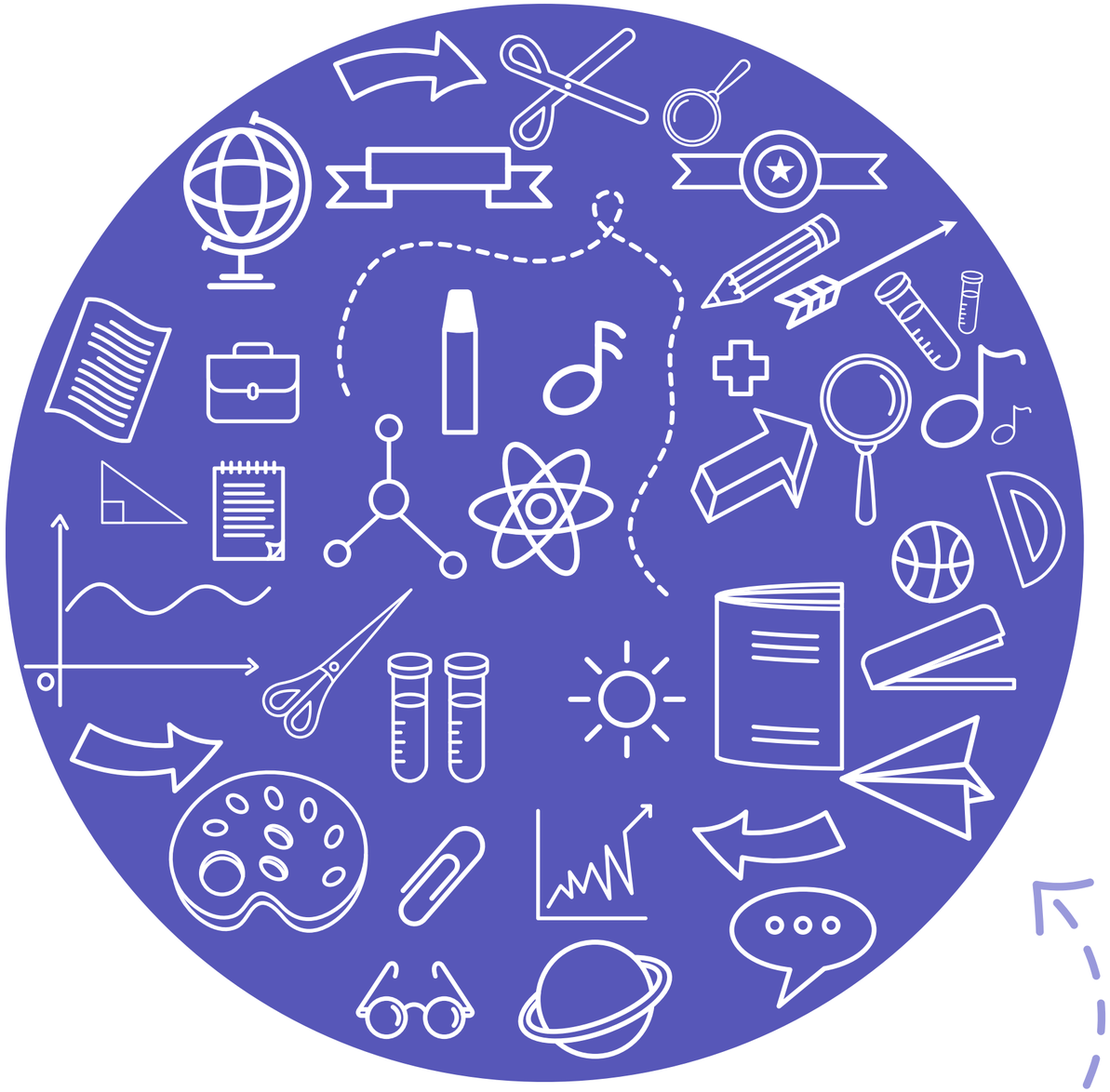
23AID212| OS | Group 9

Restaurant web application deployment using OS concepts



Buvan Subbaiah
Janani Suresh Babu Vijees
Tejas Raja Bhatt

Introduction

- In the modern hospitality industry, technology plays a crucial role in streamlining operations and enhancing customer experience.
- This project presents a Restaurant Web Application that leverages fundamental concepts of Operating Systems (OS) along with modern frontend and backend technologies.
- The application is designed to manage orders, reservations, and kitchen workflow while simulating OS principles.
- By integrating OS concepts into the design and logic of a web application, this project bridges theoretical understanding with real-world application, showcasing how low-level system behaviors can inspire efficient high-level application design.

