

# 23AID213 Operating Systems Project Report

By Adithya Nair, Kausik Muthukumar, P Ananthapadmanabhan Nair

## Proposed Plan

1. An election management system that can scale
2. Usage of monitoring tools such as eBPF to monitor the system's performance
3. Implement and document optimizations that improve the performance and uptime of the system.

## Research

1. Golang was chosen as it is an excellent language for the web. It provides a clean and simple abstraction for concurrency and multithreaded programming using **goroutines** (Golang, 2020).
2. **wrk(Glozer, 2025)** is a simple HTTP benchmarking tool to monitor how many requests our web server can handle.
3. **Gin(Gin-Gonic/gin, 2025)** is a web framework for Go to make it trivial to deploy HTTP API endpoints for our frontend app to interface with.

## What Has Been Done?

1. The backend for the voters has been implemented using `gin`, a performant web framework written in Golang with a SQLite database
2. A proper schema for the data has been made.

## What Needs To Be Completed?

1. The DB architecture will be swapped to something more robust and scalable as time goes on. Potential candidates include PostgreSQL and MySQL Server. In Golang, it is trivial to switch architectures by swapping the database driver agent.
2. Benchmarking using `wrk`(Glozer, 2025).
3. Usage of goroutines to implement concurrency.
4. A basic frontend for voters to cast their votes.
5. Investigating load balancing options.
6. Monitoring the system using eBPF modules.

## Project Plan

1. Switch to PostgreSQL
2. Use goroutines
3. Implement scheduling algorithms for voters - RR and FCFS
4. Deploy on a server

—

## Workload Distribution

1. Backend implementation - Adithya
2. Data Implementation - Kausik
3. Frontend Development - P Ananthapadmanabhan Nair

—

## Look into edge computing for election management system

---

### Bibliography

Gin-Gonic/Gin. (2025, March).

Glozer, W. (2025, March). Wg/Wrk.

Golang. (2020). Goroutines in Golang - Golang Docs.