

INDEX

HEADING	PAGE
TITLE	1
INTRODUCTION	2
FEASIBILITY STUDY	3
METHODOLOGY	4
FACILITIES REQUIRED	5
DFD	6
SCREENSHOT	7
BIBLIOGRAPHY	8

TITLE

Name	Uday Raj Singh
Roll no	2100100100182
Email	udayrajsingh288@gmail.com
Phone no	+91 9695975787
Branch	CSE
Batch	2021 - 2025
Topic	Network Programming in C

Introduction

This project is a web application built using the Winsock API in C, designed to explore low-level network programming, database interaction, and web technologies. The application features both the GET and POST HTTP methods, commonly used for retrieving and submitting data between client and server. Built with simplicity and performance in mind, it utilizes the SQLite database for efficient data storage and retrieval.

The project focuses on fundamental web application functionalities like user registration and authentication, implemented through signup and sign-in pages. The signup process is further enhanced with a One-Time Password (OTP) system, where the use of sessions plays a key role in ensuring secure and smooth interaction between the client and server. Sessions help manage user state, allowing the application to recognize logged-in users and protect sensitive operations, such as verifying the OTP during signup.

The choice of Winsock API in C emphasizes the learning experience of working at a low level, directly managing network sockets and data transfer. This approach provides deeper insight into how web servers handle requests, manage connections, and interact with clients, bypassing the abstraction layers provided by higher-level programming languages or frameworks.

SQLite was chosen for its light footprint and ease of integration, making it ideal for small to medium-scale applications. The database stores user information, such as credentials and session data, ensuring that the application can manage multiple users effectively.

The combination of Winsock and SQLite results in a lightweight and efficient application, well-suited for environments where performance and resource management are critical. This project showcases essential web development concepts, such as handling HTTP requests, managing user sessions, and interacting with a database, all while working directly with the networking stack.

Through this project, a deeper understanding of network programming, database management, and web security is gained. It also serves as a solid foundation for further expansion, such as adding more complex features, optimizing performance, or scaling the application for larger user bases. The use of OTP during signup and session management ensures that the application provides a secure and user-friendly experience, making it a practical and insightful project in the realm of web application development with C and Winsock.

Feasibility Study

This feasibility study evaluates the practicality of developing a web application using the Winsock API in C, focusing on implementing HTTP GET and POST methods, user authentication, session management, and SQLite database interaction. The study considers technical, operational, and economic aspects to determine the viability of the project.

1. Technical Feasibility

The project is technically feasible due to the availability of appropriate technologies. The Winsock API allows direct management of network sockets, providing low-level control for handling HTTP requests. SQLite, a lightweight and serverless database, is well-suited for storing user credentials and session data.

Key challenges include managing low-level socket communication and securely implementing session handling and OTP validation, which are more complex in C compared to higher-level frameworks. However, with proper security measures (e.g., encryption, session timeouts), these challenges are manageable.

Overall, the necessary tools and knowledge are available, making the project technically feasible.

2. Operational Feasibility

Operationally, the application is designed for small to medium-sized environments, such as educational projects or lightweight applications. The combination of C and SQLite ensures minimal resource usage, allowing it to run efficiently even on limited hardware.

User authentication through signup/sign-in pages, secured with OTP and session management, aligns with modern web standards, providing a smooth user experience. The manual control over network communication offers better insight into how servers work but requires ongoing maintenance and monitoring for vulnerabilities.

3 Economic Feasibility

The project has low development costs, as it uses free and open-source technologies (Winsock API, SQLite). The primary investment is the developer's time in building and maintaining the application. There are no licensing fees or significant infrastructure costs, making it economically viable, especially for educational purposes or small-scale deployments.

4. Conclusion

The project is feasible on technical, operational, and economic grounds. While it requires a strong understanding of C and network programming, the benefits of learning low-level web development and the minimal costs make this project worthwhile.

Methodology

1. Phase 1: Setup (Week 1)

- **Goal:** Set up the development environment.
 - Install necessary tools (C compiler, SQLite, Winsock libraries).
 - Set up a simple SQLite database.

2. Phase 2: Basic Networking (Weeks 2-3)

- **Goal:** Implement basic networking with Winsock API.
 - Write code to create a server that listens for client connections.
 - Handle basic HTTP GET and POST requests.

3. Phase 3: Signup and OTP (Weeks 4-5)

- **Goal:** Add user signup and OTP functionality.
 - Create signup and sign-in pages.
 - Implement OTP generation and validation for secure signup.

