Nepal Engineering College

(Affiliated to Pokhara University)

Report on

HTTP Server

Submitted to

Department of computer science and Engineering

Submitted by

Sarthak Neupane(020-335)

Date: 09/01/2025

HTTP Server

Theory:

An HTTP server is a program that listens for requests from clients (typically web browsers) and responds with requested resources. HTTP (Hypertext Transfer Protocol) is the foundation of data communication on the World Wide Web.

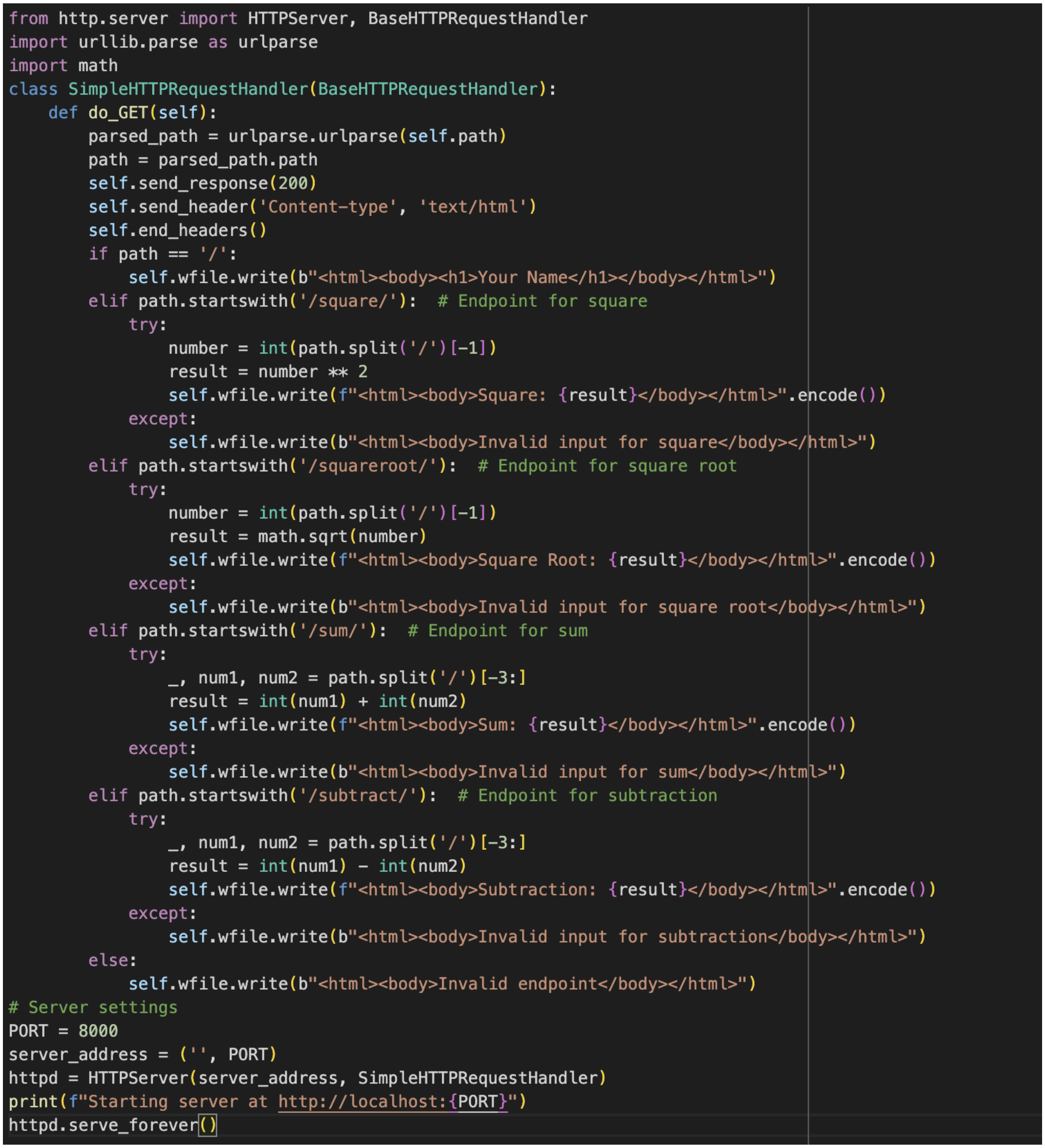
Working of HTTP Server

- The server listens for incoming HTTP requests.
- It processes the request to determine the required response.
- Based on the endpoint and data, it computes the response or retrieves resources.
- The server sends the response back to the client, which renders or processes it.

Python HTTP Server

Python provides the http server module, which can be used to create a lightweight HTTP server. This server can handle GET and POST requests, making it ideal for testing and small-scale applications.

