ACKNOWLEDGEMENTS

We wish to express my profound and sincere gratitude to Mr. Mangaldeep Chakraborty Teaching Assistant Department of Computer Science and Engineering, SoET, Vizianagaram Campus, who guided me into the intricacies of this project nonchalantly with matchless magnanimity.

We thank Prof. Dr. Subrat Kumar Parida, Head of the Dept. of Department of Computer Science and Engineering, SoET, Vizianagaram Campus for extending their support during Course of this investigation.

We extend deep appreciation to Dr. P.A. Sunny Dayal, the Dean of SoET, for his help, leadership, and the conducive academic environment fostered at our institution. Their unwavering commitment to excellence has set a benchmark for all.

Special gratitude is extended to Dr. P.K. Mohanty, the Vice Chancellor of Centurion University, Vizianagaram, for providing a conducive and inspiring educational ecosystem.

A heartfelt note of appreciation goes to all staff members who, in various capacities, contributed to the success of our project. Dedication and collective effort have been indispensable, and we sincerely thank them for their collaboration.

I express my deepest gratitude to my parents for their unwavering support, encouragement, and the countless sacrifices they made, which played an integral role in the success of this project.

ABSTRACT

The objective of this project is to design and implement a voice-controlled smart home automation system using the ESP8266 microcontroller. The system aims to provide users with a seamless and intuitive way to interact with and control various smart home devices through voice recognition. The ESP8266 will serve as the central processing unit, integrating natural language processing algorithms to enhance user communication and facilitate the automation of diverse home appliances. All of the appliances are configured by using the Blynk application and IFTTT server. Finally, Arduino IDE is used configure all of the devices and to sketch the code to the WIFI module to let the appliances run from a mobile device through google assistant. The goal is to create a user-friendly and efficient smart home environment that responds intelligently to voice commands, enhancing convenience and connectivity for the end user.

Keywords: Home Automation, Voice recognition, Blynk application, IFTTT Server

Table of Contents

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1. INTRODUCTION

In the ever-evolving landscape of modern technology, voice-controlled home automation stands at the forefront of innovation, revolutionizing the way we interact with our living spaces. This cutting-edge technology seamlessly integrates the power of voice recognition with smart home devices, enabling users to effortlessly control various aspects of their homes through simple vocal commands. Imagine a home where lights, thermostats, security systems, and entertainment devices respond intuitively to your voice, creating a personalized and hands-free environment. This transformative capability not only enhances convenience but also marks a significant stride towards creating more accessible and inclusive living spaces. As we delve into the era of voice-controlled home automation, the boundaries between science fiction and reality blur, promising a future where the phrase "smart home" takes on an entirely new dimension.

1.1 Purpose

The purpose of integrating voice-controlled home automation with ESP8266 is to create a user-friendly, interconnected, and energy-efficient living space. This technology not only simplifies daily tasks but also aligns with the growing trend of smart homes, offering a glimpse into the future of intuitive and responsive household management. voice-controlled home automation fosters a seamless integration of various smart devices within a household. The ESP8266, a versatile and cost-effective Wi-Fi module, serves as a key component in this ecosystem, facilitating communication between different devices and the central control system. This interoperability ensures that lights, thermostats, security cameras, and other smart devices can be synchronized and controlled harmoniously through voice commands. Additionally, the centralized control provided by voice automation systems allows users to monitor and adjust their home environment remotely, enhancing security and peace of mind

.

1.2 Intended Audience

The intended audience for "Voice-Controlled Smart Home Automation with ESP8266" comprises a diverse range of individuals and groups interested in smart home technology, automation enthusiasts, and those seeking innovative solutions for home management. Specific target audiences include:

Smart Home Enthusiasts: Individuals who have a keen interest in adopting and exploring advanced smart home technologies to enhance their living spaces and experience the convenience of voice-controlled automation.

Technology Enthusiasts: Technologically inclined individuals who are curious about the integration of voice control and IoT devices, seeking to stay updated on the latest innovations and trends in home automation.

Homeowners and Residents: People looking for practical and user-friendly solutions to manage their homes more efficiently, especially those who value the hands-free convenience offered by voice-controlled systems.

Developers and Programmers: Software developers and programmers interested in exploring the codebase, algorithms, and technical aspects of integrating voice recognition and automation capabilities using the ESP8266 microcontroller.

Educational Institutions: Students and educators in the fields of electronics, IoT, and home automation who may find the project valuable for learning and practical applications in their coursework or research.

IoT and Home Automation Professionals: Professionals working in the IoT and home automation industry who seek insights into innovative technologies, potential applications, and practical implementations using ESP8266.

By targeting these audiences, the project aims to cater to a broad spectrum of individuals with varying levels of technical expertise and interests in the realm of voice-controlled smart home automation.

1.3. Scope

The scope of the "Voice-Controlled Smart Home Automation with ESP8266" project is broad and encompasses various aspects of modern home automation technology. The project primarily focuses on developing an intelligent system that allows users to control and interact with their smart home devices effortlessly using voice commands. This includes the integration of the ESP8266 microcontroller to serve as a central processing unit, enabling communication with a range of home appliances. The scope extends to implementing natural language processing algorithms to enhance the system's ability to interpret and respond to diverse voice commands, making the user experience more intuitive and efficient. Furthermore, the project explores the compatibility of the system with popular voice assistant platforms such as Google Assistant, providing a seamless integration into existing smart home ecosystems. The potential scope also involves incorporating features for monitoring and diagnosing the health of connected devices, contributing to proactive maintenance and improved reliability. Overall, the project aims to offer a comprehensive solution for voice-controlled smart home automation, addressing user convenience, innovation, and the evolving landscape of IoT technology.

CHAPTER 2:
LITERATURE SURVEY

2. LITERATURE SURVEY

In recent years, the field of smart home automation has witnessed a paradigm shift, with researchers exploring innovative applications of cutting-edge technologies.

Pedro et al.[1] Drawing inspiration from this, our focus shifts to the domain of voice-controlled smart home automation with ESP8266, where the integration of voice commands and IoT technology promises to redefine user interactions with home environments.

Bhalsing et al.[2] Leveraging this spirit of innovation, our project aims to explore the fusion of ESP8266's capabilities with voice recognition systems, offering a novel approach to controlling and monitoring smart home devices through natural language commands.

Parra et al.[3] highlighted the success of deep learning in various real-world applications, setting the stage for our exploration of voice-controlled smart home automation. We delve into the potential of ESP8266 as a central hub, connecting diverse IoT devices and responding intelligently to voice instructions, thereby enhancing user experience and interaction.