4. Phase 4: Database Integration (Weeks 6-7)

- **Goal:** Connect the application to SQLite for storing user data.
 - Write code to store user credentials and session info.
 - Ensure login and signup data is saved in the database.

5. Phase 5: Security (Week 8)

- **Goal:** Add basic security features.
 - Securely manage sessions to keep users logged in.

6. Phase 6: Testing (Weeks 9-10)

- **Goal:** Test the entire application.
 - Test all features like signup, OTP, login, and session handling.
 - Fix any bugs found during testing.

7. Phase 7: Finalize and Deploy (Week 11)

- **Goal:** Finish the project.
 - Optimize the code.
 - Write basic documentation.
 - Deploy the application for use.

Facilities Required

1. Development Environment

- **Computer:** A basic system with at least 4GB RAM.
- **Operating System:** Windows (required for Winsock API).
- **IDE/Text Editor:** Visual Studio, Code::Blocks, or Visual Studio Code.

2. Software and Tools

- **C Compiler:** MinGW or Visual Studio.
- **Winsock API Libraries:** Winsock2.h and Ws2_32.lib for network programming.
- **SQLite:** SQLite for user data and session management.

3. Internet Connection

- Needed for testing network functionality and OTP sending.

4. Testing and Debugging

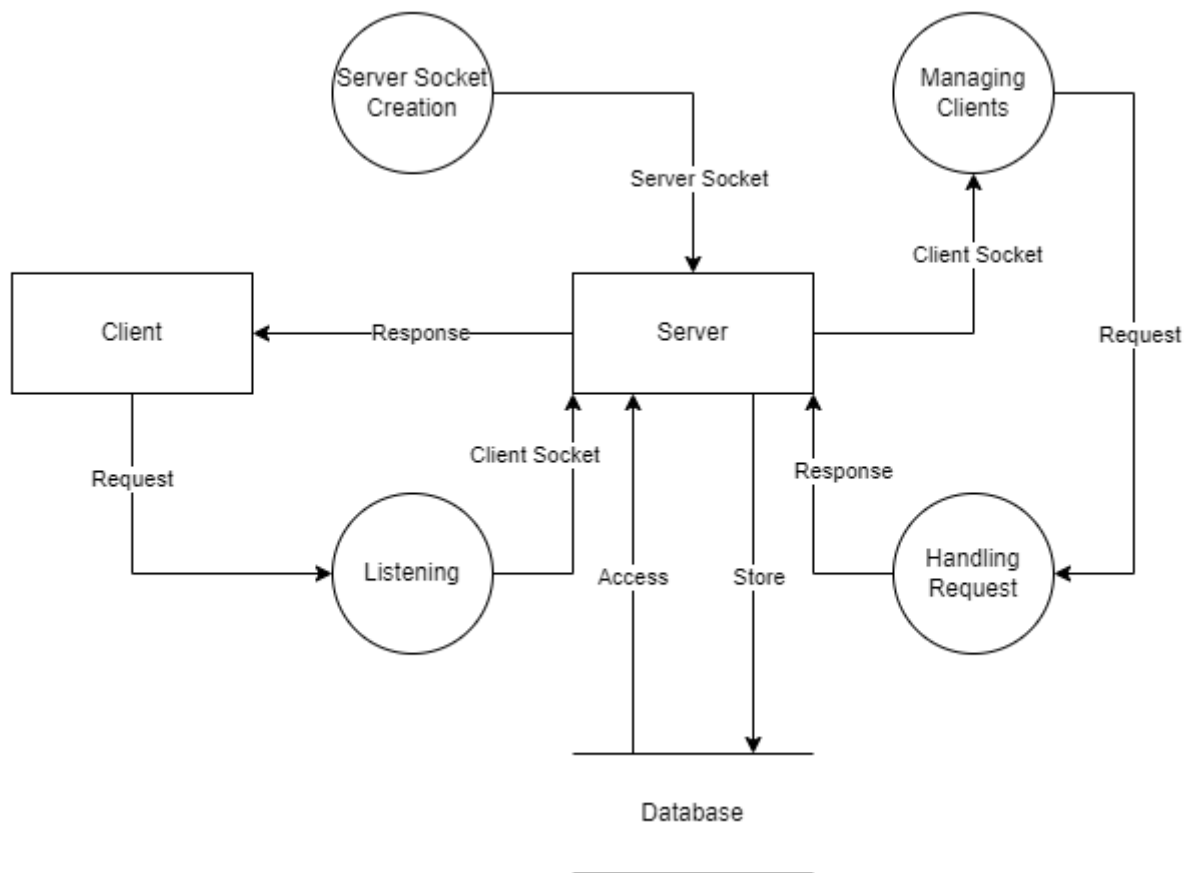
- **Postman:** For testing HTTP requests.
- **Debugger:** GDB for code debugging.