Motivation

- The motivation behind this project stems from the need to apply theoretical OS concepts in a practical and engaging way.
- Traditional OS learning often focuses on simulations or abstract problems; however, embedding these ideas into a real-world system like a restaurant management platform offers a deeper and more intuitive understanding.
- For example, customer orders can be treated as "processes" in a scheduling queue, kitchen stations can act as "resources" managed through synchronization mechanisms, etc.
- This not only demonstrates the practical value of OS principles but also enhances software design, performance, and reliability in web applications.

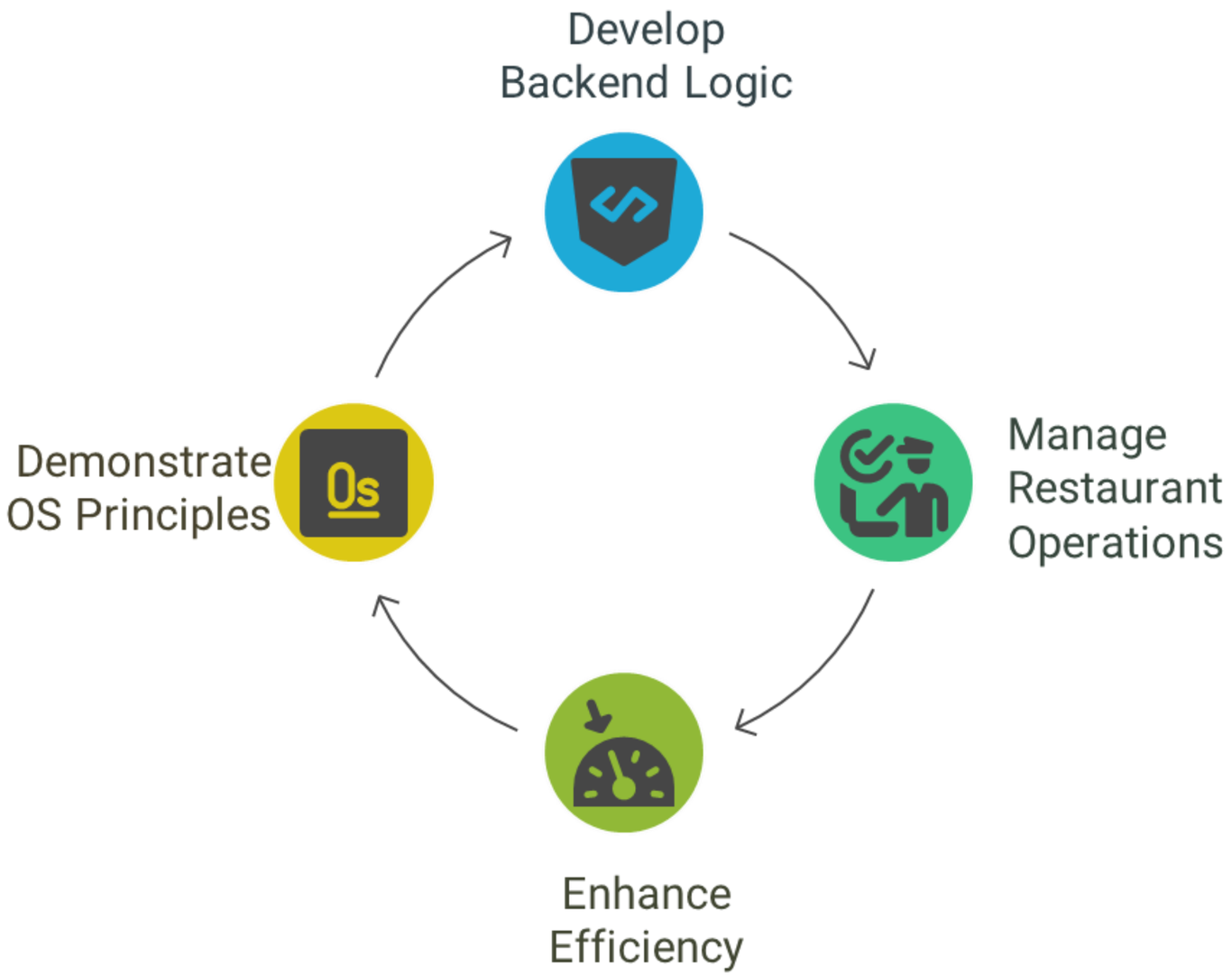
Literature survey

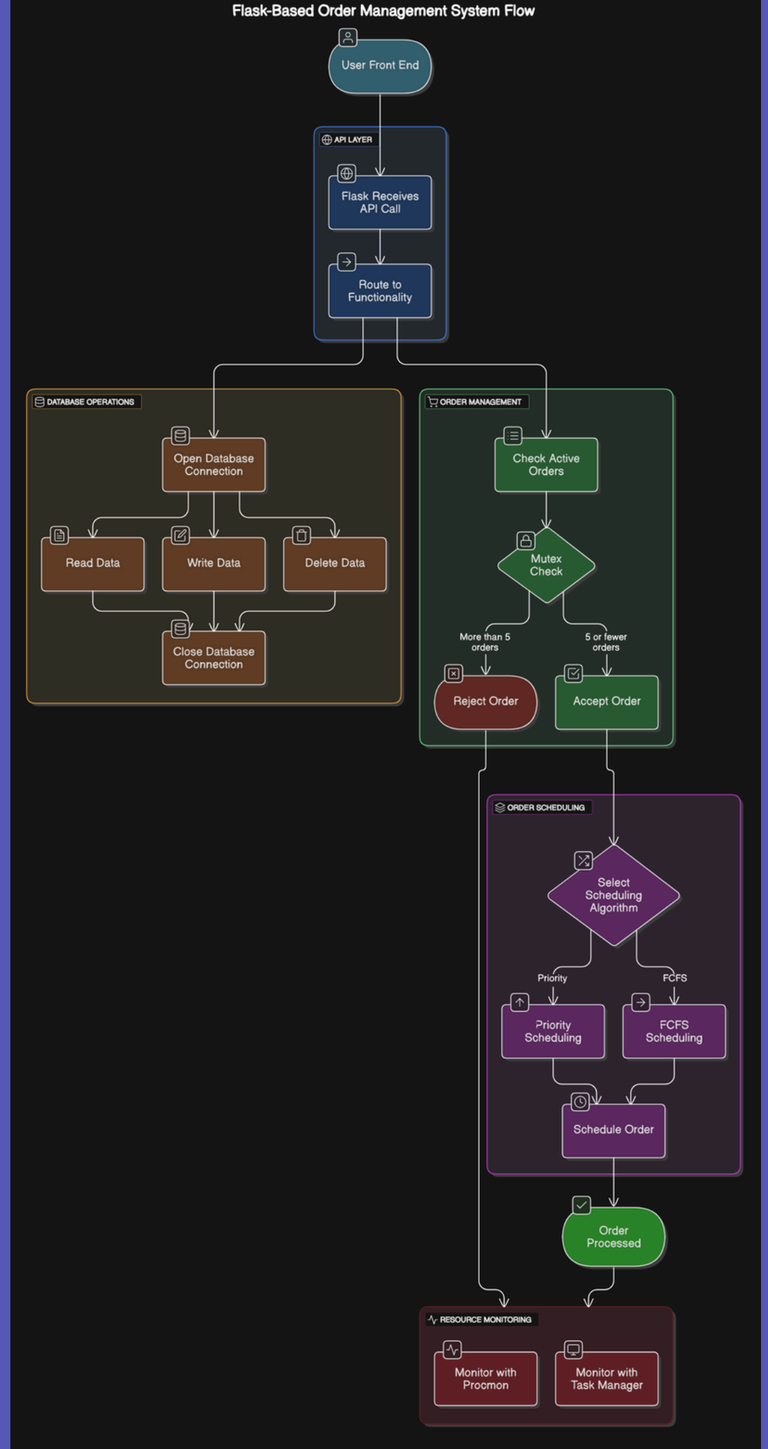
Title of Paper	System Context	Methodology	Findings
Scalability of Web-Based Electronic Commerce Systems	E-commerce systems under high load	Load balancing, caching, distributed computing	Improved scalability and response time
A Combined LIFO-Priority Scheme for Overload Control of E-Commerce Web Servers	Web server overload control	LIFO + Priority scheduling	Efficient handling of high-priority requests
Priority Mechanisms for OLTP and Transactional Web Applications	OLTP transaction systems	Priority-based scheduling	Enhanced workload and response management