He et al.[4] motivating our endeavor to revolutionize the control of home devices through voice commands. The ESP8266's efficiency becomes pivotal in achieving seamless integration and ensuring quick response times.

Van Ruitenbeek et al.[5] future innovations in smart home automation, exploring novel applications and expanding the capabilities of ESP8266.

Waqas et al.[6] explores the feasibility of training models for recognizing and responding to specific voice commands, adding an intelligent layer to home automation.

Gaykar et al.[7] and
Gaykar et al.[8] proposed enhance the interpretability of voice commands, ensuring a more intuitive and interactive smart home experience with ESP8266.

Zhang et al.[9] envisions leveraging ESP8266's capabilities to process voice inputs and execute commands intelligently, optimizing the overall smart home automation process.

Khan et al.[10] and
Widjojo et al.[11] We extend this perspective to the realm of voice-controlled smart homes, where user interactions generate valuable data for continuous improvement and customization.

Gomathy et al.[13] envisions a comprehensive integration of voice commands, ESP8266, and other IoT technologies to create a seamless and efficient smart home automation system.

Mallikarjuna et al.[14] leverage ESP8266's capabilities to enhance the detection and execution of voice commands, contributing to a more responsive and intelligent smart home environment.

Alam et al.[15] We extend this perspective to the realm of voice-controlled smart homes, where user interactions generate valuable data for continuous improvement and customization.

Yang et al.[16] the realm of smart homes, our project explores the potential of ESP8266 in predicting user preferences and adapting the home environment accordingly.

Ren et al.[17] envisions a comprehensive approach to voice-controlled smart home automation, considering not just device control but also monitoring and adjusting various components within the home environment.

Roy et al.[18] In line with this vision, our project aims to set a precedent for future innovations in smart home automation, exploring novel applications and expanding the capabilities of ESP8266.

Koch et al.[19] the potential integration of diverse algorithms to enhance the interpretability and responsiveness of voice-controlled smart home automation with ESP8266.

Kowshalya et al.[20] streamlining smart home automation through voice commands, ensuring legal and permission-based execution of commands facilitated by the ESP8266.

Wu et al.[21] aims to eliminate the need for manual controls in home automation, introducing efficiency and convenience through voice commands powered by ESP8266.

Guastadiesgni et al.[22] emphasized the adaptability of voice-controlled smart home systems, accommodating evolving user preferences and technological advancements.

Jaiswal et al.[23] emphasized refining smart home automation policies and adapting decision-making processes based on user interactions, improving overall system intelligence.

Böhm et al.[24] harnessed the capabilities of ESP8266 for voice-controlled smart home automation, incorporating image analytics, predictive analysis, and machine learning algorithms.

Keng et al.[25] explores realtime detection and execution of voice commands in smart home automation, leveraging the capabilities of ESP8266.

Egaji et al.[26] achieved high accuracy of voice recognition in smart home automation, ensuring precise and reliable execution of commands.

Bansal et al.[27] explore various models for voice command recognition, identifying the most effective algorithms for seamless interaction with ESP8266.

Ilham Dwi Pratama et al.[28] aspires to catalyze a transformation in the realm of smart home automation, promoting data-driven decision-making and innovation.

Dongre et al.[29] aimed to create a standardized and user-friendly voice-controlled smart home system using ESP8266, promoting accessibility and widespread adoption.

Asad et al.[30] the best-suited models for real-time voice command detection, with a focus on achieving high accuracy and responsiveness with ESP8266

CHAPTER 3: OVERALL DESCRIPTION

3. OVERALL DESCRIPTION

Voice-Controlled Smart Home Automation with ESP8266 integrates cutting-edge technologies to create an intuitive and efficient home management system. The ESP8266 microcontroller serves as the central hub, connecting various smart devices within the home. Through the Blynk app, users can seamlessly configure and monitor devices, ensuring a user-friendly interface. The system employs IFTTT (If This Then That) to enable conditional automation, allowing users to create custom rules for device interactions.

The integration with Google Assistant takes the user experience to the next level, allowing voice commands for hands-free control. Users can effortlessly instruct Google Assistant to perform tasks such as adjusting lighting, regulating temperature, or managing security systems. The system's intelligence is enhanced through natural language processing, making interactions more conversational and responsive.

The ESP8266's connectivity to the Internet of Things (IoT) ecosystem ensures real-time communication and updates, while the Blynk app provides a mobile interface for remote control. The implementation aims to simplify home automation, offering a comprehensive solution that adapts to user preferences. This voice-controlled smart home system not only enhances daily living by providing convenience but also represents a significant step towards creating intelligent, connected, and user-centric living spaces.

3.1. Existing System:-

The conventional home automation systems relied on manual interfaces such as physical switches, remotes, or smartphone applications. Users typically interacted with individual devices or systems using these interfaces, lacking the seamless and intuitive experience that voice control provides. The existing systems often faced the following limitations:

- Manual Control: Users had to manually operate switches or use smartphone apps to control smart home devices, which could be cumbersome and time-consuming.
- Limited Interoperability: Interoperability among different devices from various manufacturers was a challenge, as devices often operated in silos with proprietary communication protocols.
- Complex Programming: Automation tasks required programming skills, and users had to set up complex schedules or rules to automate device actions.
- Less Intuitive Interaction: Interaction with devices was less intuitive, relying on graphical user interfaces or physical controls that may not have been user-friendly for all demographics.
- Limited Mobility: Users were confined to using smartphones or remotes to control devices, limiting the flexibility and accessibility of home automation**.**

3.1.1. Proposed System:-

In our project, we have proposed home automation system which control many devices remotely through wireless communication. We are using NodeMCU which is having inbuilt Wi-Fi. Connecting NodeMCU to internet is much easier as compared to connecting UNO to internet. We can also program NodeMCU in C language using Arduino IDE directly. Node MCU comes with 128KB RAM. It can store more code compared to Arduino UNO. The voice control using IFTTT also facilitates control of appliances over the web. Hence the constraint of proximity concerned with Arduino UNO control is eliminated. The System consists of appliances like lights and fans which will be switched on and off using voice commands on Google assistant. The appliances can be control using two methods in particular, they are App and voice control. Voice control can easily be achieved on mobile using Google Assistant.

