Module 5: Build a IoT application

James Way & Venetia Furtado

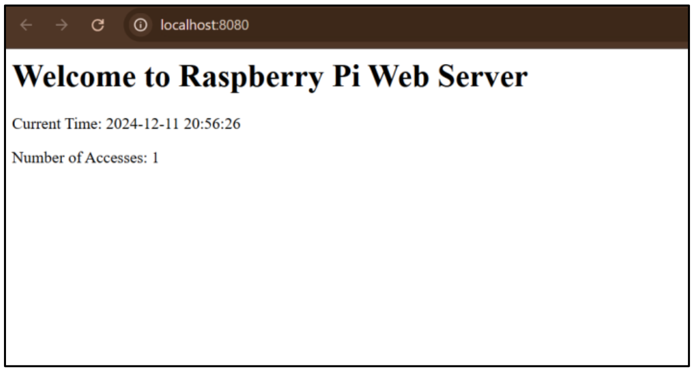
Date: 12/10/2024

Python code:

```
######################################################################
##########
# ECEN 5803 - Mastering Embedded System architecture
# Project 2 Module 5 - Web server application
# Submitted by: James Way & Venetia Furtado
#
# Description: The code for setting up the web server was adapted from
tutorials
# available on the Python website, as cited in the references. It
utilizes the
# http.server package to create a web server that displays the current
time and
# the number of page accesses. The server listens on port 8080 and
dynamically
# generates an HTML page in response to incoming requests.
#
# References:
# https://pythonbasics.org/webserver/
# https://docs.python.org/3/library/http.server.html
#
######################################################################
##########
from http.server import BaseHTTPRequestHandler, HTTPServer
from datetime import datetime

# Globals to track the number of accesses
access_count = 0
```

```python
class RequestHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        global access_count
        access_count += 1

        # Get the current time
        current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

        # Generate the response
        response = f"""
        <html>
        <head><title>Raspberry Pi Web Server</title></head>
        <body>
            <h1>Welcome to Raspberry Pi Web Server</h1>
            <p>Current Time: {current_time}</p>
            <p>Number of Accesses: {access_count}</p>
        </body>
        </html>
        """

        # Send HTTP headers
        self.send_response(200)
        self.send_header("Content-Type", "text/html")
        self.send_header("Content-Length", str(len(response)))
        self.end_headers()

        # Send the HTML content
        self.wfile.write(response.encode("utf-8"))

def run_server():
    host = "0.0.0.0"  # Listen on all available interfaces
    port = 8080       # Port to listen on
    server_address = (host, port)

    # Create the HTTP server
    httpd = HTTPServer(server_address, RequestHandler)
    print(f"Server running on http://{host}:{port}/...")
    try:
        # Start the server
        httpd.serve_forever()
    except KeyboardInterrupt:
        print("\nShutting down the server.")
        httpd.server_close()

if __name__ == "__main__":
    run_server()
```

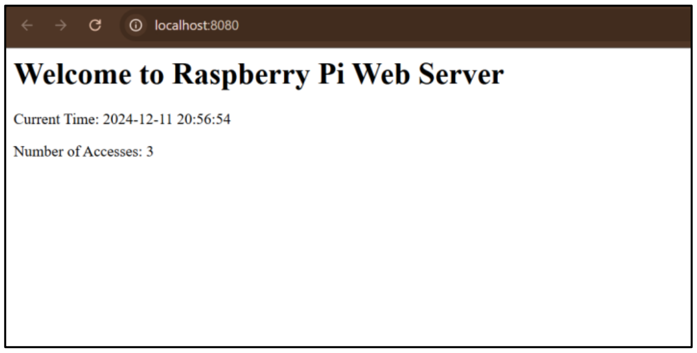
Output:


localhost:8080

Welcome to Raspberry Pi Web Server

Current Time: 2024-12-11 20:56:26

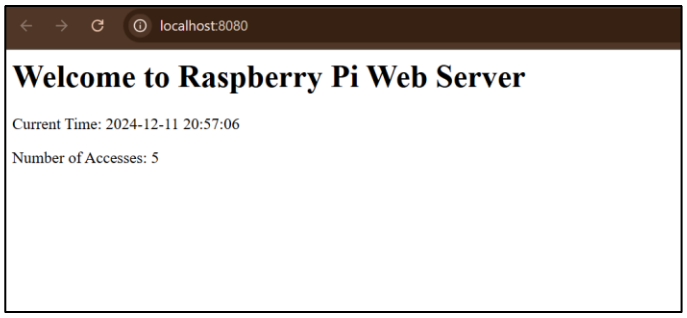
Number of Accesses: 1


localhost:8080

Welcome to Raspberry Pi Web Server

Current Time: 2024-12-11 20:56:54

Number of Accesses: 3


localhost:8080

Welcome to Raspberry Pi Web Server

Current Time: 2024-12-11 20:57:06

Number of Accesses: 5

Figure 1: The web server page at different instances.

Appendix-References

[1] https://pythonbasics.org/webserver/

[2] https://docs.python.org/3/library/http.server.html