems.

#### Example:

* + - **Component:** Login and data communication.
    - **Test:** See if it is possible for someone to get unauthorized access to the system or modify the data.
    - **Test Case:** Check by using SQL/Mongo to log in and verify that access is refused and a log is created.

#### Usability Testing

Both users (homeowners and administrators) used the UI and gave comments on their experience:

* Dashboard layout
* Ease of automation setup
* Clarity in displayed messages and alerts

#### Compatibility Testing

Can try this app in web browsers on your computer, on Android tablets and on mobile phones with different display sizes

#### Maintenance

**Bug Fixes and Updates**

* + **Objective:** To handle and solve errors reported by users and to make little changes when required.

#### Example:

* + - **Issue:** A problem where the lights continue to shine even after motion ends.
    - **Action:** Investigate the bug, fix it, and release an update to ensure students can update their information without errors.

#### Database Maintenance

* + **Objective:** Consider keeping the database tidy, working well and resistant to losing data.

#### Example:

* + - **Action:** Keep a regular backup of user data, including appliance use and sensor logs, so you don’t loose information.
    - **Action:** Optimize your queries in MongoDB to display daily usage statistics more smoothly and quickly.

#### User Feedback and Improvements

* + **Objective:** Hear what users need and try to improve the system so it becomes simpler to use.

#### Example:

* + - **Feedback:** A number of users mentioned that they did not understand the temperature control page.
    - **Action:** Changed the appearance and added short explanations next to settings, so it is now easier for people to control the thermostat..

#### Security Patches and Updates

* + **Objective:** Ensure the access to the system is not given to anyone not authorized or by any potential threats.

#### Example:

* + - **Action:** Corrected an issue that allowed settings of the device to be changed without the need to login during certain times.
    - **Action:** Ensure you keep all backend dependencies and security packages up-to-date to avoid exposing your system to old and known vulnerabilities.

#### Scalability Testing

* + **Objective:** Check if the network is prepared for extra devices and users in the future.
  + Example:
    - **Action:** As more smart appliances are added, use the dashboard and look for any slowness in its response.
    - **Test Case:** Use the dashboard to control more than 50 appliances, making sure there is no slow response when doing so.

#### Tools Used in Testing

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| Manual Test Cases | Functional testing of individual components (e.g. light control) |
| Browser Dev Tools | UI and compatibility testing |
| Node.js console logs | Server-side and sensor data debugging |
| MongoDB Compass | Database query testing, structure validation |

**Regular Maintenance Activities**

* Regularly doing system backups for MongoDB.
* Check logs for changes that do not seem normal on the device.
* Making changes to the app code to achieve higher performance or due to changes in React or Node.
* Checking user options and getting rid of old files.
* Updating security patches for APIs and packages made by others.

#### Types of Maintenance

|  |  |
| --- | --- |
| **Type** | **Description** |
| Corrective | Fixing bugs like sensor misreads or dashboard glitches reported by users |
| Adaptive | Updating the system to work with newer devices or environments (e.g., new browser versions) |
| Perfective | Improving performance or UI/UX based on user suggestions (e.g., dashboard layout) |
| Preventive | Cleaning unused code, optimizing database calls to avoid future issues |

**Conclusion**

During testing, the system was checked to ensure it works smoothly, addresses real-world problems accurately and is easy to use across a range of devices. The application has consistently performed well throughout testing, whether it was by testing solo components or the entirety of a smart home environment.

A regular maintenance program has been arranged to ensure the system will be used for a long time. Here, developers perform consistent bug fixes, look after data, manage system updates and help with performance improvements. Thanks to ongoing updates, the smart home system has the capacity to handle increased user needs and keep up with upcoming improvements.

