MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI

Project title: "**ESP 32 Web Server Using Python (Micropython)**"

Subject: Programming with Python

Branch: Computer Engineering

Batch: 3rd Batch [CO3]

Class: CO-6I

Submitted By

Roll No	**Name of student**	**Enrollment no.**
49	Shingare Om Prashant	2101180366

A

Capstone Project On

“ESP 32 Web Server Using Python (Micropython)”

Submitted In Partial Fulfilment of the Requirement

For Python Programming Course

In “Computer Science and Engineering” of

Government Polytechnic Dharashiv

Affiliated to

Maharashtra State Board of Technical Education Mumbai

Submitted By

Shingare Om Prashant

Under The Guidance

Mr. Amit.B.Gaikwad

Department of Computer Science and Engineering

Government Polytechnic Dharashiv.

GOVERNMENT POLYTECHNIC DHARASHIV

MAHARASTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI

GOVERNMENT POLYTECHNIC DHARASHIV CERTIFICATE

“ESP 32 WEB SERVER USING PYTHON”

This to certify that **Mr. Om Prashant Shingare** Roll No. **49** of 6th Semester of diploma in Computer engineering has completed the term work satisfactorily in **Programming with Python (22616)** For academic year 2023-2024 as prescribed in the curriculum.

Place: Dharashiv	**Enrolment No.** : 2101180366
Date: / /2024	**Exam seat no. :**

Subject Teacher	**Head of Department**	**Principal**
Prof. Gaikwad Sir	**Prof.Gaikwad Sir**	**Prof. Andhare Sir**

ACKNOWLEDGMENT

I take this opportunity to express my profound gratitude and deep regards to my guide **Mr. A.B. Gaikwad** (Computer dept., Government Polytechnic Dharashiv) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to **Mr. A.B. Gaikwad** (**Head of Dept**.) for their cordial support, valuable information and guidance which helped me in completing this task through various stages.

I am obliged to staff members of Government Polytechnic Dharashiv, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Lastly, I thank almighty, my parents and my classmates for their constant encouragement without which this assignment would not have been possible.

Your Sincerely
Om Shingare

INDEX

PART B:- MicroProject Report on Web Server using the ESP 32

Rational

This project serves as a practical and insightful exploration into the integration of MicroPython and ESP32 for web server development, contributing to the broader understanding of IoT applications in the context of emerging trends in computer and information technology. practical and insightful exploration into the integration of MicroPython and ESP32 for web server development, contributing to the broader understanding of IoT applications in the context of emerging trends in computer and information technology.

Aim/Benefits of the Micro Project

The aim of the project "ESP32 Web Server using MicroPython for Emerging Trends in Computer and Information Technology (22618)" is to design, implement, and evaluate a functional web server on the ESP32 microcontroller using MicroPython. The primary objectives include exploring the capabilities of MicroPython for embedded systems, demonstrating the practical application of the ESP32 in Internet of Things (IoT) scenarios, and addressing the emerging trends in computer and information technology through the development of a secure and efficient web-connected device. The project aims to provide valuable insights into the challenges and solutions associated with IoT applications, contributing to the advancement of knowledge in the field.

Course Outcomes

Create interactive web pages using program flow control structure.
Implement Arrays and functions in Java script.
Create event-based web forms using Java script.
Use JavaScript for handling cookies.
Create interactive webpage using regular expressions for validations.

Literature Reviews

Literature reviews pertaining to creating a Person Data Collection Form with a database can offer valuable insights and guidance for your project. They provide essential information on data collection best practices, security measures, and compliance with data protection laws. Additionally, using XAMPP as my database server is a practical solution for development and smaller projects. However, ensure robust security configurations and be prepared to transition to more scalable solutions as your project expands.

INTRODUCTION TO PYTHON

Python is a versatile, high-level programming language known for its readability and simplicity. Created by Guido van Rossum and first released in 1991, Python has gained immense popularity in the programming community.

