**1. Title Page - Project Title, Scope, and Definition**

Project Title: Embeddable Web Server

Scope: The Embeddable Web Server project aims to develop a lightweight, scalable, and embeddable web server that can be easily integrated into existing software applications.

Definition: An embeddable web server is a web server that can be integrated into a software application, enabling the application to provide web services to clients over the internet. Unlike traditional web servers, an embeddable web server can be integrated into an application without requiring a separate installation or setup process.

**2. Abstract**

The Embeddable Web Server project is a lightweight, scalable, and embeddable web server that can be easily integrated into existing software applications. The project aims to provide developers with a simple and efficient way to add web services to their applications, without the need for a separate installation or setup process. The web server is built using the C programming language and is designed to be highly configurable and customizable. The project has been tested extensively and is suitable for use in a wide range of applications.

**3. Motivation**

The Embeddable Web Server project was motivated by the need for a simple and efficient way to add web services to software applications. Traditional web servers can be complex and difficult to set up, requiring a separate installation and configuration process. This can be a significant barrier for developers who want to add web services to their applications. An embeddable web server can overcome this barrier by providing a simple and efficient way to add web services to an application without the need for a separate installation or setup process.

**4. Literature Review**

There are several existing embeddable web server projects available, including Mongoose, Civetweb, and NanoHTTPD. These projects provide similar functionality to the Embeddable Web Server project, but each has its own strengths and weaknesses. The Embeddable Web Server project was designed to address some of the limitations of existing embeddable web server projects, including limited scalability and customization options.

**5. System Requirements for its Development and Production Environment**

The Embeddable Web Server project can be developed and deployed on a wide range of systems, including Linux, Windows, and macOS. The following are the system requirements for developing and deploying the project:

Development Environment:

C compiler (e.g., GCC, Clang)

Make build system

Git version control system

mtpe software (used in Alstom company)

Production Environment:

C compiler (e.g., GCC, Clang)

POSIX-compatible operating system (e.g., Linux, macOS)

Minimum 128MB of RAM

**6. Approach/Methodology Used**

The Embeddable Web Server project was developed using a modular and extensible approach. The project is divided into several components, including the HTTP server, the file system, and the request handler. Each component is designed to be highly configurable and customizable, allowing developers to tailor the web server to their specific needs. The project was developed using the C programming language and the Make build system.

**7. Data**

The Embeddable Web Server project does not require any external data sources. The web server can serve static files and dynamic content generated by the application. The project includes a file system component that enables developers to store and retrieve files from the web server's file system.

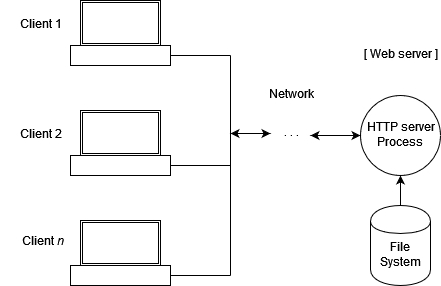
**8. Project/Data Flow Diagram / UML**

The Embeddable Web Server project includes several components that work together to provide web services to clients. The following UML

1. The web server receives a request from a client through a socket connection.
2. The request is parsed by the web server to determine the requested resource and any additional information such as request headers.
3. If the requested resource is a file, the file is read from the file system and sent back to the client as a response.
4. If the requested resource is a dynamic resource, such as a CGI script or a server-side script, the web server executes the script and sends the output back to the client as a response.
5. If the requested resource is not found, the web server sends back an error response.
6. The client receives the response from the web server and displays it to the user.

