

# Secure System Call Interface

Course: CSE234 - Operating Systems

Name: Hemant Sihag

Registration No.: 12416315

Roll No.: 3

Under the Guidance of Faculty

Lovely Professional University

## PROJECT REPORT

Submitted in partial fulfillment of the requirements for the course CSE234 - Operating Systems

### 1. Project Overview

The Secure System Call Interface is a Flask-based web application that provides a safe and controlled interface for users to interact with simulated operating system-level system calls. The project ensures security by validating inputs, limiting operations, and providing a web dashboard to visualize system call behavior in a sandboxed environment.

### 2. Functionalities

- \*\*User Authentication:\*\* Secure login and session management for authorized access.
- \*\*System Call Simulation:\*\* Safe interface to perform simulated system calls (read, write, open, etc.) using Flask routes.
- \*\*Web Dashboard:\*\* HTML-based user interface for executing system calls and visualizing results.
- \*\*Error Handling:\*\* Prevents unsafe or undefined system calls from execution.
- \*\*Logging:\*\* Records every system call attempt for audit and debugging.

### 3. Technologies Used

- \*\*Backend:\*\* Python Flask
- \*\*Frontend:\*\* HTML, CSS, JavaScript
- \*\*Database (optional):\*\* SQLite / JSON storage for logs
- \*\*Tools:\*\* Git, GitHub, VS Code

### 4. Flow Diagram

The system flow consists of a user accessing the Flask web interface, submitting a system call request, the server validating it, executing a safe version, logging the event, and returning the output to the frontend. (Flow diagram provided in accompanying 'Flow\_Diagram.png')

## **5. Revision Tracking on GitHub**

Repository Name: project-ca2-os GitHub Link: <https://github.com/hemantsihag007/project-ca2-os> This repository contains at least seven commits demonstrating iterative development with clear commit messages and use of feature branches merged into the main branch.

## **6. Conclusion and Future Scope**

The Secure System Call Interface successfully demonstrates safe interaction with operating system-like functions through a web-based interface. Future enhancements include integrating real kernel call tracing, multi-user support, and expanding the number of simulated system calls to cover more OS functionalities.

## **7. References**

- Flask Documentation (<https://flask.palletsprojects.com/>) - Python Official Docs (<https://docs.python.org/3/>) - Linux System Calls Guide (<https://man7.org/linux/man-pages/>)