Epoll HTTP Server with io_uring Upgrade

Team Members

S.No.	Name	Roll No.
1.	Tanushree Jatia	202351147
2.	Sanju Rana	202351126
3.	Srushti Chewale	202351140

Tech Stack

- Programming Language- C
- Networking APIs- Sockets, epoll, io_uring(via liburing)
- OS- Linux(Ubuntu-Dual boot)
- System Call- socket(), accept(), read(), write(), close(), listen(), bind(), fcntl()
- Benchmarking Tool- wrk/ApacheBench(ab)
- Version Control- Github+Git
- Documentation-Word

Weekly Roadmap

Week 1-2 – Research and Discussion

- Goal: Understanding the project title and output required.
- Tasks:
    - Understood the project requirements.
    - Researched epoll vs io_uring and Linux I/O basics using ChatGPT, Google, and Gemini.
    - Created the project roadmap and finalized the tech stack.
- Deliverable: Pdf containing roadmap and Tech stack.

Week 3-4 – Basic Blocking HTTP Server

- Goal: Basic socket server with blocking I/O.
- Tasks:
    - Learn Socket API basics.
    - Implement a minimal blocking HTTP Server that accepts a connection at a time.
    - Reads a request and respond to static .html file.
- Deliverable: One-client-at-a-time server responding with 'Hello, World!'.

Week 5-6 – Epoll-Based HTTP Server

- Goal: Build a concurrent server with epoll.
- Tasks:
    - Convert Sockets to non-blocking mode.
    - Register listening socket and client socket for read/write readiness.
    - Use epoll_create to create instance of epoll and then add listening socket to it to watch incoming connection.
- Deliverable: Multi-client epoll server.

Week 7 – Benchmark epoll-based server

- Goal: Measure and compare performance of epoll-based server.
- Tasks: Benchmark using wrk/ab, measure throughput, latency, CPU usage.
- Deliverable: Metrics of epoll-based server.

Week 8-9 – io_uring Server Upgrade

- Goal: Replace core I/O with io_uring.
- Tasks:
    - Initialize io_uring for accept().
    - Initialize ring buffer, submit read/write requests and collect completions.
    - Ensure server correctly responds to multiple clients.
- Deliverable: Fully io_uring-based HTTP server.

Week 10 – Benchmarking io_uring-based server

- Goal: Measure and compare performance of io_uring-based server.
- Tasks: Benchmark using wrk/ab, measure throughput, latency, CPU usage.
- Deliverable: Metrics of io_uring-based server.

Week 11 – Comparison and Report

- Goal: Compare performance of io_uring-based server and epoll-based server.
- Tasks: Compared results of both servers.
- Deliverable: Report including Benchmarking and comparison results.