3.1.2. Functional Requirements

- Voice Recognition: Implement a robust voice recognition system to accurately capture and interpret user commands for controlling smart home devices.
- Natural Language Processing (NLP): Integrate NLP algorithms to enhance the system's ability to understand context and interpret more complex and natural language commands.
- ESP8266 Integration: Utilize the ESP8266 microcontroller as the central processing unit to manage and coordinate communication between various smart home devices.
- Device Compatibility: Ensure compatibility with a diverse range of smart home devices, including lights, thermostats, security cameras, and other appliances, allowing for comprehensive home automation.
- Blynk App Integration: Integrate with the Blynk app to provide users with a mobile interface for configuring, monitoring, and controlling connected devices from anywhere.
- IFTTT Integration: Implement IFTTT functionality to enable conditional automation, allowing users to create custom rules and automation sequences based on specific triggers or events.
- Internet of Things (IoT) Connectivity: Enable connectivity to the broader IoT ecosystem to facilitate real-time communication, updates, and potential integration with other IoT-enabled devices.
- User Profiles and Preferences: Incorporate user profiles and preferences to customize the smart home experience based on individual preferences, allowing for personalized automation.
- Error Handling and Feedback: Implement error handling mechanisms and provide user feedback to address issues such as misinterpreted commands or device connectivity problems.
- Remote Access: Allow users to remotely access and control their smart home devices through voice commands, enhancing convenience and accessibility.
- Scalability: Design the system to be scalable, accommodating the addition of new devices or functionalities without compromising performance.

- Offline Mode: Provide a failsafe mechanism or offline mode to ensure basic functionality in case of temporary loss of internet connectivity.

These functional requirements collectively contribute to the development of a comprehensive and reliable voice-controlled smart home automation system using ESP8266.

3.1.3. Non- Functional Requirements

Non-functional requirements for a Voice Controlled Home Automation system with ESP8266 can be crucial for ensuring the system's overall performance, security, and user experience.

- Response Time: The system should respond to voice commands with minimal latency, providing quick feedback to the user.
- Throughput: The system should be able to handle a certain number of concurrent voice commands without degradation in performance.
- Reliability: The system should be available and responsive most of the time to ensure a reliable user experience and handle failures gracefully and recover from errors without compromising functionality.
- Scalability: The system should support a scalable number of connected devices to accommodate future expansions and capable of handling a growing number of users interacting with the system simultaneously.
- Security: Ensure that only authorized users can access and control the home automation system via voice commands. Voice data and control signals should be encrypted to prevent eavesdropping or unauthorized access.
- Compatibility: The system should be compatible with a wide range of home automation devices and standards
- Scalability: The system should support a scalable number of connected devices to accommodate future expansions.
- Remote Monitoring and Management: Enable remote monitoring and management of the system to facilitate updates, troubleshooting, and maintenance.

3.1.4. Hardware Interfaces

- RAM: 16GB
- Hard Disk: 512 GB
- System: intel CORE i5

3.1.5. Software Interfaces

- Operating System: Windows 11(64 Bit)
- Arduino IDE

3.2. UML DIAGRAMS

UML is a method for describing the system architecture in detail using the blueprint. UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. UML is a very important part of developing objects-oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

Definition :UML is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of the software system.

UML is a Language

It will provide vocabulary and rules for communications and function on conceptual and physical representation. So it is modeling language.

UML Specifying

Specifying means building models that are precise, unambiguous and complete. In particular, the UML address the specification of all the important analysis, design and implementation decisions that must be made in developing and displaying a software intensive system.

UML Visualization

The UML includes both graphical and textual representation. It makes easy to visualize the system and for better understanding.

3.2.1. BLOCK DIAGRAM

Block Diagram: The block diagram illustrates the interconnected components of a voice-controlled smart home automation system with ESP8266. The NodeMCU ESP8266 acts as the central controller, linking with the Blynk application for device control, IFTTT for event triggers, and Google Assistant for voice commands. Home appliances such as bulbs and fans are connected to the NodeMCU, creating a cohesive network that enables seamless automation and user interaction within the smart home ecosystem in figure 3.1

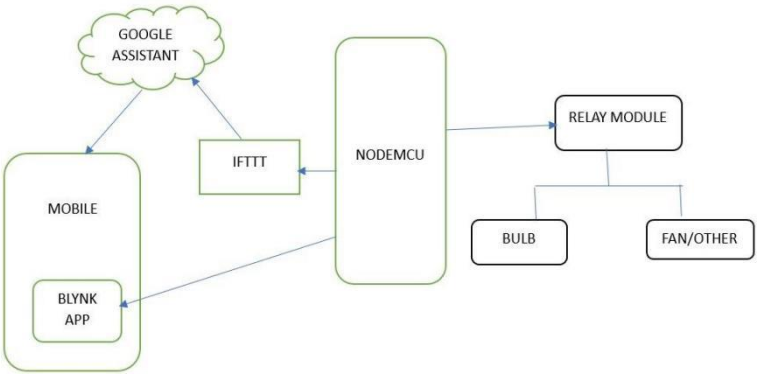


Figure 3.1: Block Diagram

3.2.2. CIRCUIT DIAGRAM : The circuit diagram illustrates the interconnected components and the flow of control signals, creating a comprehensive smart home automation system that responds to both manual and voice-based user inputs in figure 3.2

