Static Web Pages Using a Node.js Web Server

Domain:	Appointment Scheduling
Estimated time:	60 minutes

This assignment introduces the fundamentals of web servers and client–server architecture. You will learn how web pages are delivered to browsers and how Node.js can be used to create a simple web server using JavaScript. This server will be reused and extended throughout the semester.

1. Learning Objectives

- Understanding what a client–server architecture is
- Understanding how web servers handle requests and responses
- Learning the difference between traditional web servers and application servers
- Creating a basic Node.js web server
- Serving static HTML pages using Node.js

2. Client–Server Architecture (Conceptual Overview)

Most web applications follow a client–server architecture.

The client is typically a web browser. It requests information such as web pages or data. The server receives requests, processes them, and sends responses back to the client.

Communication happens over HTTP. Each interaction follows this pattern:

1. Client sends a request
2. Server processes the request
3. Server sends a response
4. Client displays the response

3. Types of Servers

1. Traditional web servers (e.g., Apache, Nginx) are designed to serve files efficiently and are configured using configuration files.
2. Node.js servers are application servers. You write JavaScript code that explicitly handles requests and generates responses.
3. Cloud platforms (e.g., Azure, AWS, Cloudflare) provide managed infrastructure but still rely on the same client–server principles.

4. Required Software

- Node.js (LTS version)
- Code editor (VS Code recommended)
- Web browser

5. Assignment Tasks

1. Install/Verify that your system has: Chrome, VS Code, and Node.js
2. Create a project folder with the following structure:

```
ppa1/
  server.js
  public/
    index.html
    provider.html
    client.html
```

3. Create basic HTML pages.
4. Write a Node.js server that serves these pages.
5. Verify pages load via http://localhost:3000.

6. Example Node.js Server Code

```
import http from "http";
import fs from "fs";

const server = http.createServer((req, res) => {

  let filePath = "./public/index.html";

  if (req.url === "/provider") { filePath = "./public/provider.html";
  if (req.url === "/client")   { filePath = "./public/client.html";   }

  fs.readFile(filePath, (err, content) => {
    if (err) {
      res.writeHead(500);
      res.end("Server error");
      return;
    }

    res.writeHead(200, { "Content-Type": "text/html" });
    res.end(content);
  });
});

server.listen(3000, () => {
  console.log("Server running at http://localhost:3000");
});
```

7. Deliverables

- server.js
- index.html, provider.html, client.html
- Pages served via Node.js