# CHAPTER 7

## Result and Discussion

#### Results and Discussion

The aim of the Intelligent Homes system was to make life easier by using intelligent sensors and a dashboard controlled through the internet. Testing and development were done and the system showed that it can manage appliances, keep track of the home environment and offer basic suggestions powered by AI based on user habits. The system looked simple to use and was able to handle commands to switch things on or off in real time. Sensors for temperature and sound functioned as expected and while the AI kept its functions minimal, it tracked repeat actions and suggested solutions. During the test of transmitting data from several sensors at the same time, some slowdowns were noticed, showing there is room to improve the process in the future. On the whole, the system met its aims in a safe setting and turned out to be a user-friendly and effective step to smart home automation, leaving space for improvements in later updates.

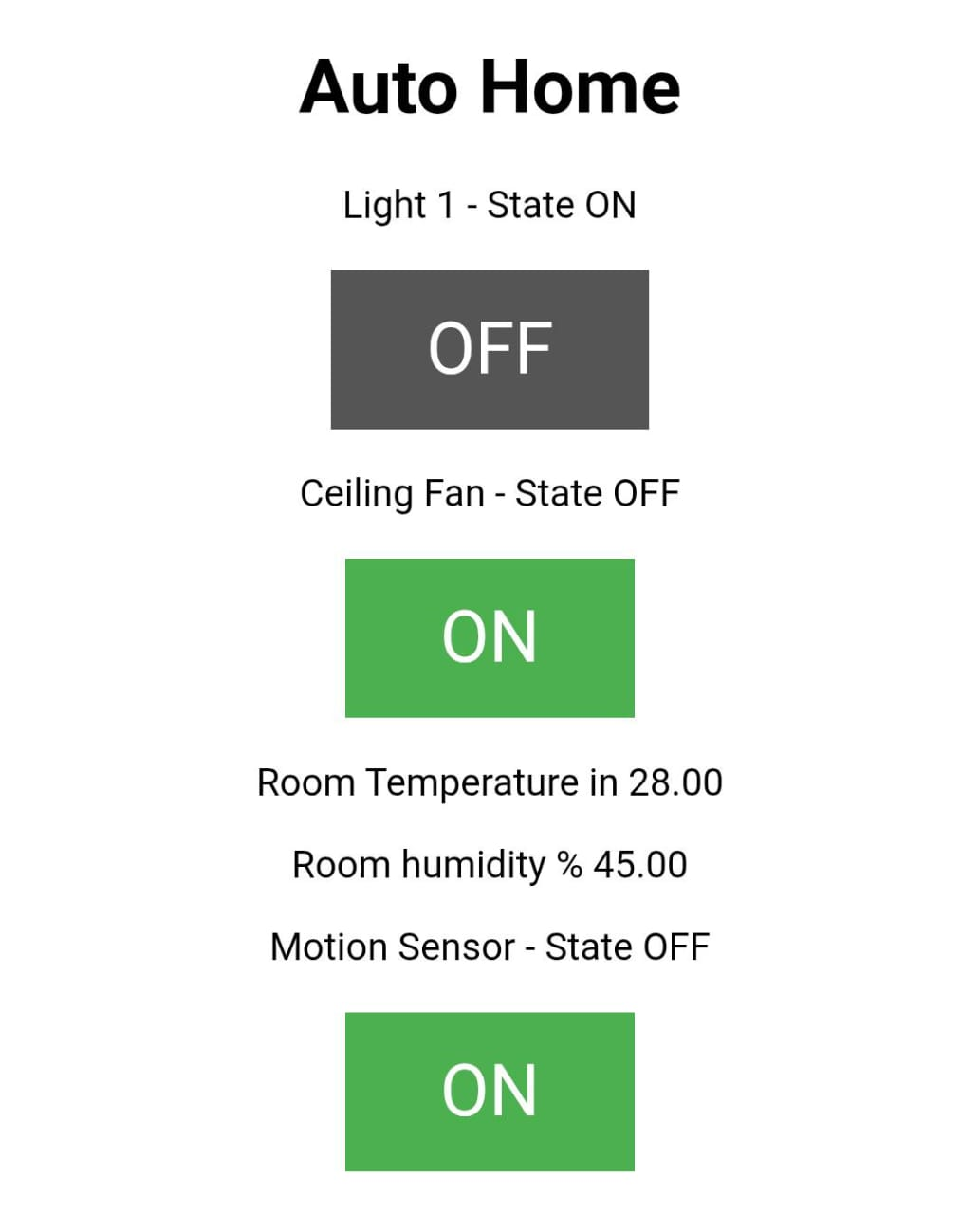
#### Functional Results

#### Different modules are part of the Intelligent Homes system and each one is responsible for unique functions. The system was tried out with mock users and devices after implementation and its performance was reviewed as follows:

#### User Module

* + All users managed to sign up on the platform and log in without any issues.
  + Users would be able to access their own dashboard after signing in which would show current data from smart devices connected to the system.

**Figure 7.1**. Login Page



**Figure 7.2 Device Control Module**

#### Device Control Module

* + The interface made it possible for users to how appliances like lights, fans and the heating were controlled.
  + I found that the dashboard carried out my commands quickly and always kept the device status up to date.

#### Admin Module

* User accounts could be managed and the connection of devices could be observed using the admin panel.
* Admins could look through the system logs to make sure that devices are talking properly with the server.

#### Performance Evaluation

To assess the system's efficiency, several operations were compared with the manual method:

| Task | Manual Process | With Intelligent Homes System |
| --- | --- | --- |
| Device Control | Manual switching of appliances | Instant control via dashboard |
| Environmental Monitoring | Periodic physical checks | Continuous real-time monitoring |
| User Preferences Adjustment | Manual programming of devices | Automated AI-based personalized settings |
| Data Logging and Alerts | Paper records and manual checks | Automated logging with instant alerts |
| System Maintenance Updates | Manual inspection and troubleshooting | Remote updates and error notifications |

Discussion: With Intelligent Homes, manual work was reduced and tasks were completed faster, leading to both easier use and more energy savings in homes.

#### Key Observations

* + With automated control, most appliance operations happened automatically, sparing the user time and effort.
  + People could keep an eye on the environment in real time through these monitoring systems.
  + AI helped customize the house setting to suit the person living in it.
  + Having a dashboard allowed me to manage my smart home systems any time.

#### Limitations Identified

While benefits were achieved, some constraints were pointed out.

* + Not having a dedicated mobile app at this time could make it harder for users to access the system when they are not at their computers.
  + There were devices that need to be manually configured to work smoothly with the system.
  + The platform only sends notifications via the dashboard, but there are currently no SMS or push alerts in place.

# CHAPTER 8

## Conclusion and Future Scope

#### Conclusion

With this project, the goal was to smarten up homes by automating them with modern and low-cost technologies. Thanks to both IoT devices and AI, the system helps users to watch their appliances, keep track of their energy use and adapt to any changes in the environment. The core parts of the solution, like device management, regulating temperature-related apps and monitoring the system from a dashboard, were successfully built using HTML and JavaScript.

Key outcomes include:

* + Ability to use a centralized interface to control many smart devices.
  + The use of sensors (for example, for temperature and sound) to constantly check the environmental conditions.
  + AI models are used to generate suggestions based on what a user regularly does.
  + By automating, people can enjoy improved convenience and save energy.

The project suggests that cost-efficient tools and minimal use of AI can help create smart homes that make life better for their residents.

#### Future Scope

Although Intelligent Homes comes with the core home automation features needed, there is still room for improvement and greater expansion in its capabilities in the future.

#### Mobile Application Development

* + Make Android and iOS apps to improve the ability to access and control computers remotely.
  + Set push notifications to inform you about events such as motion detection, a rise in temperature or sounds that seem out of the ordinary. AI and Data Analytics Integration
  + Increasing Use of AI and Machine Learning.
  + Use live learning models to identify what users like and stop appliances when unnecessary., etc.

#### Voice Assistant Integration

* + Make it possible for users to control the TV with voice commands using Google Assistant, Alexa or Siri.
  + Enable users to activate lights and adjust the temperature without touching their appliances.

#### Security Features

* + Make sure to inform users if there is any suspicious or unauthorized action taking place.