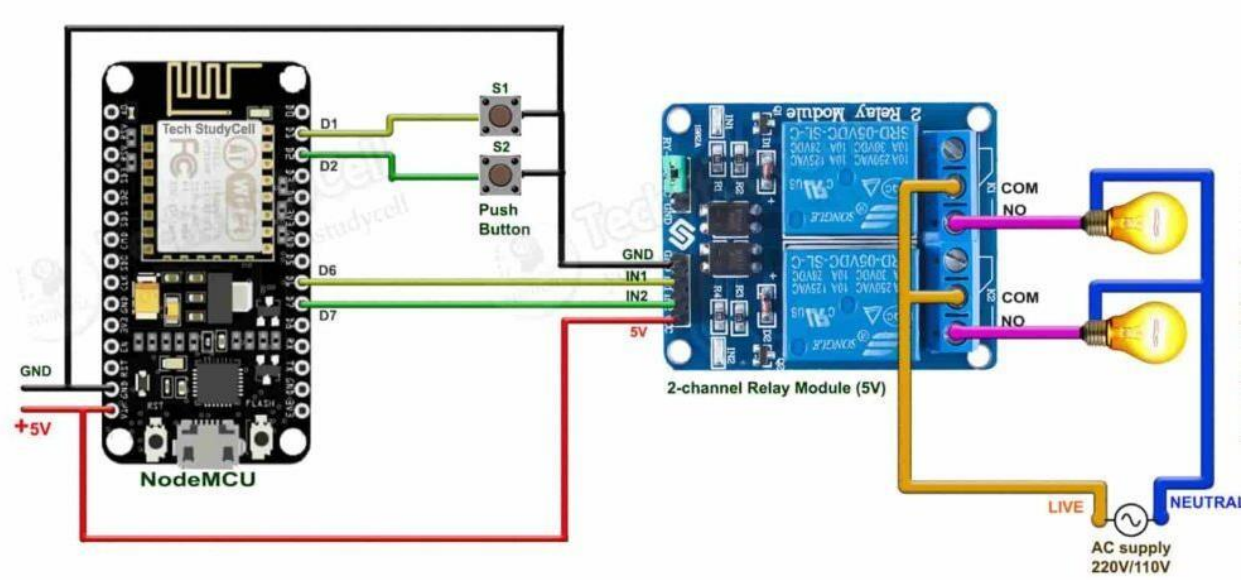


Figure 3.2: Sequence Diagram

Figure 3.2.3: The flowchart for voice-controlled smart home automation with ESP8266 begins communicates with the Blynk app for manual control and triggers events via IFTTT based on predefined conditions. Finally, the system responds to voice commands from Google Assistant, completing the loop for a seamless and intuitive smart home automation experience claim in figure 3.3.

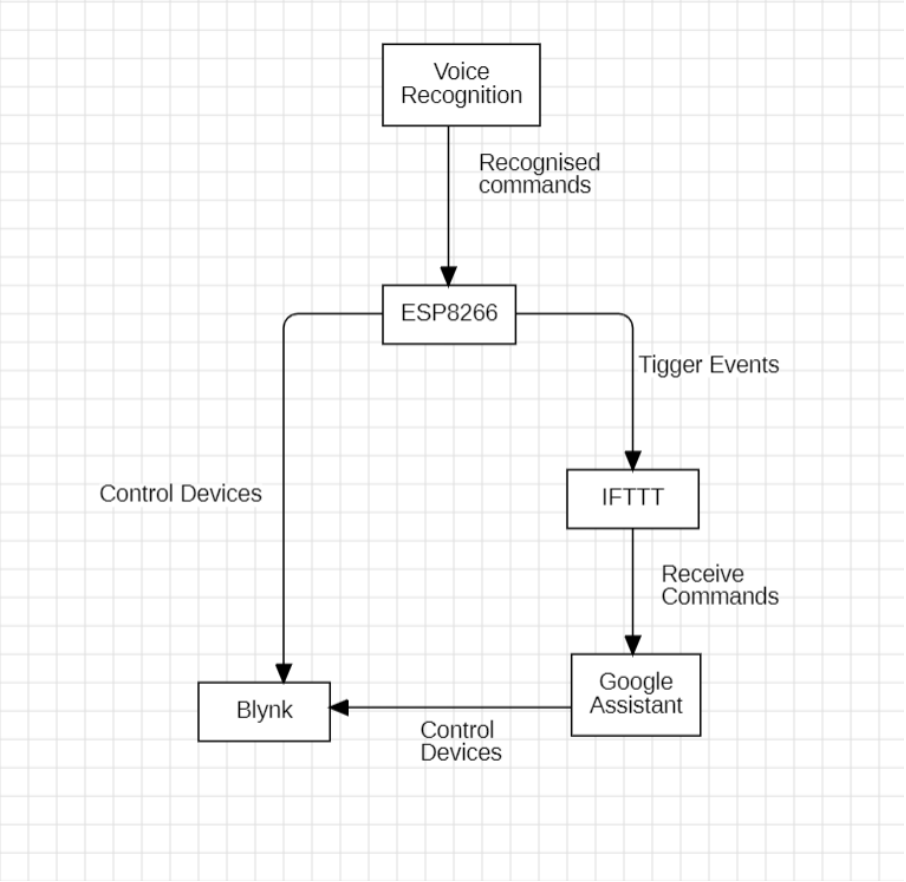


Figure 3.3: Flow Chart

CHAPTER 4: INTERNET OF THINGS

4. INTERNET OF THINGS

4.1. Internet of Things

The Internet of Things (IoT) refers to the interconnected network embedded with sensors, software, and network connectivity, allowing them to collect and exchange data. This integration of the physical world with the digital realm enables these devices to communicate and collaborate, equipped with various communication technologies, sensors . collected data is transmitted to a central system or the cloud for analysis. IoT facilitates automation by enabling devices to respond to specific conditions without human intervention. IOT holds the potential to enhance efficiency, improve decision-making, and create new opportunities across different sectors.

4.2. IFTTT

IFTTT, short for "If This Then That," is a web-based service that allows users to create automated workflows and connections between various online platforms and smart devices. The platform operates on the principle of conditional statements, known as applets, which define triggers and corresponding actions. In the context of voice-controlled smart home automation with ESP8266, IFTTT serves as a pivotal link between the NodeMCU (ESP8266), Blynk application, and external services like Google Assistant.

Users can create applets on the IFTTT platform where specific events or triggers (the "If This" part) initiate predefined actions (the "Then That" part). For example, an applet could be configured to trigger a Blynk event on the NodeMCU when a voice command is detected by Google Assistant. This allows for seamless integration between voice commands and device control, creating a cohesive ecosystem for smart home automation. IFTTT provides a user-friendly interface and supports a wide array of services, enabling users to tailor automation scenarios according to their preferences and the capabilities of connected devices**.**

4.3. Google Assistant

Google Assistant is a virtual voice-controlled personal assistant developed by Google, accessible on various devices such as smartphones, smart speakers, and smart home devices. Leveraging natural language processing and machine learning, Google Assistant interprets voice commands and responds with information, performs tasks, or controls connected devices. As part of the broader Google ecosystem, it seamlessly integrates with services like Google Calendar, Gmail, and Google Home, providing users with a unified and intuitive interface for managing their digital lives. Google Assistant is continuously evolving, offering expanded language support, third-party app integrations, and enhanced capabilities, making it a versatile and powerful tool for hands-free interactions and smart home control.

CHAPTER 5: IMPLEMENTATION

5. IMPLEMENTATION

REQUIREMENTS

Hardware

- ESP8266 NodeMCU
- Relay Module (for controlling the light)
- Jumper wires
- Breadboard
- Google Assistant Account
- IFTTT Account

Software

- Arduino IDE
- Blynk App (iOS/Android)
- Blynk Library for Arduino

5.1. ESP8266 NodeMCU

The ESP8266 NodeMCU is a powerful and versatile microcontroller that has significantly impacted the field of Internet of Things (IoT) development. NodeMCU is based on the ESP8266 WiFi module, which integrates a microcontroller with built-in WiFi capabilities. This compact and cost-effective device It comes with an integrated USB-to-serial converter, eliminating the need for external programming tools The heart of the ESP8266 NodeMCU is the ESP8266 chip, which operates on a 32-bit microcontroller architecture Programming the ESP8266 NodeMCU is typically done using the Arduino IDE The ESP8266 NodeMCU's versatility is further enhanced by the abundance of GPIO (General Purpose Input/Output) pins, which allow for the connection of various sensors, actuators, and other peripherals the ESP8266 NodeMCU stands as a testament to the impact of accessible and versatile technology in driving innovation..

