This project showcases a straightforward Proxy Application in Go, encompassing essential firewall capabilities, rate limiting, geo-blocking, logging, and a user-friendly web interface.

The Proxy Application comprises the following functionalities:

Basic Firewall Features:

* Blocking based on Source/Destination IP.
* Blocking based on Source/Destination port.
* Protocol-based blocking (e.g., blocking all HTTPS traffic).

Rate Limiting:

* Restricting the number of requests from a specific source IP within a defined time frame (e.g., a maximum of 100 requests per minute).
* Controlling the overall bandwidth usage by a specific IP or service.

Geo-blocking:

* Permitting or blocking traffic based on the geographic location of the source or destination IP using a GeoIP database.

Logging:

* Recording all blocked traffic with timestamps, source/destination IPs, and the reason for blocking.
* Implementing a mechanism for regular log rotation and archiving.

Information:

* Creating a web-based UI enabling administrators to set rules, view logs, and monitor the system.
* Displaying statistics such as bandwidth usage over time.

Test Coverage:

* Establishing unit tests for both the firewall logic and the UI.
* Aiming for a minimum of 80% code coverage.

Static Analysis (Linting):

* Ensuring that the code adheres to best practices and standards.
* Utilizing golangci-lint for static analysis.

Proxy Application Architecture:

Firewall Core:

*Description:* The pivotal component overseeing the fundamental functions of the firewall, encompassing the handling of incoming packets, rule enforcement, and traffic statistics management.

*Components:*

* Packet Handling
* Rule Enforcement
* Traffic Counters
* Rate Limiting
* Geo-blocking

SSL Inspection (if applicable)

* *Dependencies:*
* iptables (for rule enforcement)
* geoip2.Reader (for GeoIP data)

Logging Management:

*Description:* Manages the logging of blocked traffic, including timestamps, source/destination IPs, and reasons for blocking.

*Components:*

* Log Entry Struct
* Log Rotation
* Log Archiving

Web-Based UI (Dashboard):

*Description:* Provides a user interface for administrators to configure firewall rules, view logs, and monitor system statistics.

*Components:*

* Fiber Framework
* Dashboard Routes
* Log Viewer
* System Statistics Display

Packet Capture:

*Description:* Captures incoming and outgoing packets for analysis and processing by the firewall core.

*Components:*

* Packet Capture Handler
* Interface Detection
* Packet Inspection

Testing:

*Description:* Includes unit tests to ensure the reliability and correctness of the firewall's functionality.