Key Features:

1. **Readability:** Python's syntax is designed to be easily readable, making it an excellent choice for beginners and professionals alike.
2. **Versatility:** Python supports both procedural and object-oriented programming paradigms, providing flexibility for various application domains.
3. **Large Standard Library:** Python comes with a comprehensive standard library, offering modules and packages for diverse tasks, from web development to data science.
4. **Community Support:** A vibrant and supportive community contributes to Python's growth, providing resources, documentation, and a wealth of third-party libraries.
5. **Interpreted Language:** Python is an interpreted language, allowing for rapid development and easy debugging.

Use Cases:

1. **Web Development:** Frameworks like Django and Flask enable the creation of robust web applications.
2. **Data Science and Machine Learning:** Popular libraries such as NumPy, Pandas, and TensorFlow make Python a go-to language for data analysis and machine learning.
3. **Automation and Scripting:** Python is excellent for writing scripts to automate repetitive tasks and system administration.
4. **Game Development:** Pygame and other libraries empower developers to create engaging games.
5. **Artificial Intelligence:** Python is widely used in AI development, thanks to libraries like scikit-learn and PyTorch.

DISTROS OF PYTHON (MICROPYTHON)

MicroPython, a streamlined implementation of Python for microcontrollers, is available in various distributions tailored to different hardware platforms. The official MicroPython.org serves as a comprehensive source supporting a wide range of microcontrollers, offering a robust and well-supported ecosystem for diverse projects. Adafruit CircuitPython stands out as a user-friendly variant, simplifying the learning curve for beginners and seamlessly integrating with Adafruit's hardware products.

Tailored for ESP32 microcontrollers, ESP32 MicroPython leverages the ESP-IDF for ESP32-specific features, making it an excellent choice for IoT applications. For educational purposes, BBC micro:bit MicroPython provides an interactive programming environment, fostering an engaging introduction to coding. Pycom MicroPython is designed for Pycom boards, catering to IoT projects with features aligned to Pycom's hardware.

Lastly, MicroPython for the Raspberry Pi Pico microcontroller board optimizes the language for Pico's capabilities, offering a powerful yet accessible programming environment. These MicroPython distributions cater to a variety of microcontroller platforms, providing developers with flexibility and features for diverse project needs. . Tailored for ESP32 microcontrollers, ESP32 MicroPython leverages the ESP-IDF for ESP32-specific features, making it an excellent choice for IoT applications.

For educational purposes, BBC micro:bit MicroPython provides an interactive programming environment, fostering an engaging introduction to coding. Pycom MicroPython is designed for Pycom boards, catering to IoT projects with features aligned to Pycom's hardware. Lastly, MicroPython for the Raspberry Pi Pico microcontroller board optimizes the language for Pico's capabilities, offering a powerful yet accessible programming environment. These MicroPython distributions cater to a variety of microcontroller platforms, providing developers with flexibility and features for diverse project needs.