Figure 5.1: ESP8266 NodeMCU

5.2. RELAY MODULE

Relay module is an electronic device that is used to control high-power electrical devices or circuits using low-power signals. It acts as a switch that can be controlled by an external signal, allowing it to open or close the circuit it is connected to. Relays are commonly used to

control a high-power load with a low-power signal These are pre-packaged relay units that often include additional components such as driver circuits, protection diodes, and sometimes even microcontrollers. they simplify the wiring and control of relays.

Figure 5.2: Relay Module

5.3. JUMPER WIRES

Jumper wires are simple, flexible wires with connectors at each end that are used to create electrical connections between components on a breadboard, circuit board, or other electronic devices. They play a crucial role in prototyping, experimenting, and building electronic circuits . types of wires are Male to Male , Male to Female, Female to Female.

Figure 5.3: Jumper Wire

5.4. BREAD BOARD

A breadboard, also known as a protoboard is a versatile tool used in electronics for prototyping and testing circuits without the need for soldering. It consists of a rectangular board with a grid of interconnected holes, allowing electronic components to be easily inserted and connected using jumper wires. The board typically features power rails on the sides for convenient access to positive and negative power supplies.Their reusability makes them a valuable resource for trying out various electronic configurations before committing to permanent soldered connections. Breadboards come in different sizes, essential component.

Figure 5.4: Bread Board

5.6 GOOGLE ASSISTANT

Google Assistant an artificial intelligence service created by the Google. Through this, we can get the service using our voice. Also, we can see this service in all smart devices. That is, smartphones, smartwatches, Tabs and smart home devices.

5.6 IFTTT

IFTTT is a service designed to respond to events. That is, an app with a large amount of software that connects devices and services together. Also, this IFTTT application works closely with various service providers. We can use this application mainly for various services like home automation and posting the same content on several social media platform. For that, we can use the “applets” in this application.

5.7 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a user-friendly software platform designed for programming and developing applications for Arduino microcontrollers. It provides a simple and accessible interface for writing, compiling, and uploading code to Arduino boards. With a streamlined code editor, library management, and a serial monitor for debugging,It supports the Arduino programming language, which is based on a simplified version of C and C++. The IDE plays a crucial role in the Arduino ecosystem, enabling users to easily create and upload code

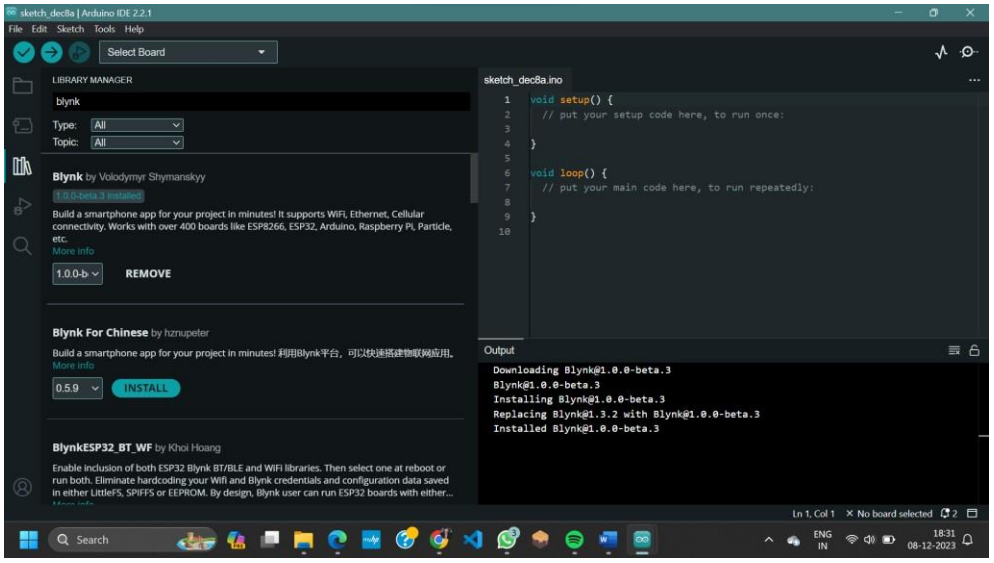
5.8 BLYNK APP

Blynk is a popular Internet of Things (IoT) platform that enables users to create custom interfaces for controlling and monitoring hardware devices. The Blynk app serves as the user interface, allowing individuals to design interactive dashboards effortlessly. With a drag-and-drop interface, users can add various widgets such as buttons, sliders, graphs, and displays to create a personalized control panel for their IoT projects. Blynk supports a wide range of hardware platforms, including NodeMCU ESP8266, Arduino, Raspberry Pi, and others, making it versatile for different IoT applications. Users can remotely manage connected devices, receive real-time data, and even integrate third-party services, contributing to the platform's flexibility and ease of use in creating smart home automation solutions.

PROCEDURE

1. Install Required Libraries: -

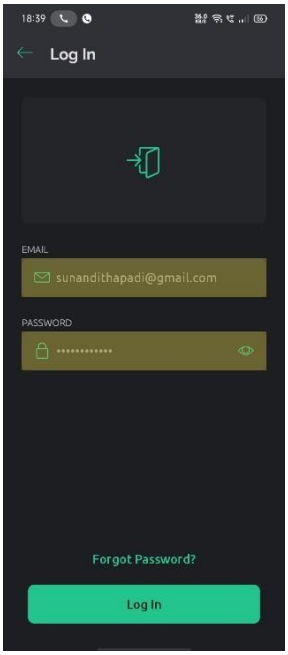
Install the Blynk library in Arduino IDE



2. Create a Blynk Account: -

Download the Blynk app on your smartphone.

Create an account and log in.