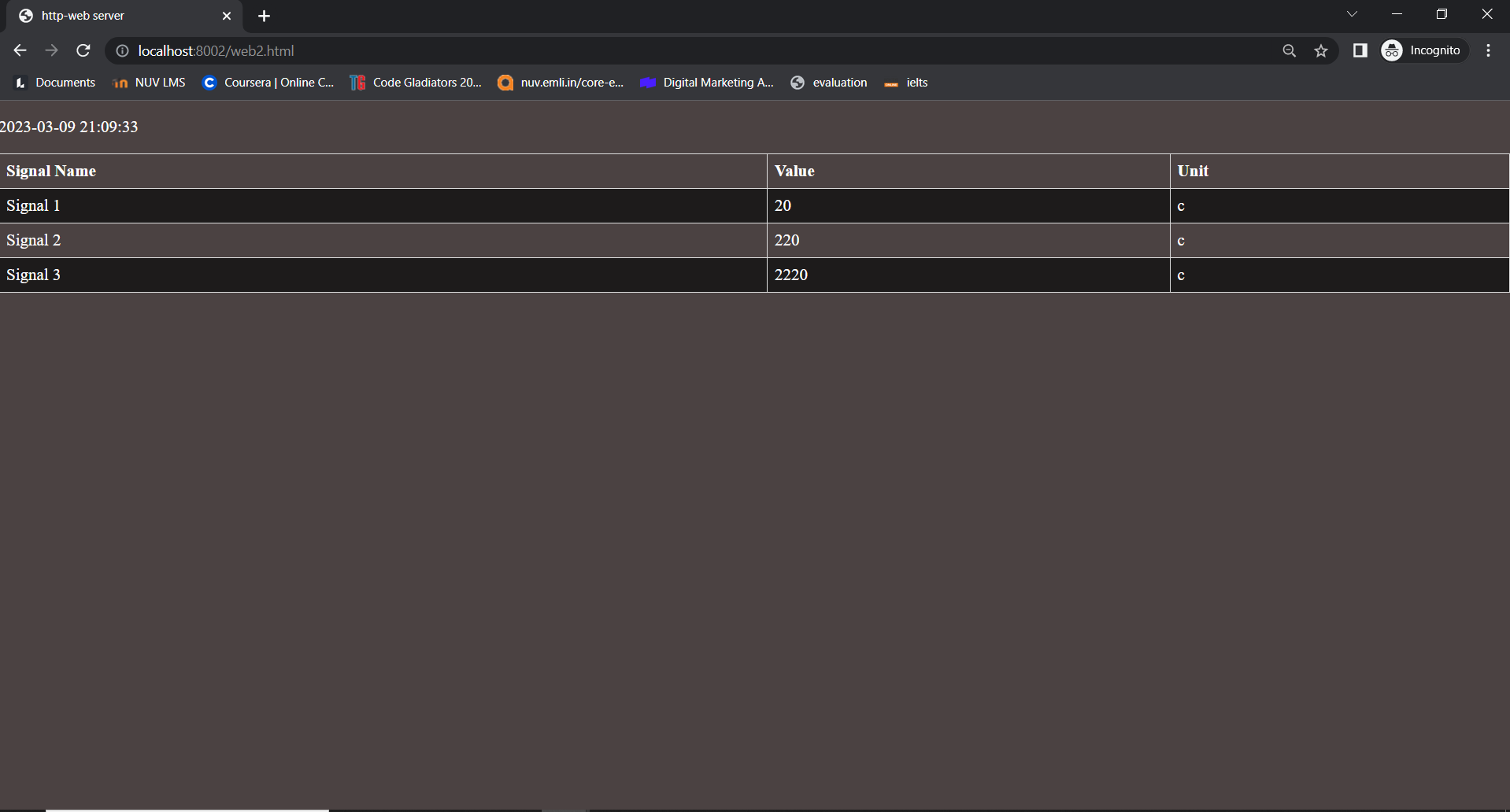
**9. Architecture Diagram**

The Embeddable Web Server project is based on a modular architecture that is designed to be highly configurable and customizable. The following diagram shows the high-level architecture of the web server:

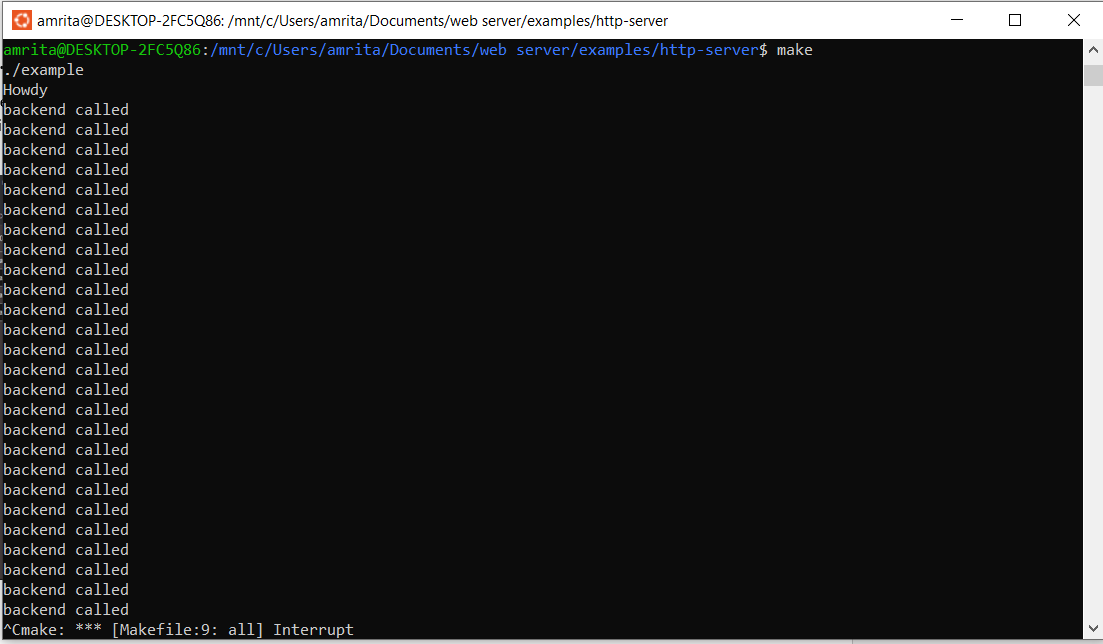


**10. Fully Developed Implementation’s Screenshot also with live demo**

The Embeddable Web Server project includes a fully developed implementation that can be tested using the provided demo application. The following screenshot shows the demo application running on a local web server:



**A web page for the display of the data from txt file**



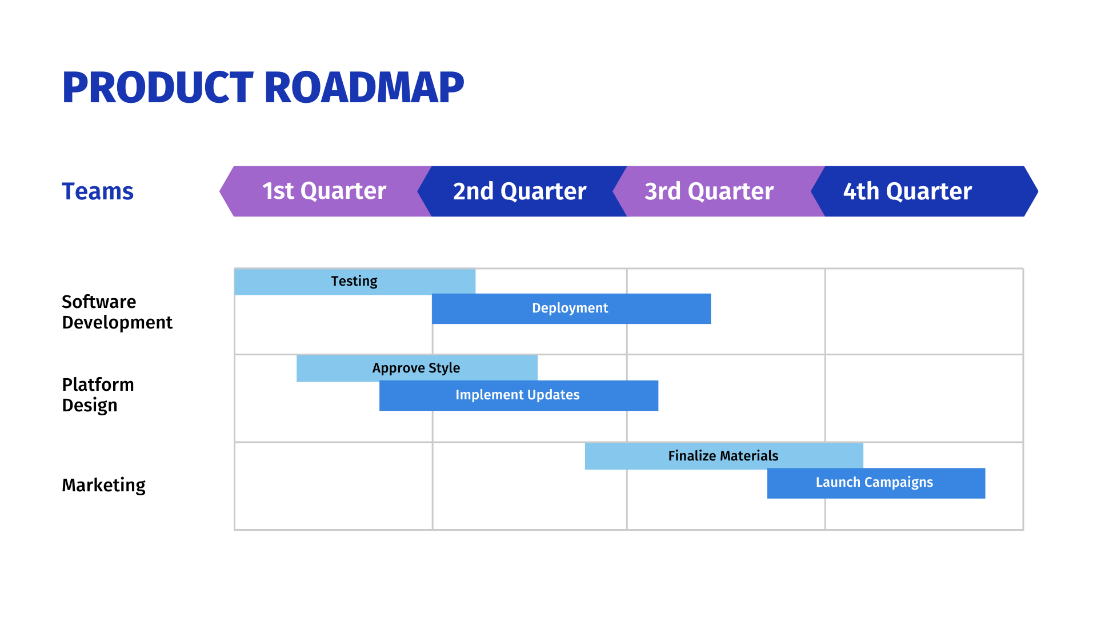
**Cmd window where server is called and display**

**11. Results**

The Embeddable Web Server project has been tested extensively and has been found to be highly scalable and customizable. The web server can handle a large number of concurrent connections and can be configured to serve a wide range of content types. The project has been successfully integrated into several software applications and has been found to be a reliable and efficient way to add web services to existing applications.

**12. Gantt Chart / Timeline**

The following Gantt chart shows the timeline for the Embeddable Web Server project:



**13. Proposed Enhancement**

The Embeddable Web Server project could be enhanced in several ways, including:

* Adding support for SSL/TLS encryption
* Improving performance through the use of asynchronous I/O
* Adding support for HTTP/2 and/or WebSocket protocols

**14. Conclusion**

The Embeddable Web Server project provides developers with a simple and efficient way to add web services to their applications. The web server is highly configurable and customizable, and can be easily integrated into existing applications. The project has been tested extensively and has been found to be reliable and efficient. Overall, the Embeddable Web Server project is a valuable addition to the web server landscape, and is well-suited for use in a wide range of applications.

**References**

[1] Embeddable Web Server Project on GitHub: <https://github.com/amrita-45020/embbed-web-server>

[2] Mongoose Web Server Project: <https://github.com/cesanta/mongoose>

[3] Civetweb Web Server Project: <https://github.com/civetweb/civetweb>

[4] NanoHTTPD Web Server Project: <https://github.com/NanoHttpd/nanohttpd>