Developing an E-Commerce Web App with ReactJS & Firebase	Frontend performance for e-commerce	ReactJS, Firebase UI & database optimization	Improved UI responsiveness and DB interaction
Developing a Full-Stack E-Commerce App with Next.js	Full-stack web system	Next.js SSR, efficient data fetching	High responsiveness and better performance
E-Commerce Website Using MERN Stack	MERN stack web app	API design, authentication, query optimization	Enhanced frontend-backend interaction
Saphire: Sandboxing PHP Apps	Security in PHP apps	Tailored syscall allowlists	Improved API security without loss in speed
Scheduling Algorithms in Web Cluster Servers	Web clusters	Scheduling policies for load balance	Better response and throughput
Scheduling for Distributed Web Servers	Distributed web server infrastructure	Resource allocation techniques	Efficient handling of multiple requests
Hybrid CPU Scheduling	CPU task scheduling	Round-robin + Priority hybrid	Balanced execution, prevents starvation
Memory Mgmt. for Web Apps on Multicore CPUs	Web apps on multicore processors	Memory allocation techniques	Improved performance on multicore systems
Study of Page Replacement Algorithms	Memory optimization	Comparative algorithm analysis	Better caching/page swap decisions
AI-based Memory Optimization	OS-level resource optimization	AI-driven memory management	Predictive resource allocation efficiency