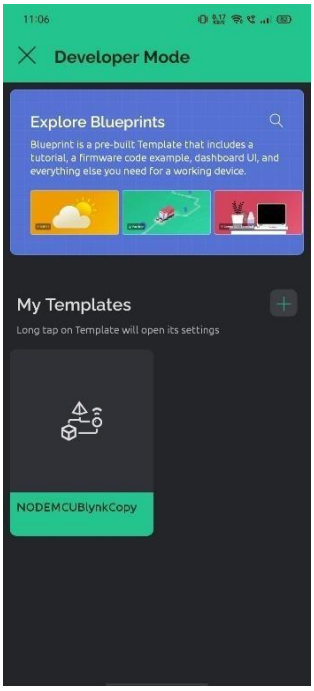


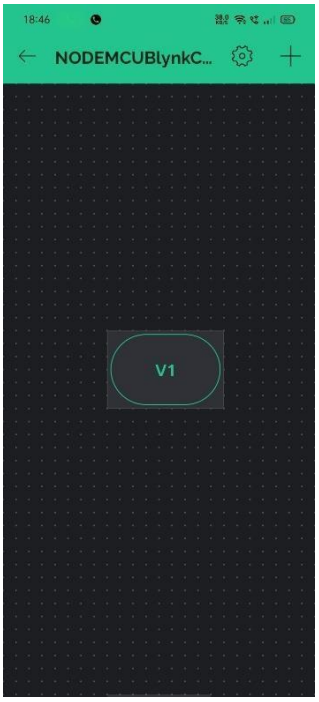
3. Create a New Project on Blynk: -

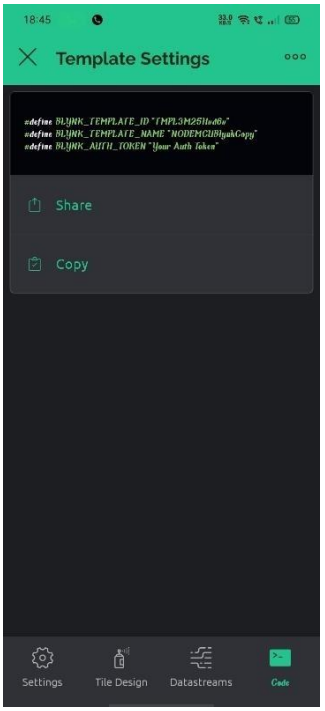
Create a new project in the Blynk app.

Add a Button widget (for manual control) and a Terminal widget (for debugging).

Note down the authentication token.







4. Connect Relay Module to ESP8266: -
Connect VCC of the relay to 3.3V on NodeMCU.
Connect GND of the relay to GND on NodeMCU.
Connect IN1 of the relay to D2 on NodeMCU.

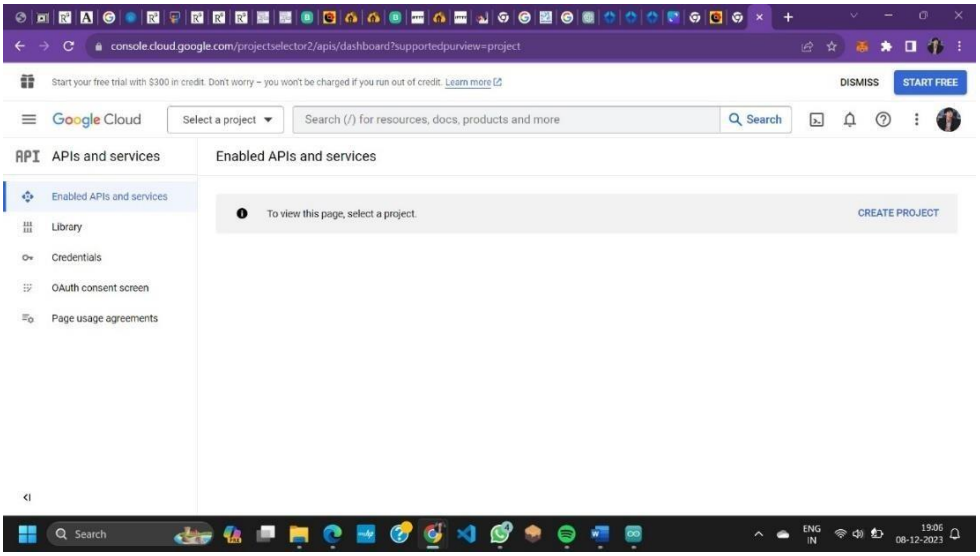
5. Code:

```
#include #include char auth[] = "YourAuthCode";
char ssid[] = "YourWiFiSSID";
char pass[] = "YourWiFiPassword";
 void setup() {
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
 }
void loop() {
Blynk.run();
}
BLYNK_WRITE(V1) {
 int value = param.asInt();
digitalWrite(D2, value);
 }
```

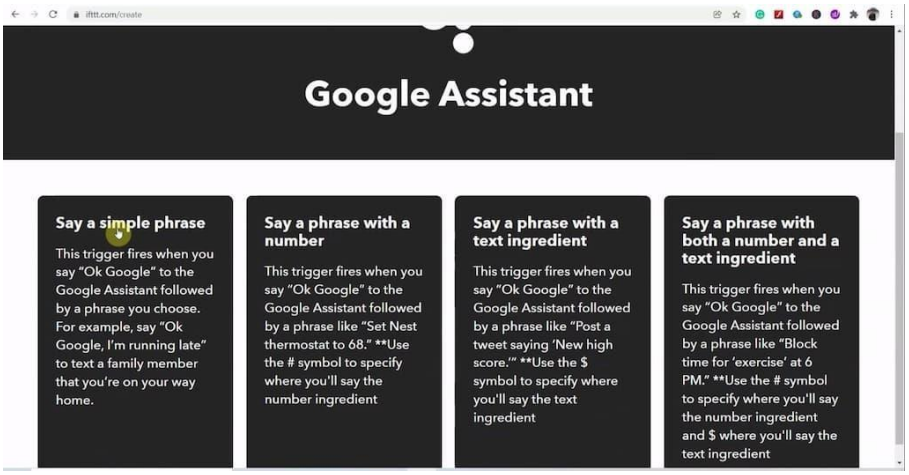
- Replace "YourWiFiSSID" and "YourWiFiPassword" with your Wi-Fi credentials.

6. Create a Google Assistant App: -
Go to the [Google Cloud Console](https://console.cloud.google.com/). –
Create a new project. - Enable the "Google Assistant API" and create credentials.

Note down the client ID and client secret.