5. Documentation and Version Control

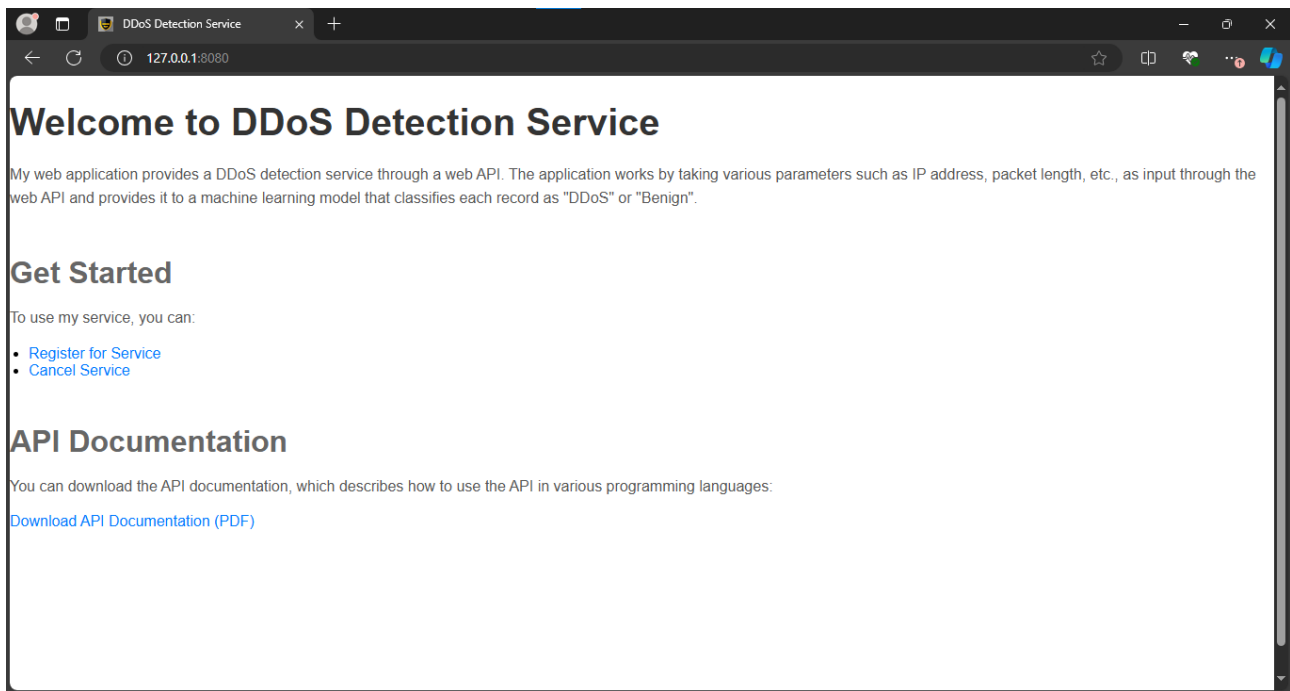
- **Word Processor:** For writing documentation.
- **Git:** For version control and project management.

This setup provides everything necessary to develop, test, and maintain the project.

DFD



Screenshot



Bibliography

1. **Beej's Guide to Network Programming**

- A comprehensive guide on socket programming in C, offering in-depth explanations of how to use network protocols and Winsock API. Available at: <http://beej.us/guide/bgnet/>.

2. **Microsoft Documentation: Windows Sockets 2**

- Official documentation on using Winsock API in Windows for network communication. Provides detailed API references and examples. Available at: <https://learn.microsoft.com/en-us/windows/win32/winsock/windows-sockets-start-page-2>

3. **SQLite Documentation**

- Official SQLite documentation offering detailed guidance on database setup, usage, and integration with various programming languages, including C. Available at: <https://www.sqlite.org/docs.html>

4. **The C Programming Language (2nd Edition) by Brian W. Kernighan and Dennis M. Ritchie**

- A classic reference for the C programming language, essential for understanding how to handle memory, data structures, and file I/O, which are critical in socket programming and database integration.

5. **Postman Documentation**

- Guide for using Postman to test HTTP requests, which is crucial for testing the web application's functionality. Available at: <https://learning.postman.com/docs/>

6. **GDB: The GNU Project Debugger Documentation**

- Official documentation on using GDB for debugging C programs, which is essential for troubleshooting during development. Available at: <https://www.gnu.org/software/gdb/documentation/>