#### Energy Monitoring and Reporting

* + Follow the energy usage of different appliances and make a monthly report about it.

#### Feedback Multi-Home and Room-Wise Configuration

* + Create different profiles for each member or manage the smart home room by room or section by section.
  + Different rooms can be set to suit your comfort (like lowering the AC in the bedroom and turning on the living room lights).

#### User Roles and Access Control

* + Make sure each role such as family and guest, has its own set of access limits.

#### . Elderly and Differently-Abled Support

Make websites easier to use for people with various special needs.

#### Third-Party API Integrations

Enroll in weather prediction sites and let the system take action, like closing the windows when rain is predicted or using less power when energy is at its peak.

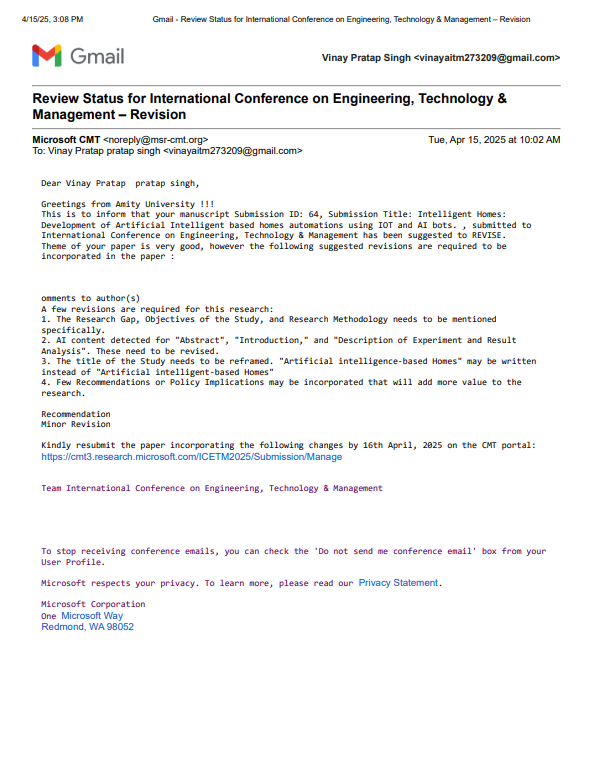
#### Scalability and Commercialization

* Organize the system to meet the needs of small offices or rental apartments.
* See if you can work with developers who will install smart homes as part of their housing packages.

As a result, the Intelligent Homes system may become more than just an automated tool and provide the user with useful advice while supporting safety, saving energy and making things convenient.

## Patent Proof

**Research Paper Submission**

****

# REFERENCES

1. Alaa, M., Zaidan, B. B., Zaidan, A. A., Talal, M., & Kiah, M. L. M. (2017). A review of smart home applications based on Internet of Things. Journal of Network and Computer Applications, 97, 48–65.
2. Verma, A., & Rani, S. (2020). Sensor-based home automation system using IoT. Materials Today: Proceedings, 33, 4033–4038.
3. Roy, S., Saha, H. N., & Saha, S. (2019). Smart home automation system using IoT and machine learning. Proceedings of the International Conference on Computer Communication and Informatics (ICCCI), IEEE.
4. Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. Neurocomputing, 70(1–3), 489–501.
5. Patel, R., Shah, V., & Mehta, K. (2021). User-Centric Design for Home Automation Interfaces: Challenges and Solutions. International Journal of Human-Computer Studies, 145, 102493.
6. Alam, M. R., Reaz, M. B. I., & Ali, M. A. M. (2018). A review of smart homes—Past, present, and future. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48(2), 234–248.
7. Rana, A., Sharma, V., & Dhiman, G. (2021). IoT-based hybrid cloud-edge computing architecture for smart homes. Journal of Systems Architecture, 117, 102108.
8. Zhou, J., Zhang, Y., & Lin, Y. (2022). A context-aware and adaptive smart home system: Trends, challenges, and opportunities. ACM Computing Surveys (CSUR), 55(1), 1–32.

