Task 5: Building a Basic Web Server

Objective:

Build a basic web application using the built-in `http` module in Node.js. The application should handle a few routes (e.g., `/`, `/about`, `/contact`). Test the routes using Postman or a browser to ensure they return the correct responses.

Prerequisites:

- Basic understanding of JavaScript.
- Node.js installation.

Concepts:

- **HTTP Module:**
  - The `http` module is a built-in module in Node.js used to create an HTTP server.
  - It allows handling of HTTP requests and responses.

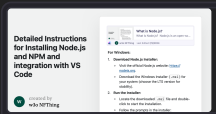
  Example:

```
// server.js
const http = require('http');

const server = http.createServer((req, res) => {
  if (req.url === '/') {
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    res.end('Welcome to the Home Page');
  } else {
    res.writeHead(404, { 'Content-Type': 'text/plain' });
    res.end('404 Not Found');
  }
});

server.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

Setup:

1. **Install Node.js:**

   Ensure Node.js is installed on your machine. You can access the instructions here:

   - 
     Detailed Instructions for Installing Node.js and NPM and integration...
     For Windows: — Download Node.js Installer: — Visit the official Node.js website:...
     w3o NFThing Last Edited 7/3/2024

Tasks:

1. **Basic Routes:**
   - **Task:**

- Create a file named `server.js` and write your own code that:
  - Creates a basic web server using the `http` module.
  - Handles routes for `/`, `/about`, and `/contact`.
  - Returns appropriate responses for each route.
- **Outcome:**
  - Ensure the routes return the correct responses when accessed via Postman or a browser.

Instructions:

- **Perform the following tasks:**
  - Write the required code in `server.js`.
  - Run the server using Node.js to ensure the code executes without errors and demonstrates the use of the `http` module.

Resources:

- HTTP | Node.js v22.4.0 Documentation
  This module, containing both a client and server, can be imported via — require('node:http') (Commo...
  https://nodejs.org/api/http.html

Videos:

GitHub Instructions:

1. **Open in Visual Studio Code:**

- After clicking on the "Open in Visual Studio Code" button from the GitHub Classroom confirmation page, Visual Studio Code (VSCode) will open the repository directly.
- If prompted, select "Open" or "Allow" to open the repository in VSCode.

2. **Complete the Task:**
   - Write your solution in `server.js`.
3. **Run and Test Your Code:**
   - Run your code to ensure it works correctly. Use the following command:

```
node server.js
```

4. **Commit Your Changes:**
   - Commit your changes with a meaningful message:

```
git commit -m "Completed Basic Web Server task"
```

5. **Push Your Changes to Your Forked Repository:**
   - Push your changes to your forked repository:

```
git push origin main
```

6. **Create a Pull Request:**
   - Go to your forked repository on GitHub.
   - Click on the "Pull Requests" tab.
   - Click the "New Pull Request" button.
   - Ensure the base repository is the original template repository and the base branch is `main`.
   - Ensure the head repository is your forked repository and the compare branch is `main`.
   - Click "Create Pull Request".
   - Add a title and description for your pull request and submit it.

Summary of Commands:

```
# Fork the repository on GitHub

# Clone the forked repository
git clone https://github.com/your-github-username/repository-name.git
cd repository-name

# Complete the task by writing and running the code in server.js

# Add, commit, and push your changes
git commit -m "Completed Basic Web Server task"
git push origin main
```

```
# Create a pull request on GitHub
```