Demo: Building a WebSockets Chat Application with Node.js

This lab will guide you through using GitHub Copilot to build a WebSockets-based chat application in Node.js while exploring Copilot's features.

Prerequisites

- Node.js installed
- VS Code with GitHub Copilot extension
- Basic understanding of JavaScript and WebSockets

Step 1: Initialize a Node.js Project

1. Open VS Code and create a new folder for your project.
2. Open a terminal and run:

   ```
   npm init -y
   ```

3. Install dependencies:

   ```
   npm install express ws
   ```

Step 2: Generate Code with comments using Copilot

1. Create a file named `server.js`.

2. Open inline chat using `Ctrl+I` and type the following prompt and press Enter.

   ```
   Create an Express server with WebSockets support
   ```

3. Let GitHub Copilot suggest the code and accept it.

Step 3: Build the Frontend (HTML & JS)

1. Create a `public` folder and add `index.html`.

2. Open Copilot Edits and write the following prompt. Also attach the `server.js` a `package.json` to the Copilot edits using `Add file` option. Ensure the `index.html` is currently opened in editor.

```
Create a bootstrap CSS based chat UI that can send message to web socket
application on clicking the button. Use the bootstrap CSS CDN url.
```

3. This will update the `index.html` file and add the HTML code and JavaScript code for communicating with Web Socket server.

Step 4: Update application to serve html page

1. Open the `server.js` on the editor. Then open the GitHub Copilot Edits and run the following prompt

```
Update the node app to serve index.html  from public folder on path /
```

2. This will update the code in `server.js` to server html page on root path. Accept the suggestions.

Step 5: Run the application

1. Run the server:

```
node server.js
```

2. Open the browser and start chatting.