

BASAVARAJESWARI GROUP OF INSTITUTIONS



## **BALLARI INSTITUTE OF TECHNOLOGY AND MANAGEMENT**



NACC Accredited Institution  
(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to  
Visvesvaraya Technological University, Belagavi)  
"Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur,  
Ballari-583 104 (Karnataka) (India)  
Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197



### **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Mobile Application Development (18CSMP68)**

Mini Project Synopsis

On

**“Home Automation Controller”**

*Submitted By*

**B SAI KIRAN REDDY**

**3BR20CS022**

**B SHRAVANI**

**3BR20CS023**

**Under the Guidance of**

**Mrs. Sheetal J**

**Ms. Madhuri A**

**Dept of CS&E,**

**BITM, Ballari.**



**Visvesvaraya Technological University**

**Belagavi, Karnataka**

**2022-2023**

## **ABSTRACT**

The Android App for ESP8266 Module Control and Monitoring is a mobile application designed to simplify the control and monitoring of an ESP8266 module from an Android device. By establishing a connection between the app and the module using the provided IP address, users can remotely control the state of the GPIO pins on the module and receive real-time feedback on their status. The app leverages the capabilities of Android Studio and the Java programming language to deliver a user-friendly interface, seamless communication with the ESP8266 module, and efficient management of objectives stored on the module.

The Android App for ESP8266 Module Control and Monitoring is developed to offer an intuitive mobile interface specifically tailored for interacting with ESP8266 modules. By eliminating the reliance on web browsers, the app simplifies the connection process and enhances user experience. Users can easily input the IP address of their ESP8266 module through the app, establish a connection, and start controlling the GPIO pins remotely using buttons within the app's interface.

The app goes beyond basic control functionality by enabling users to send custom commands to the module. This feature empowers users to execute specific actions or trigger events on the module, expanding its capabilities and allowing for more complex operations. Additionally, users can monitor the real-time status of the GPIO pins through the app, providing immediate feedback on the state of connected devices or systems.

Moreover, the app incorporates objective management functionality, enabling users to define and manage objectives stored on the ESP8266 module. By setting objectives and configuring the module's behavior accordingly, users can automate processes or create predefined scenarios that can be triggered remotely using the app. This feature adds a layer of flexibility and customization to the app's functionality, catering to a diverse range of use cases and requirements.

## **1. INTRODUCTION**

The Android App for ESP8266 Module Control and Monitoring is a mobile application developed using Android Studio in Java. It is designed to establish a connection between an Android device and an ESP8266 module, enabling users to control and monitor the GPIO (General Purpose Input/Output) pins of the module remotely.

The app provides a user-friendly and intuitive interface that allows users to input the IP address of the ESP8266 module. Once connected, users can interact with the GPIO pins through the app's controls, such as buttons, to toggle their states. This functionality enables users to remotely control devices or systems connected to the ESP8266 module, providing convenience and flexibility.

In addition to controlling the GPIO pins, the app allows users to send commands to the ESP8266 module. This feature enables users to execute specific actions or trigger events on the module, extending its functionality beyond basic pin control. By sending custom commands, users can interact with other components or devices connected to the module and achieve more complex operations.

Furthermore, the app enables users to retrieve the current state of the GPIO pins. This functionality provides real-time information about the status of the connected devices or systems. Users can easily monitor the states of the pins and receive feedback on the app's interface, keeping them informed about the current state of their remote setup.

The app is built using Android Studio, the official IDE for Android development. Android Studio provides a wide range of tools and resources to facilitate the development process, including a visual UI designer, code editing features, debugging tools, and emulator support. With these capabilities, developers can create a user-friendly interface, implement the necessary functionality, and test the app on virtual or physical Android devices.

## **2. PROBLEM STATEMENT**

To design and develop a remote control and monitoring of GPIO pins on an ESP8266 module often requires users to interact with the module through a web browser, which can be inconvenient and less intuitive. Additionally, there is a lack of dedicated mobile applications that provide a seamless and user-friendly experience for controlling and monitoring ESP8266 module. Therefore, there is a need for an Android app that enables users to easily connect to an ESP8266 module, control GPIO pins, retrieve their status, and manage objectives stored on the module.

### **3. OBJECTIVES**

- To develop an Android app that can connect to an ESP8266 module. Allow user to input the IP address of the ESP8266 module.
- To display buttons for each GPIO pin on the ESP8266 to control and monitor their state (on/off).
- To implement functionality to send commands to the ESP8266 module to control the GPIO pins. Implement functionality to receive and display the current state of the GPIO pins from the ESP8266 module.

## **4. HARDWARE & SOFTWARE REQUIREMENTS**

### Hardware Requirements:

1. Android device (smartphone, tablet) with minimum specifications:
  - Processor: Dual-core 1.2 GHz or higher.
  - RAM: 2GB or higher.
  - Storage: 50 MB free space or more
  - Display: Minimum resolution of 480x800 pixels.
2. ESP8266 Development Board.

### Software Requirements:

1. Android Operating System: Android 5.0 (Lollipop) or higher
2. Android Development Environment:
  - Android Studio: Version 4.0 or higher
  - Java Development Kit (JDK): Version 8 or higher
3. Android Software Development Kit (SDK):
  - Minimum SDK Version: API level 21 (Android 5.0 Lollipop)
  - Target SDK Version: Latest stable Android version.
4. HTTP Communication: Use HTTP communication (GET and POST requests) to send commands and retrieve the state of GPIO pins on the ESP8266 module.
5. Access Point (AP) Mode: Configure the ESP8266 module to act as an access point. If necessary, to allow the Android app to connect to it directly.