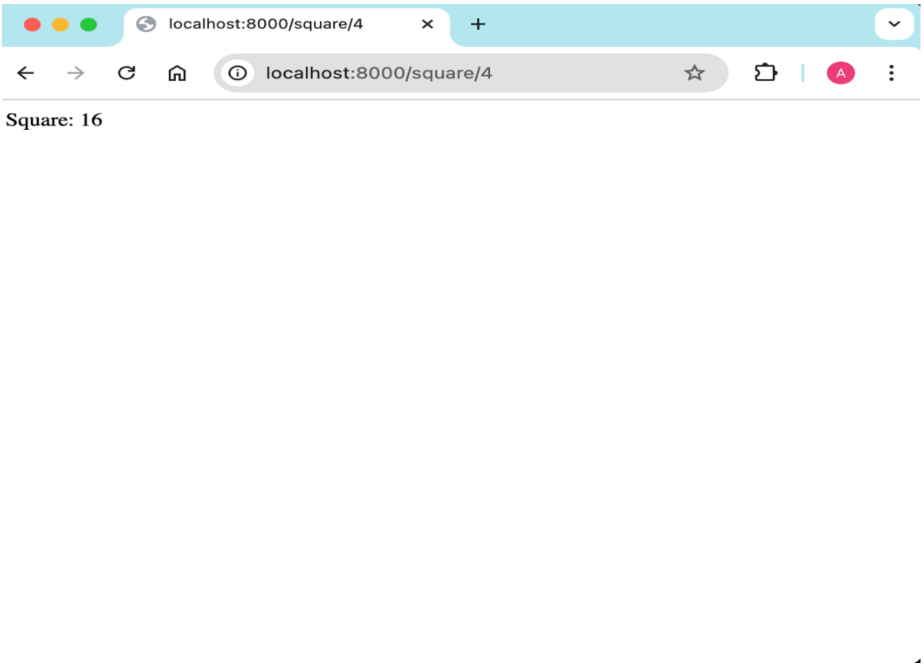
HTTP server program in python that sends your name as response to web browser (client) as default page.



```python
from http.server import HTTPServer, BaseHTTPRequestHandler
import urllib.parse as urlparse
import math
class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        parsed_path = urlparse.urlparse(self.path)
        path = parsed_path.path
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        if path == '/':
            self.wfile.write(b"<html><body><h1>Your Name</h1></body></html>")
        elif path.startswith('/square/'):  # Endpoint for square
            try:
                number = int(path.split('/')[-1])
                result = number ** 2
                self.wfile.write(f"<html><body>Square: {result}</body></html>".encode())
            except:
                self.wfile.write(b"<html><body>Invalid input for square</body></html>")
        elif path.startswith('/squareroot/'):  # Endpoint for square root
            try:
                number = int(path.split('/')[-1])
                result = math.sqrt(number)
                self.wfile.write(f"<html><body>Square Root: {result}</body></html>".encode())
            except:
                self.wfile.write(b"<html><body>Invalid input for square root</body></html>")
        elif path.startswith('/sum/'):  # Endpoint for sum
            try:
                _, num1, num2 = path.split('/')[-3:]
                result = int(num1) + int(num2)
                self.wfile.write(f"<html><body>Sum: {result}</body></html>".encode())
            except:
                self.wfile.write(b"<html><body>Invalid input for sum</body></html>")
        elif path.startswith('/subtract/'):  # Endpoint for subtraction
            try:
                _, num1, num2 = path.split('/')[-3:]
                result = int(num1) - int(num2)
                self.wfile.write(f"<html><body>Subtraction: {result}</body></html>".encode())
            except:
                self.wfile.write(b"<html><body>Invalid input for subtraction</body></html>")
        else:
            self.wfile.write(b"<html><body>Invalid endpoint</body></html>")
# Server settings
PORT = 8000
server_address = ('', PORT)
httpd = HTTPServer(server_address, SimpleHTTPRequestHandler)
print(f"Starting server at http://localhost:{PORT}")
httpd.serve_forever()
```