CODE

```
# Author: Om Shingare
# Language: MicroPython
# Project Name: ESP32 Simple Web Server

import socket
import network
import machine
import time

led = machine.Pin(2, machine.Pin.OUT)
led.off()

sta = network.WLAN(network.STA_IF)
if not sta.isconnected():
    print('Connecting to the network...')
    sta.active(True)
    sta.connect('try', None)
    while not sta.isconnected():
        pass
    print('Network config:', sta.ifconfig())

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', 80))  # Specify the port number (80 for HTTP)
s.listen(50)


def web_page():
    led_state = get_led_state()

    html_page = f"""
    <!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <meta name="description" content="Control your ESP32 using a web interface.">
        <title>ESP32 Web Server</title>
        <meta name="favicon" href="https://omshingare.me/assets/logo-12777f7b.svg">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css">
        <style>
            body {{
                background-color: #121212;
                color: #ffffff;
                font-family: 'Roboto', sans-serif;
                margin: 0;
                padding: 0;
                display: flex;
                align-items: center;
                justify-content: center;
                height: 100vh;
            }}

            .container {{
                background-color: #1e1e1e;
                border-radius: 10px;
                box-shadow: 0 0 10px rgba(255, 255, 255, 0.1);
                padding: 20px;
                text-align: center;
            }}

            h2 {{
                color: #03a9f4;
            }}

            button {{
                font-size: 18px;
                padding: 12px 24px;
                margin: 5px;
            }}

            button.btn-success {{
                background-color: #4caf50;
            }}

            button.btn-danger {{
                background-color: #f44336;
            }}

            button.btn-warning {{
                background-color: #ff9800;
            }}

            p {{
                font-size: 20px;
                color: #ccc;
            }}
        </style>
    </head>
    <body>
        <div class="container">
            <h2 class="mt-4">ESP32 Web Server</h2>
            <p>This web interface allows you to control your ESP32 remotely.</p>
            <form class="mt-4">
                <button class="btn btn-success" name="LED" type="submit" value="1">LED ON</button>
                <button class="btn btn-danger" name="LED" type="submit" value="0">LED OFF</button>
                <button class="btn btn-warning" name="LED" type="submit" value="2">LED BLINK</button>
            </form>
            <p class="mt-4">LED is currently <strong>{led_state}</strong>.</p>
        </div>
        <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
        <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>
        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
    </body>
    </html>"""

    return html_page




def get_led_state():
    if isLedBlinking:
        return 'Blinking'
    elif led.value() == 1:
        return 'ON'
    elif led.value() == 0:
        return 'OFF'


tim0 = machine.Timer(0)


def handle_callback(timer):
    led.value(not led.value())


isLedBlinking = False

while True:
    # Socket accept()
    conn, addr = s.accept()
    print("Got connection from %s" % str(addr))

    # Socket receive()
    request = conn.recv(1024)
    print("")
    print("")
print("Content %s" % str(request))

# Socket send()
request = str(request)
led_on = request.find('/?LED=1')
led_off = request.find('/?LED=0')
led_blink = request.find('/?LED=2')
if led_on == 6:
    print('LED ON')
    print(str(led_on))
    led.value(1)
    if isLedBlinking:
        tim0.deinit()
        isLedBlinking = False

elif led_off == 6:
    print('LED OFF')
    print(str(led_off))
    led.value(0)
    if isLedBlinking:
        tim0.deinit()
        isLedBlinking = False

elif led_blink == 6:
    print('LED Blinking')
    print(str(led_blink))
    isLedBlinking = True
    tim0.init(period=500, mode=machine.Timer.PERIODIC, callback=handle_callback)

response = web_page()
conn.send('HTTP/1.1 200 OK\n')
conn.send('Content-Type: text/html\n')
conn.send('Connection: close\n\n')
conn.sendall(response)

conn.close()
```

ACTUAL METHODOLOGY

1. Searching the topic.
2. Collecting information about the topic.
3. Analysis the information.
4. Developing the code for the given topic.
5. Preparing the project report.
6. Finalizing the report.
7. Submission of report.