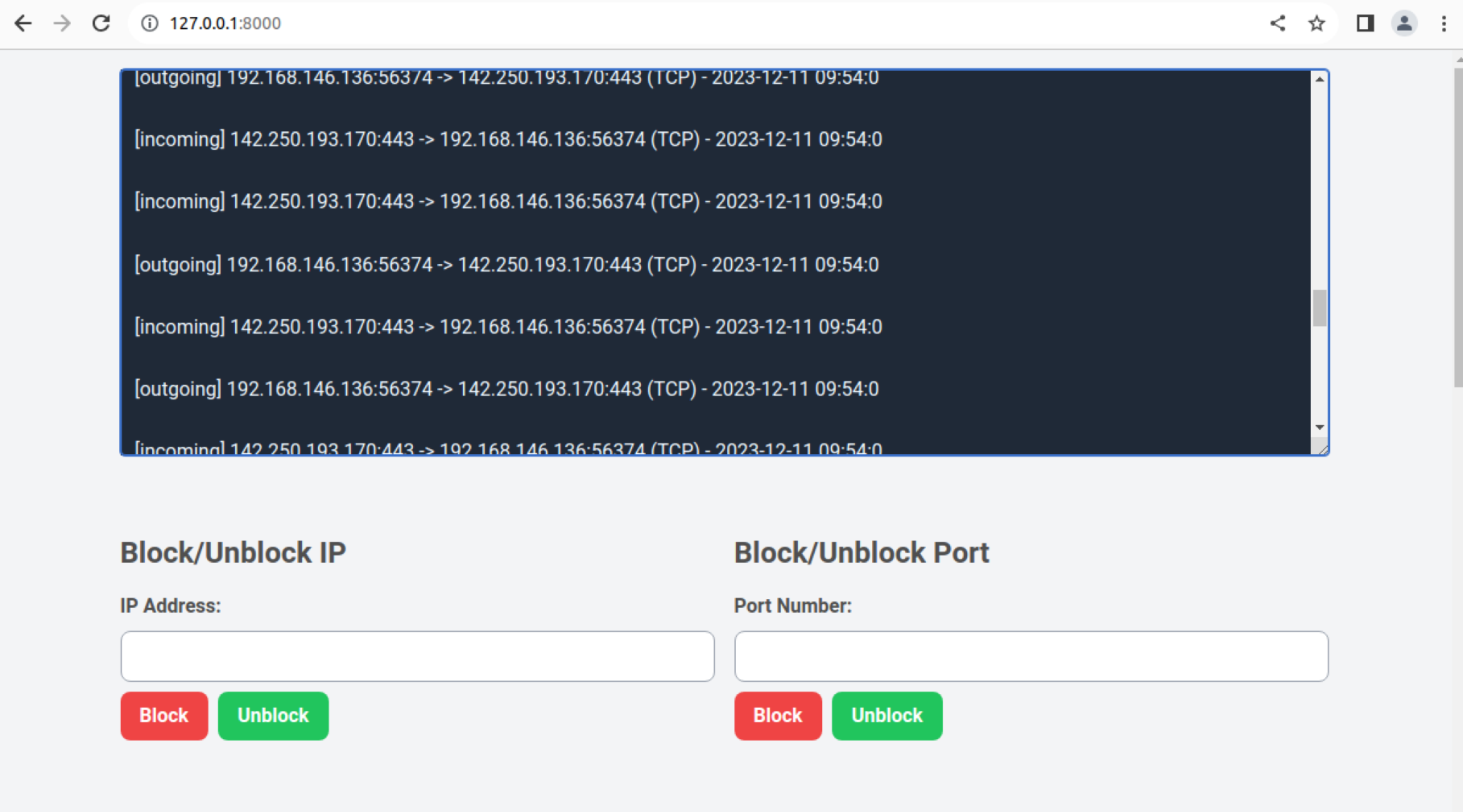
*Components:*

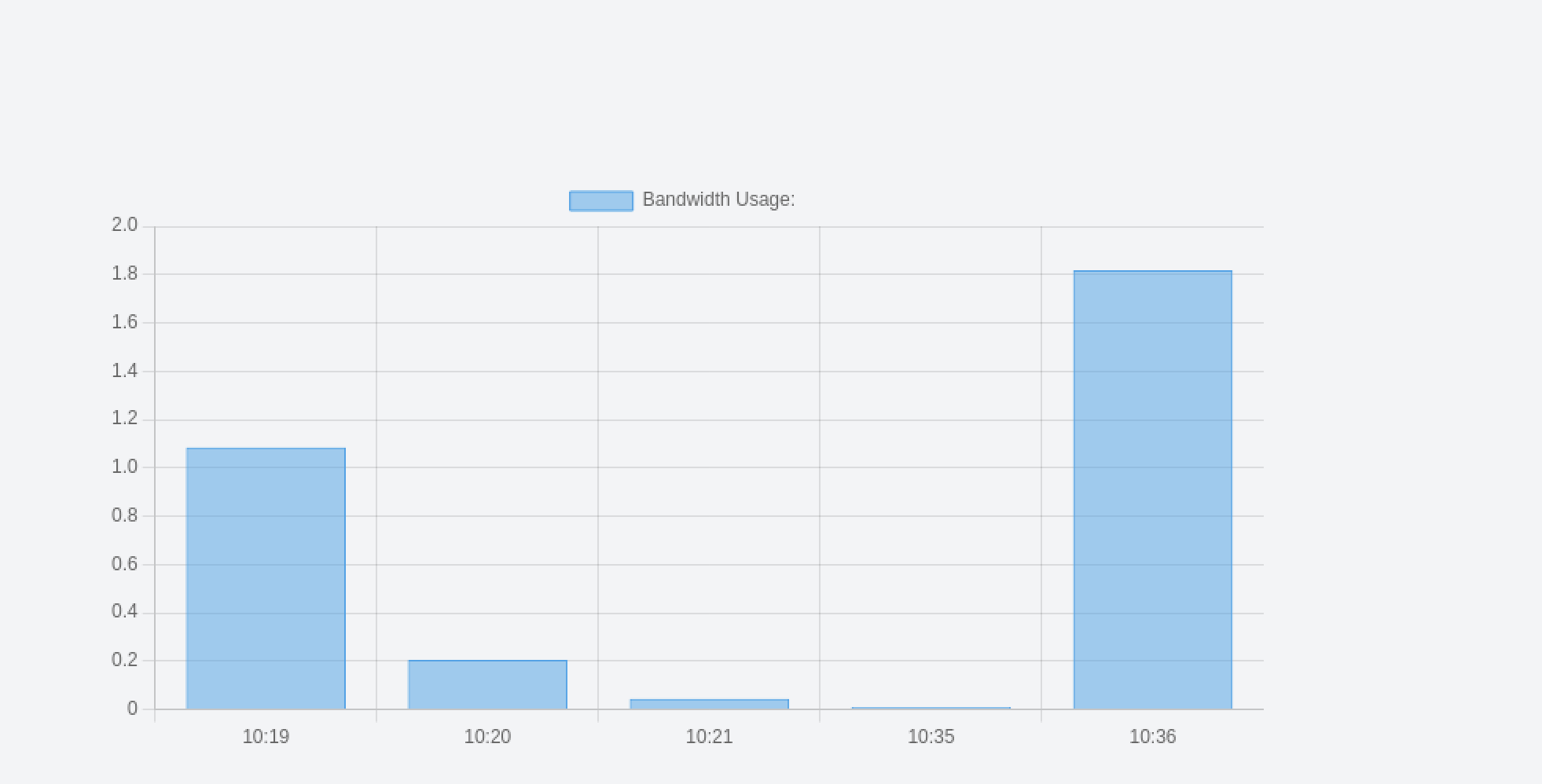
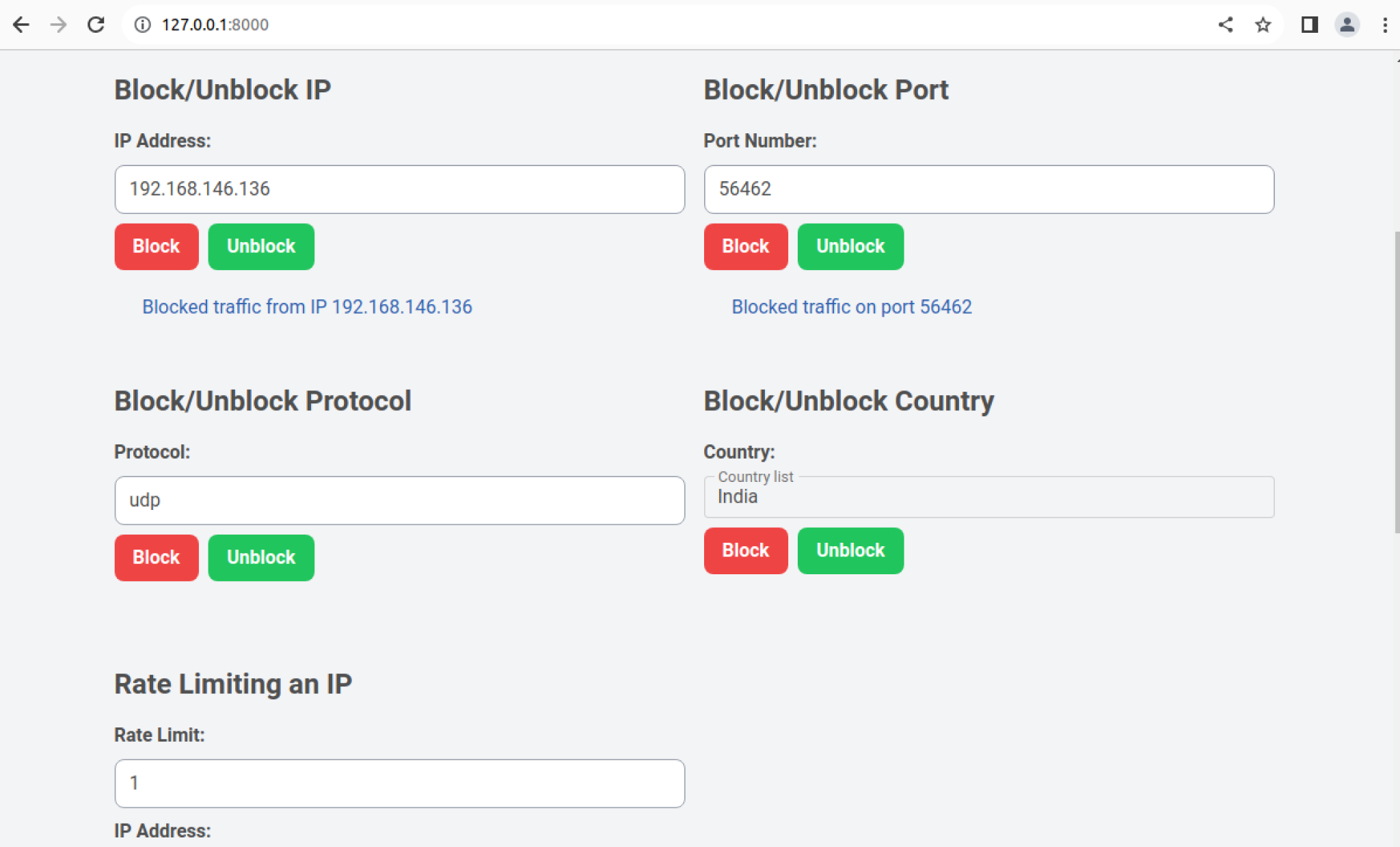
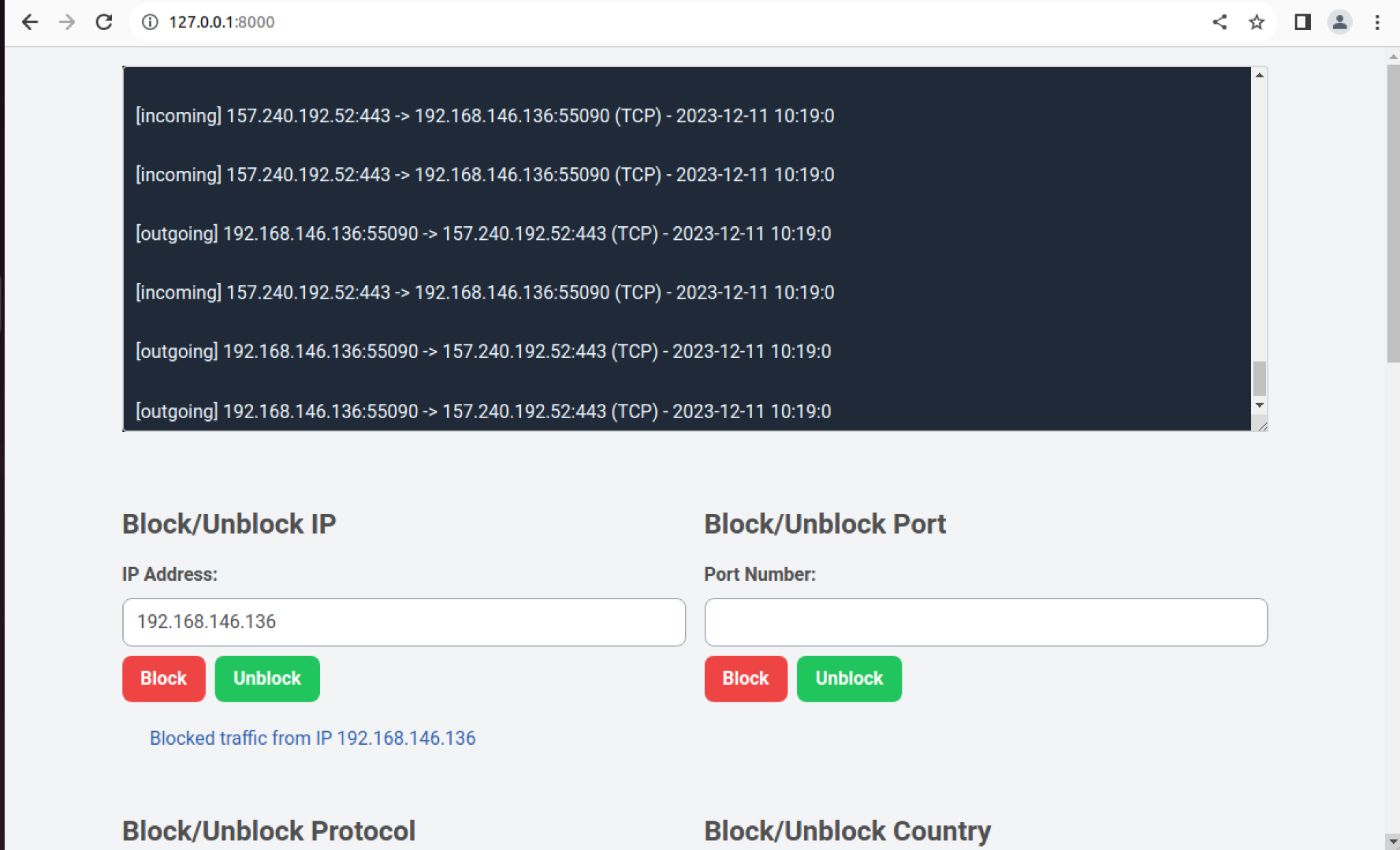
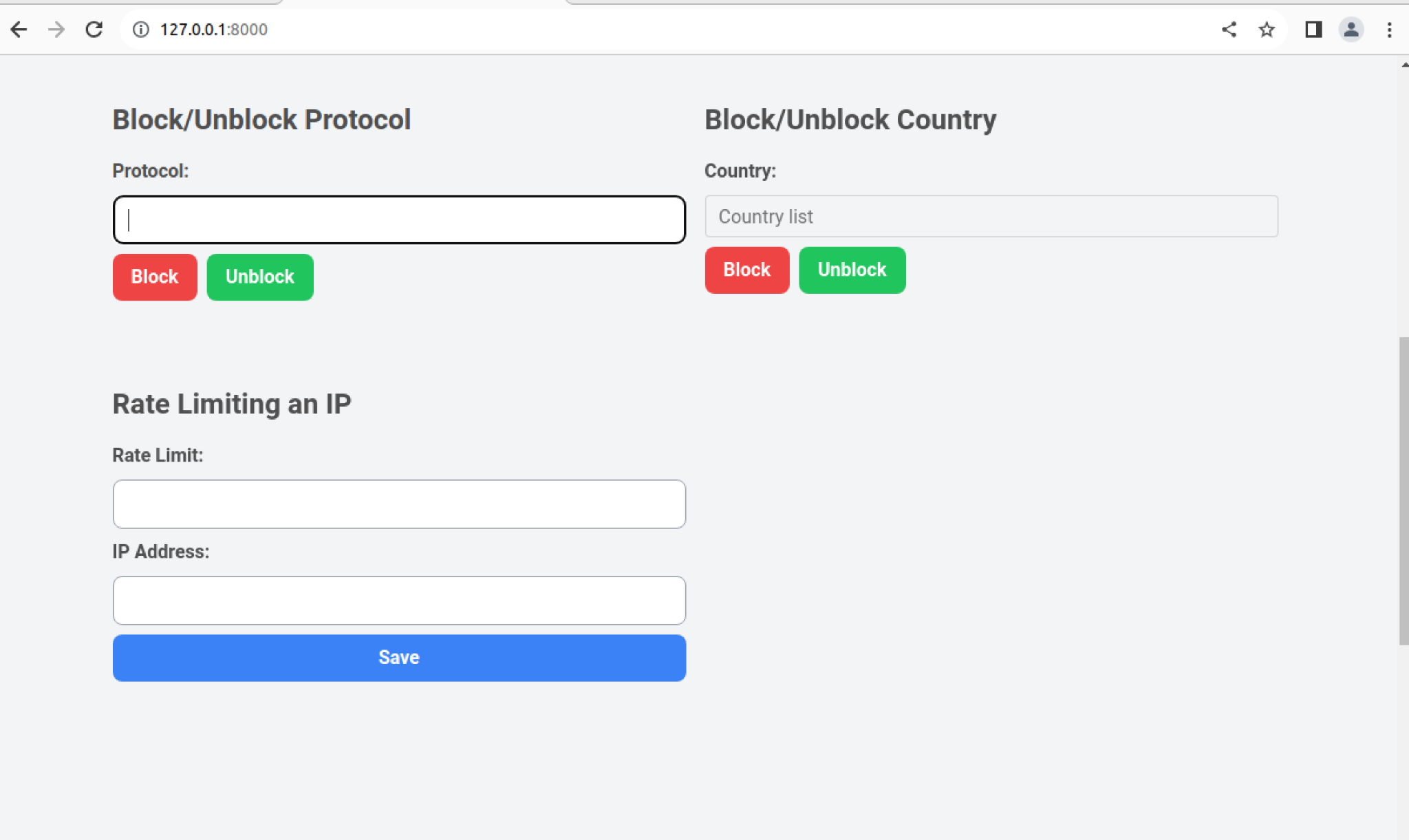
* Test Suites for Core Components
* Mocks and Stubs for External Dependencies

Interaction Flow:

* Incoming packets are captured by the Packet Capture component.
* Captured packets are passed to the Firewall Core for analysis and rule enforcement.
* The Firewall Core updates traffic counters, enforces rules (blocking or allowing traffic), and logs blocked traffic.
* The Web-Based UI interacts with the Firewall Core to display statistics, configure rules, and view logs.
* Logging handles the storage and rotation of log entries over time.

Screenshots:-





Test Coverage:-