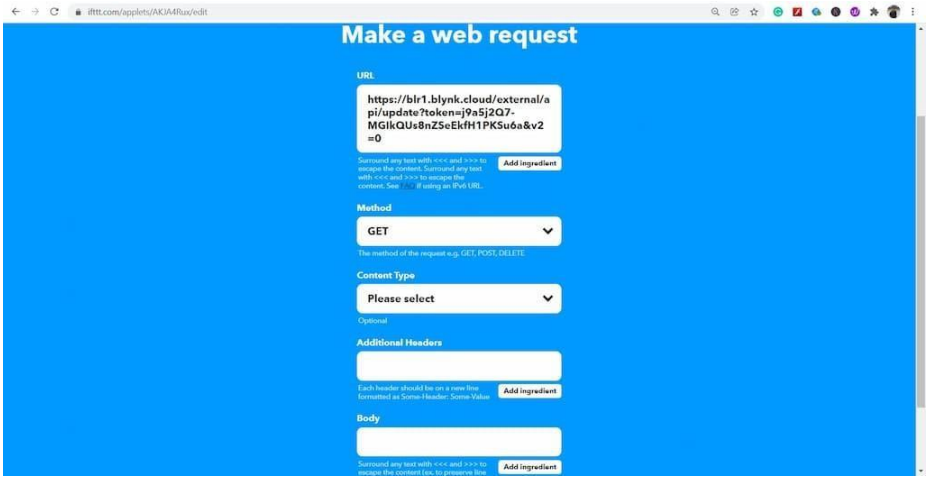


7. Create an IFTTT Account and Applet:
    - Go to [IFTTT](https://ifttt.com/) and create an account.
    - Create a new applet: - If "Google Assistant" -> "Say a simple phrase".
    - Then "Webhooks" -> "Make a web request":
    - URL: `http://blynk-cloud.com/YOUR_AUTH_TOKEN/update/V1`
    - Method: `POST`
    - Content Type: `application/json`
    - Body: `{"value1": "{{TextField}}"}`
    - Connect IFTTT to Blynk

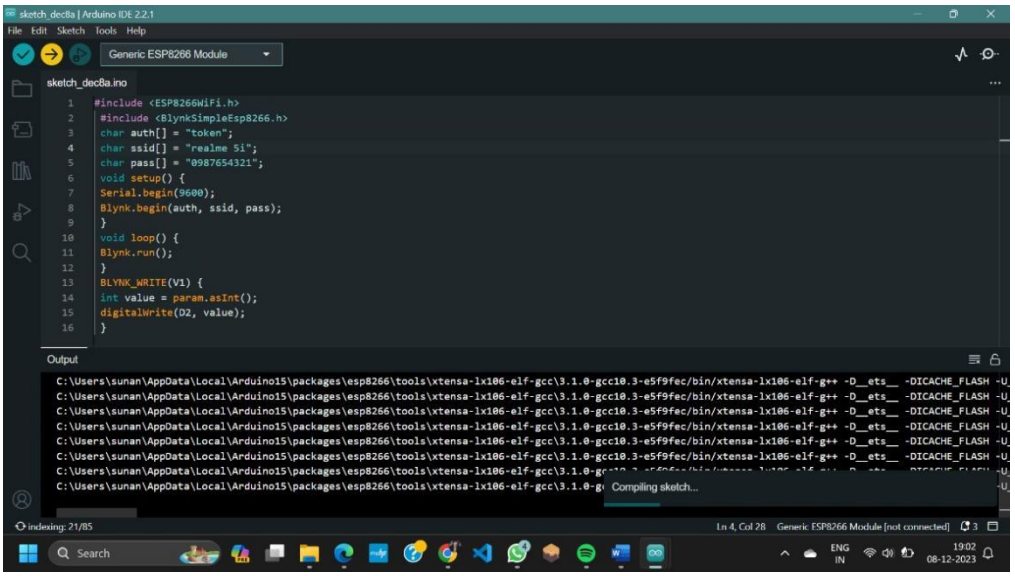



Say a simple phrase
What do you want to say?
Turn off room light
What's another way to say it? (optional)
room light off
And another way? (optional)
What do you want the Assistant to say in response?
Turning off the room light


Webhooks
Make a web request
This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.


Make a web request
URL
https://blr1.blynk.cloud/external/api/update?token=j9a5j2Q7-MGlkQUs8nZ5eEkfH1PKSu6a&v2=0
Add ingredient
Method
GET
Content Type
Please select
Additional Headers
Add ingredient
Body
Add ingredient

8. Upload Code to ESP8266: -
   Connect your NodeMCU to your computer.
   Select the correct board and port in Arduino IDE.
   Click the upload button.



CHAPTER 6: RESULT & DISCUSSIONS

CHAPTER 7: CONCLUSION & FUTURE ENHANCEMENT

7. CONCLUSION & FUTURE ENHANCEMENT

In conclusion, the development of a voice-controlled smart home automation system with ESP8266 has demonstrated the potential for creating an intuitive and convenient interface for users to interact with their home devices. Integrating voice recognition technology with the ESP8266 microcontroller, along with platforms like Blynk and IFTTT, offers a seamless and hands-free experience. Users can effortlessly control various appliances and devices through simple voice commands, enhancing the overall accessibility and efficiency of home automation.

For future enhancements, the system could benefit from advancements in natural language processing (NLP) and machine learning to improve the accuracy and versatility of voice commands. Implementing more sophisticated voice recognition algorithms could enable the system to understand complex instructions and adapt to individual user preferences. Additionally, expanding device compatibility, integrating additional sensors for environmental monitoring, and enhancing security features would contribute to a more comprehensive and intelligent smart home ecosystem. As technology evolves, potential integrations with emerging voice-controlled platforms and the continual refinement of user interfaces can further elevate the user experience, making voice-controlled smart home automation with ESP8266 a pivotal part of the evolving smart home landscape.

REFERENCES

[1]J. Pedro, L. Marinho, and T. Moreira, “Automatic vehicle damage detection with images,” 2017.

[2]V. Bhalsing, M. Lende, T. Maramwar, and C. Naikwadi, “CAR DAMAGE DETECTION AND PRICE,” vol. 11, no. 6, pp. 759–766, 2023.

[3]D. Parra and A. E. Gonzalez, “Car Damage Assesment”.

[4]K. He and W. D. Zhu, “Structural damage detection using changes in natural frequencies: Theory and applications,” *J. Phys. Conf. Ser.*, vol. 305, no. 1, 2011, doi: 10.1088/1742-6596/305/1/012054.

[5]R. E. van Ruitenbeek and S. Bhulai, “Convolutional Neural Networks for vehicle damage detection,” *Mach. Learn. with Appl.*, vol. 9, p. 100332, Sep. 2022, doi: 10.1016/j.mlwa.2022.100332.

[6]U. Waqas, N. Akram, S. Kim, D. Lee, and J. Jeon, “Vehicle Damage Classification and Fraudulent Image Detection Including Moiré Effect Using Deep Learning,” in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, IEEE, Aug. 2020, pp. 1–5. doi: 10.1109/CCECE47787.2020.9255806.