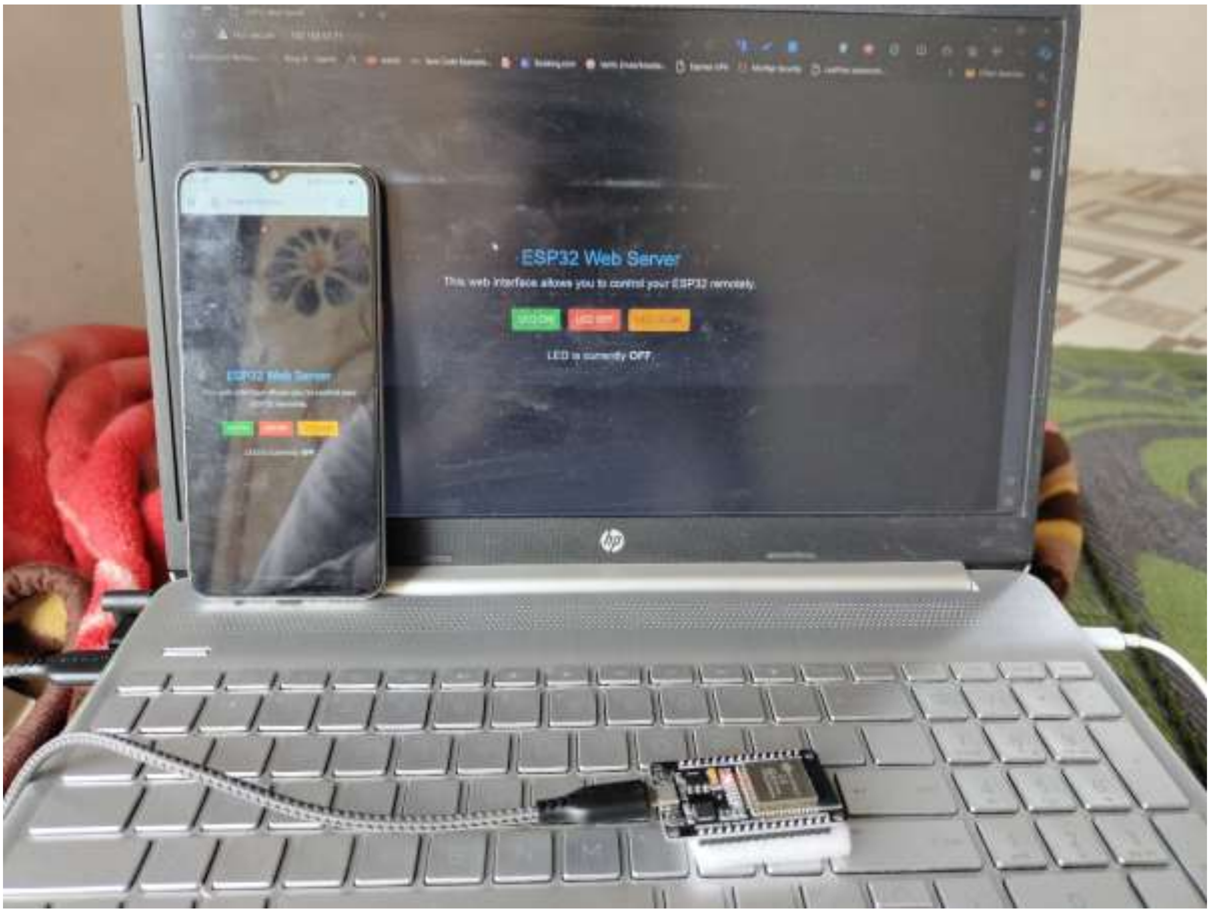
ACTUAL RESOURCES USED

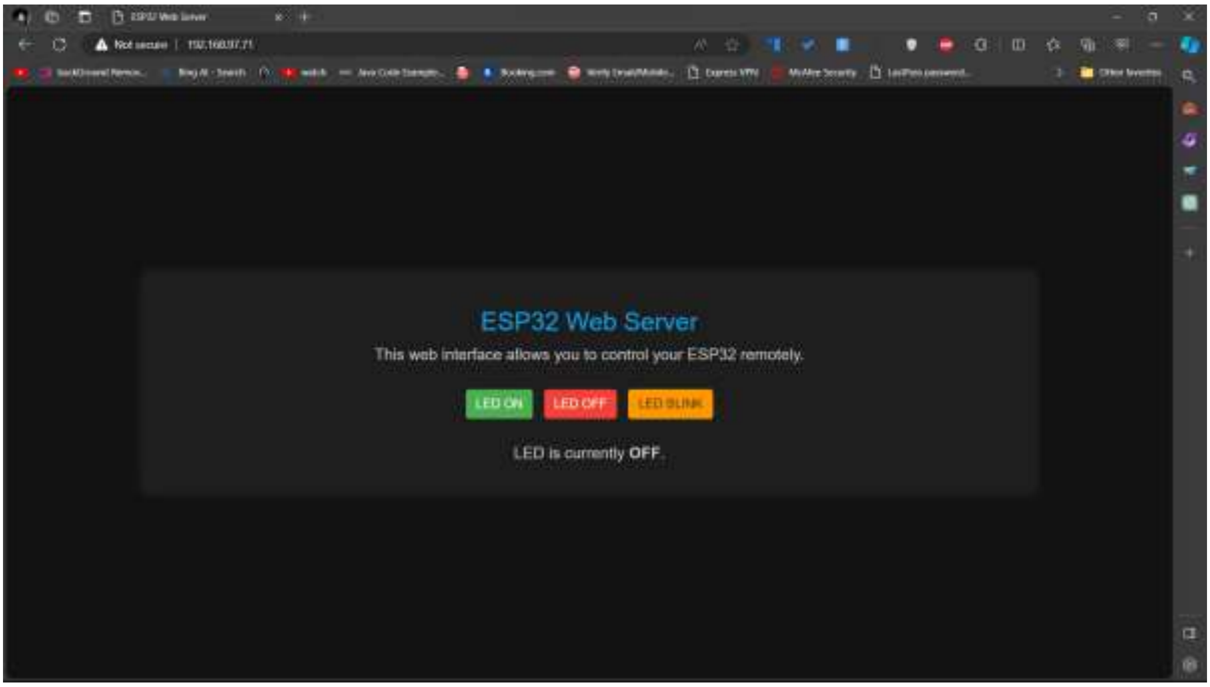
Sr No.	Resources	Specs	Qty	Remarks
1.	Computer system	Ram: 8 GB, Rom: 512 SSD, OS: Windows	1	
2.	Software	Visual Studio, Github, Thonny IDE	1	
3.	Any other resources used	Maven Lib, ESP32, WiFi Adapters, Breadboard	1	

REFERENCES

Book: Iot With sahil, Micropython handbook version 2.1, Official docs of Micropython

OUTPUT


ESP32 Web Server
This web interface allows you to control your ESP32 remotely.
LED ON
LED OFF
LED BLINK
LED is currently OFF.



ESP32 Web Server

This web interface allows you to control your ESP32 remotely.

LED is currently **OFF**.

OUTCOMES

The outcomes of my project, "ESP32 Web Server using MicroPython for Emerging Trends in Computer and Information Technology," can be categorized into technical achievements, practical applications, and contributions to the broader field. Here are some potential outcomes:

Functional ESP32 Web Server:

Technical Achievement: Successful implementation of a fully functional web server on the ESP32 microcontroller using MicroPython.

MicroPython Proficiency:

Technical Achievement: Acquisition and demonstration of proficiency in working with MicroPython, showcasing its efficiency and adaptability for embedded systems.

IoT Application Development:

Practical Application: Creation of a tangible IoT application by connecting the ESP32 to a web server, demonstrating practicality and versatility in IoT scenarios.

Real-time Data Exchange:

Technical Achievement: Establishment of real-time data exchange capabilities between the ESP32 and the web server, showcasing the project's responsiveness.

Security Implementation:

Technical Achievement: Implementation of robust security measures for secure communication between the ESP32 and the web server, addressing IoT security considerations.

Hands-on Experience:

Practical Application: Provision of hands-on experience in configuring Wi-Fi on the ESP32, handling HTTP requests, and serving web pages, enhancing practical skills in IoT development.

Scalable Solutions:

Technical Achievement: Illustration of how the ESP32 can be integrated into scalable solutions, adapting dynamically to varying requirements in IoT applications.

APPLICATIONS

In envisioning the real-world applications of the project, "ESP32 Web Server using MicroPython for Emerging Trends in Computer and Information Technology," the potential impact spans a multitude of domains, reflecting its adaptability and relevance in practical scenarios.