Problem Statement

Optimizing a web application using OS concepts like scheduling algorithms, process handling algorithms, resource allocation algorithms, etc and making sure that users will have the best experience

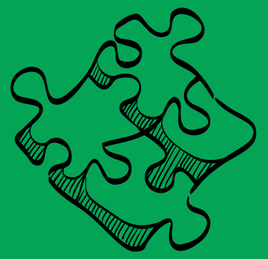
Objectives


Develop
Backend Logic
Manage
Restaurant
Operations
Enhance
Efficiency
Demonstrate
OS Principles
Os

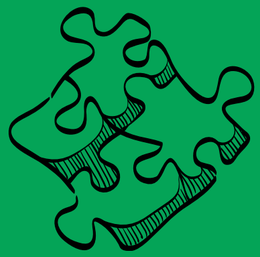

Flask-Based Order Management System Flow
User Front End
API LAYER
Flask Receives API Call
Route to Functionality
DATABASE OPERATIONS
Open Database Connection
Read Data
Write Data
Delete Data
Close Database Connection
ORDER MANAGEMENT
Check Active Orders
Mutex Check
More than 5 orders
5 or fewer orders
Reject Order
Accept Order
ORDER SCHEDULING
Select Scheduling Algorithm
Priority
FCFS
Priority Scheduling
FCFS Scheduling
Schedule Order
Order Processed
RESOURCE MONITORING
Monitor with Procmon
Monitor with Task Manager

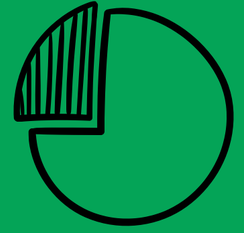
Results





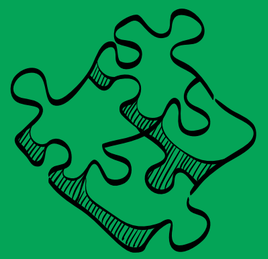




- memory management analysis:
  - 1 user login results were as follows:
    - Total Events: 325,157 system calls made by python.exe
    - Unique Paths Accessed: 2,599 different files, folders, or system objects

      - top 10 accessed paths
        - site-packages\win32\lib_pycache_: 810 accesses
        - site-packages\win32\lib: 810
        - win32\Demos_pycache_: 594
        - pythonwin\pywin\tools_pycache_: 540
        - pythonwin\pywin\debugger_pycache_: 540
        - pythonwin\pywin\framework_pycache_: 540
        - pythonwin\pywin\idle_pycache_: 540
        - pythonwin\pywin\mfc_pycache_: 540
        - pythonwin\pywin\scintilla_pycache_: 540
        - pythonwin\pywin\dialogs_pycache_: 540

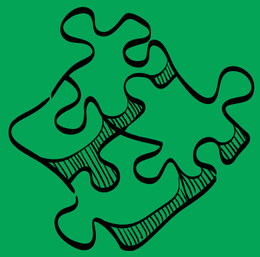
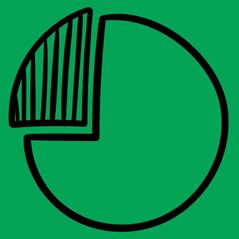

- top 10 system operations
  - IRP_MJ_CLOSE: 77,517 — file or object closures
  - CreateFile: 72,532 — attempts to open files (includes directories and devices)
  - CloseFile: 72,420 — file handles being closed
  - QueryDirectory: 43,470 — listings of directories (e.g., for module loading)
  - WriteFile: 83 — actual writes
  - ReadFile: 82 — actual reads
  - QueryOpen, Process Profiling, LockFile — low but present