[7]R. S. Gaykar, V. Khanaa, and S. D. Joshi, “Detection of Faulty Nodes in Distributed Environment using Machine Learning,” in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, IEEE, Dec. 2021, pp. 228–232. doi: 10.1109/ICAC3N53548.2021.9725478.

[8]R. S. Gaykar, C. Nalini, and S. D. Joshi, “Identification of Straggler Node in Distributed Environment Using Soft Computing Algorithms,” in *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, IEEE, Mar. 2021, pp. 1–5. doi: 10.1109/ESCI50559.2021.9396825.

[9]Q. Zhang, X. Chang, and S. B. Bian, “Vehicle-Damage-Detection Segmentation Algorithm Based on Improved Mask RCNN,” *IEEE Access*, vol. 8, pp. 6997–7004, 2020, doi: 10.1109/ACCESS.2020.2964055.

[10] M. H.-M. Khan, M. Z. Hussein Sk Heerah, and Z. Basgeeth, “Automated Detection of Multi-class Vehicle Exterior Damages using Deep Learning,” in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, IEEE, Oct. 2021, pp. 01–06. doi: 10.1109/ICECCME52200.2021.9590927.

[11] D. Widjojo, E. Setyati, and Y. Kristian, “Integrated Deep Learning System for Car Damage Detection and Classification Using Deep Transfer Learning,” in *2022 IEEE 8th Information Technology International Seminar (ITIS)*, IEEE, Oct. 2022, pp. 21–26. doi: 10.1109/ITIS57155.2022.10010292.

[12] P. M. Kyu and K. Woraratpanya, “Car Damage Detection and Classification,” in *Proceedings of the 11th International Conference on Advances in Information Technology*, New York, NY, USA: ACM, Jul. 2020, pp. 1–6. doi: 10.1145/3406601.3406651.

[13] C. K. Gomathy and S. Rajalakshmi, “A software quality metric performance of

professional management in service oriented architecture," in *Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014*, IEEE, Jul. 2014, pp. 41–47. doi: 10.1109/ICCTET.2014.6966260.

[14] Mallikarjuna B and Arun Kumar K L, "Vehicle Damage Detection and Classification Using Image Processing," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 568–574, Jun. 2022, doi: 10.48175/IJARSCT-5414.

[15] M. M. Alam and M. T. Islam, "Machine learning approach of automatic identification and counting of blood cells," *Healthc. Technol. Lett.*, vol. 6, no. 4, pp. 103–108, Aug. 2019, doi: 10.1049/htl.2018.5098.

[16] J. Yang, L. Ji, X. Geng, X. Yang, and Y. Zhao, "Building detection in high spatial resolution remote sensing imagery with the U-Rotation Detection Network," *Int. J. Remote Sens.*, vol. 40, no. 15, pp. 6036–6058, Aug. 2019, doi: 10.1080/01431161.2019.1587200.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[18] R. Roy and K. T. George, "Detecting insurance claims fraud using machine learning techniques," in *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*, IEEE, Apr. 2017, pp. 1–6. doi: 10.1109/ICCPCT.2017.8074258.

[19] M. Koch, H. Wang, and T. Back, "Machine Learning for Predicting the Damaged Parts of a Low Speed Vehicle Crash," in *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, IEEE, Sep. 2018, pp. 179–184. doi: 10.1109/ICDIM.2018.8846974.

[20] G. Kowshalya and M. Nandhini, "Predicting Fraudulent Claims in Automobile Insurance," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, IEEE, Apr. 2018, pp. 1338–1343. doi: 10.1109/ICICCT.2018.8473034.

[21] J. Wu, J. Wang, and Y. Liu, "Design and Research of Insurance Survey Claims System Based on Big Data Analysis," in *2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, IEEE, Sep. 2019, pp. 211–214. doi: 10.1109/ICVRIS.2019.00059.

[22] G. Guastadisegni *et al.*, "Ride analysis tools for passenger cars: objective and subjective evaluation techniques and correlation processes – a review," *Veh. Syst. Dyn.*, pp. 1–27, Oct. 2023, doi: 10.1080/00423114.2023.2259024.

[23] R. C. Jaiswal and S. D. Lokhande, "Machine learning based internet traffic recognition with statistical approach," in *2013 Annual IEEE India Conference (INDICON)*, IEEE, Dec. 2013, pp. 1–6. doi: 10.1109/INDCON.2013.6726074.

[24] M. Böhm and T. Scheider, "Requirements for vehicle positioning and map referencing for co-operative Systems on motorways," *ENC-GNSS 2008 - Eur. Navig. Conf.*, 2008.

[25] Y. K. Yik, N. E. Alias, Y. Yusof, and S. Isaak, "A Real-time Pothole Detection Based on Deep Learning Approach," *J. Phys. Conf. Ser.*, vol. 1828, no. 1, p. 012001, Feb. 2021, doi: 10.1088/1742-6596/1828/1/012001.

[26] O. A. Egaji, G. Evans, M. G. Griffiths, and G. Islas, “Real-time machine learning-based approach for car detection,” *Expert Syst. Appl.*, vol. 184, p. 115562, Dec. 2021, doi: 10.1016/j.eswa.2021.115562.

[27] K. Bansal, K. Mittal, G. Ahuja, A. Singh, and S. S. Gill, “<scp>DeepBus</scp> : Machine learning based real time car detection system for smart transportation using <scp>IoT</scp>,” *Internet Technol. Lett.*, vol. 3, no. 3, May 2020, doi: 10.1002/itl2.156.

[28] I. D. Pratama, H. Mahmudah, and R. W. Sudibyo, “Design and Implementation of Real-time Car Detection using Convolutional Neural Network for IoT Smart Environment,” in *2021 International Electronics Symposium (IES)*, IEEE, Sep. 2021, pp. 675–679. doi: 10.1109/IES53407.2021.9594038.

[29] D. Dongre, R. Jangam, A. Gosavi, M. Sonekar, R. Kumbhalkar, and A. P. Nanotkar, “Car damaged detection and Navigation System Based on IoT,” in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, Jun. 2021, pp. 395–398. doi: 10.1109/ICOEI51242.2021.9453010.

[30] M. H. Asad, S. Khaliq, M. H. Yousaf, M. O. Ullah, and A. Ahmad, “Car Detection Using Deep Learning: A Real-Time and AI-on-the-Edge Perspective,” *Adv. Civ. Eng.*, vol. 2022, pp. 1–13, Apr. 2022, doi: 10.1155/2022/9221211.