

Detailed Environment Description and Testing Process for the Node.js Web Server

Environment Setup

1. Node.js Environment:

- **Node.js Version:** 18.17.1
 - Node.js is used as the runtime environment to execute the JavaScript code for the web server. Ensure a stable version of Node.js (LTS recommended) is installed for maximum compatibility.
- **NPM Version:** 9.8.0
 - NPM, bundled with Node.js, is used to manage dependencies if required in future development.
- **Operating System:**
 - Tested on the following:
 - **Windows 10 (64-bit):** A widely used operating system suitable for both development and production environments.
 - **macOS Ventura 13.2:** Tested for compatibility with Apple's macOS ecosystem.
- **System Architecture:** x64 (64-bit architecture ensures the compatibility of Node.js and other tools).

2. Browser Details:

- **Google Chrome:** Version 118.0.5993.88 (Official Build, 64-bit)
 - Chrome is the most commonly used browser, ensuring wide user compatibility.
- **Mozilla Firefox:** Version 118.0.1 (64-bit)
 - Firefox is another popular browser, ensuring compatibility with an alternative rendering engine.
- Optionally tested on Microsoft Edge and Safari for additional validation.

Testing Steps

Setup and Initial Test

1. Install Node.js:

- Download Node.js from the [official Node.js website](https://nodejs.org/) and install it according to the operating system's guidelines.

Verify the installation by running:
`node -v`

npm -v

2. Prepare Project Files:

- Ensure the `server.js` file and `index.html` file are in the same directory.
- The `index.html` file contains the webpage that the server will serve to users.

3. Run the Server:

- Open a terminal or command prompt. Navigate to the project directory:
`cd path/to/project`
- Start the server by executing:
`node server.js`

4. Access the Web Server:

Open a browser and go to the URL:

`http://localhost:3000`

- Confirm that the webpage is served correctly with the following elements:
 - Title: "My First Web Server"
 - Heading: "Welcome to My First Node.js Web Server"
 - Paragraph: A description about the web server.

Functionality Testing

1. Successful Content Delivery:

- Ensure that the `index.html` file is served correctly, and the content displays as intended in the browser.

2. Error Handling:

- Temporarily rename or delete the `index.html` file to simulate a missing file scenario.
- Refresh the browser and confirm that the server responds with:
 - HTTP Status Code: `500 Internal Server Error`
 - Message: "Internal Server Error"

3. Cross-Browser Compatibility:

- Open the web page in multiple browsers (e.g., Chrome, Firefox, Edge, Safari) to ensure consistent rendering across different platforms.

4. Cross-Platform Testing:

- Run the web server on different operating systems (Windows, macOS, Linux) to verify that the code executes without any platform-specific issues.

Results ScreenShots