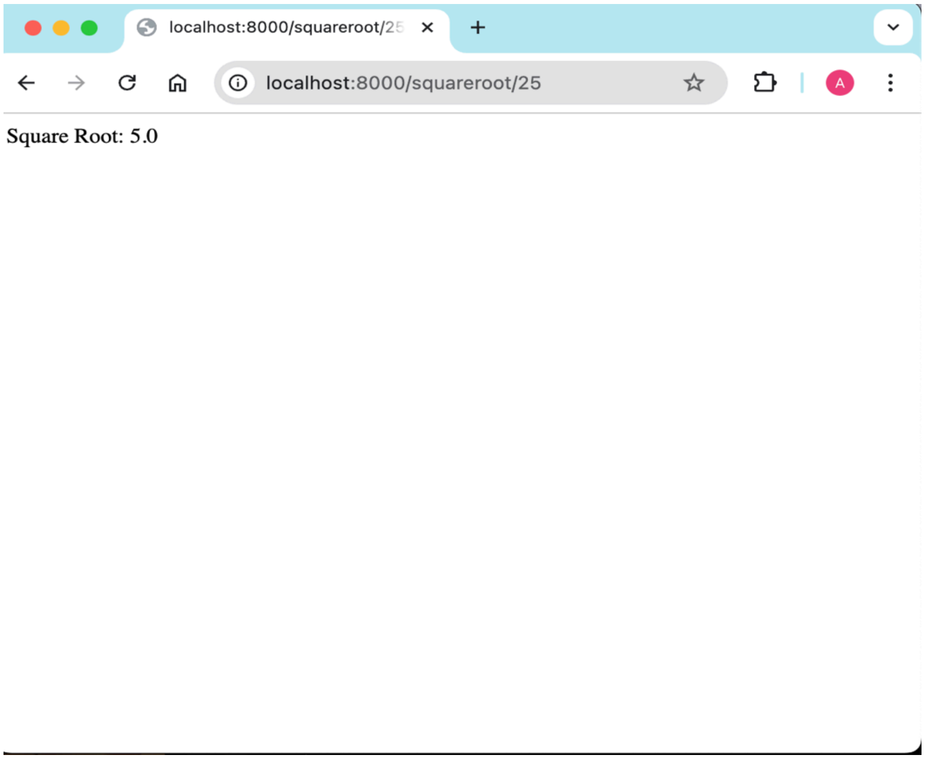
1. **Use python interpreter to run this python script.**
2. **Modify the server program to respond to the endpoints.**
3. **Use browser to test the endpoints.**

a) a) /square/4



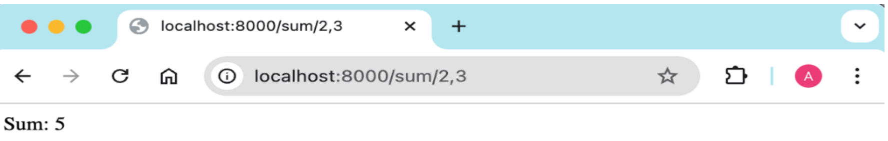
```
Square: 16
```

b) /squareroot/25



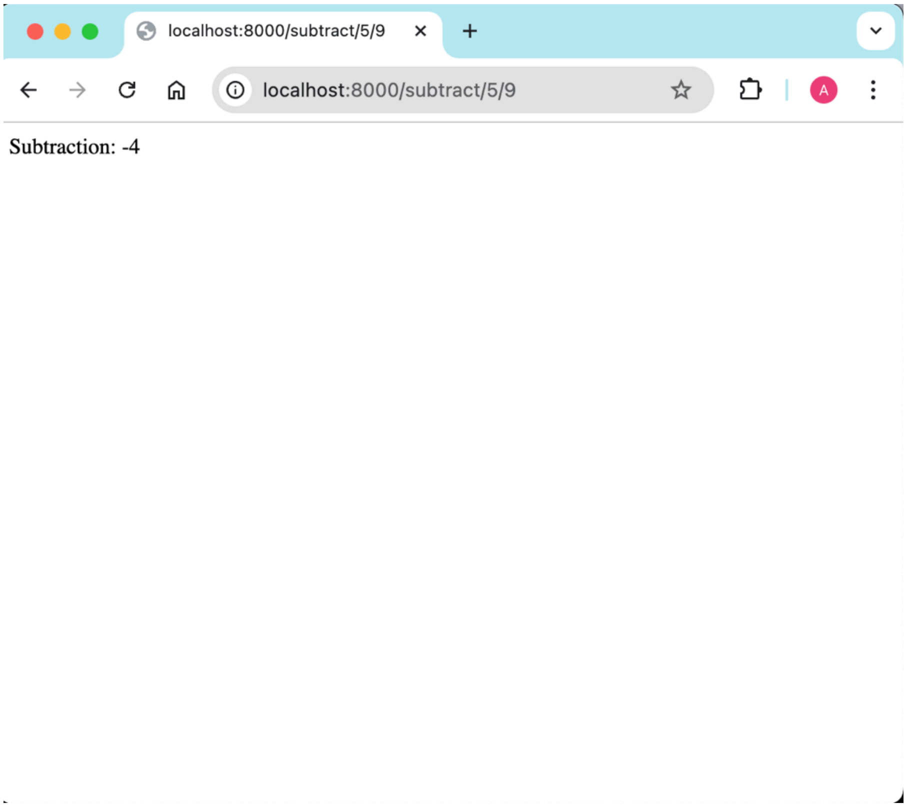
```
Square Root: 5.0
```

c) /sum/2,3



```
Sum: 5
```

d) /subtract/5/9



Subtraction: -4