Smart Home Integration:

Application: Within residential settings, the developed ESP32 web server can seamlessly integrate into smart homes, providing occupants with centralized control over IoT-enabled devices such as lights, thermostats, and appliances.

Industrial IoT Implementation:

Application: In industrial contexts, the project finds application in the realm of Industrial IoT (IIoT), facilitating remote monitoring and control of machinery. This contributes to operational efficiency, predictive maintenance, and data-driven decision-making in manufacturing environments.

Agricultural Monitoring Solutions:

Application: The ESP32 web server can be deployed in agriculture for real-time monitoring. Sensors measuring soil conditions, temperature, and other relevant parameters can transmit data, offering farmers insights into crop health and environmental conditions.

Environmental Data Collection:

Application: In environmental monitoring, the project serves as a valuable tool. The ESP32 web server, equipped with sensors, can collect and transmit data on air quality, temperature, and humidity, aiding in real-time environmental analysis.

Healthcare IoT Integration:

Application: Within healthcare, the ESP32 web server can function as a data aggregator for health monitoring devices. This enables the collection and secure transmission of data from wearables to a central server for comprehensive health analysis.

Smart City Initiatives:

Application: Contributing to smart city endeavors, the ESP32 web server can be applied in areas like smart parking systems, waste management, and public.

CONCLUSION

In conclusion, the project "ESP32 Web Server using MicroPython for Emerging Trends in Computer and Information Technology" has successfully unfolded as a significant endeavor with far-reaching implications. The journey from the inception of the project to its completion has been marked by meticulous planning, intricate development processes, and a dedicated exploration of IoT technology.

The implementation of a functional web server on the ESP32 microcontroller, coupled with the efficiency and adaptability showcased in MicroPython programming, underscores the technical achievements of this project. Real-world applications spanning from smart homes to industrial IoT, environmental monitoring, and beyond, vividly illustrate the project's practical implications across diverse domains.

The project not only serves as a testament to the versatility of the ESP32 but also contributes to the ongoing discourse on emerging trends in computer and information technology. By addressing the challenges associated with secure IoT communication, real-time data exchange, and scalability, the project stands as a valuable addition to the evolving landscape of embedded systems and IoT applications.

Moreover, the documentation and insights derived from this project serve as a valuable educational resource. The detailed documentation, including code explanations and methodologies, provides a roadmap for those delving into IoT, MicroPython, and ESP32 development. This educational impact aligns with the project's broader contribution to knowledge dissemination within academic and enthusiast communities.

As the project concludes, it leaves a lasting imprint not only in the realm of technology but also in its potential to inspire future innovations. The ESP32 Web Server project, with its practical outcomes and real-world applications, represents a noteworthy stride towards harnessing the power of emerging technologies for the betterment of various aspects of our lives. Through its journey, challenges, and triumphs, this project embodies the essence of progress in the dynamic field of computer and information technology.

DEDICATION

Hello, I’m Om Shingare. Actually, I’m deeply passionate about creating projects that serve a purpose and have a meaningful impact on my college and the broader community. My dedication lies in the belief that technology should be a force for good, addressing real-world challenges and enhancing educational experiences.

In my journey as a developer, I have consistently focused on projects that are not just about code, but about their practical applications and the value they bring to others. I have a profound aversion to investing my time and skills in projects that do not contribute positively to society. Instead, my commitment is to craft innovative solutions that are helpful, accessible, and transformative for both my peers and educators.

My dedication stems from the conviction that every project I undertake should have a purpose beyond technical proficiency. It is not merely about coding; it is about creating solutions that matter. I am motivated by the prospect of making a difference, whether it is in the classroom, the learning experiences of my fellow students, or the educational landscape of my institution.

I strongly believe in the power of education and technology to drive positive change. By dedicating my efforts to projects that are genuinely helpful for my college and others, I aim to create an impact that resonates far beyond the confines of a computer screen. Each project I undertake is a testament to my commitment to meaningful innovation, and my dedication serves as a guiding principle in my journey as a developer and a lifelong learner.