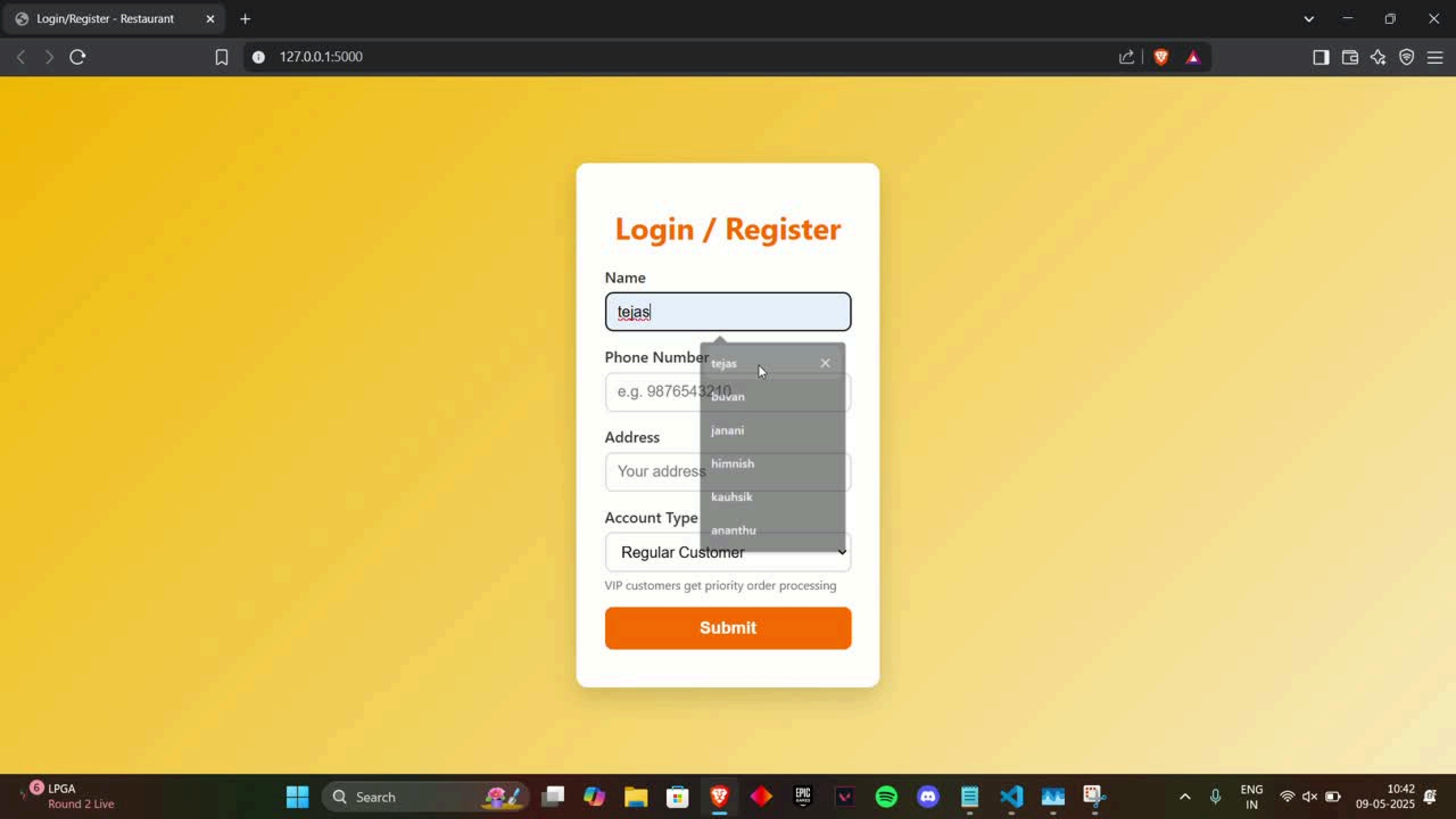
- top 10 result types (system call results)
  - SUCCESS: 310,867 calls completed successfully
  - NO MORE FILES: 14,040 — often part of directory scans
  - NAME NOT FOUND: 129 — attempts to access missing files
  - FAST IO DISALLOWED: 96 — fallback to slower I/O paths
  - BUFFER OVERFLOW, END OF FILE, and others — minor but expected edge cases

- Memory related operations

VirtualAlloc
VirtualFree
VirtualProtect
CreateFileMapping
MapViewOfFile
UnmapViewOfFile
ReadVirtualMemory
WriteVirtualMemory
Section
Memory

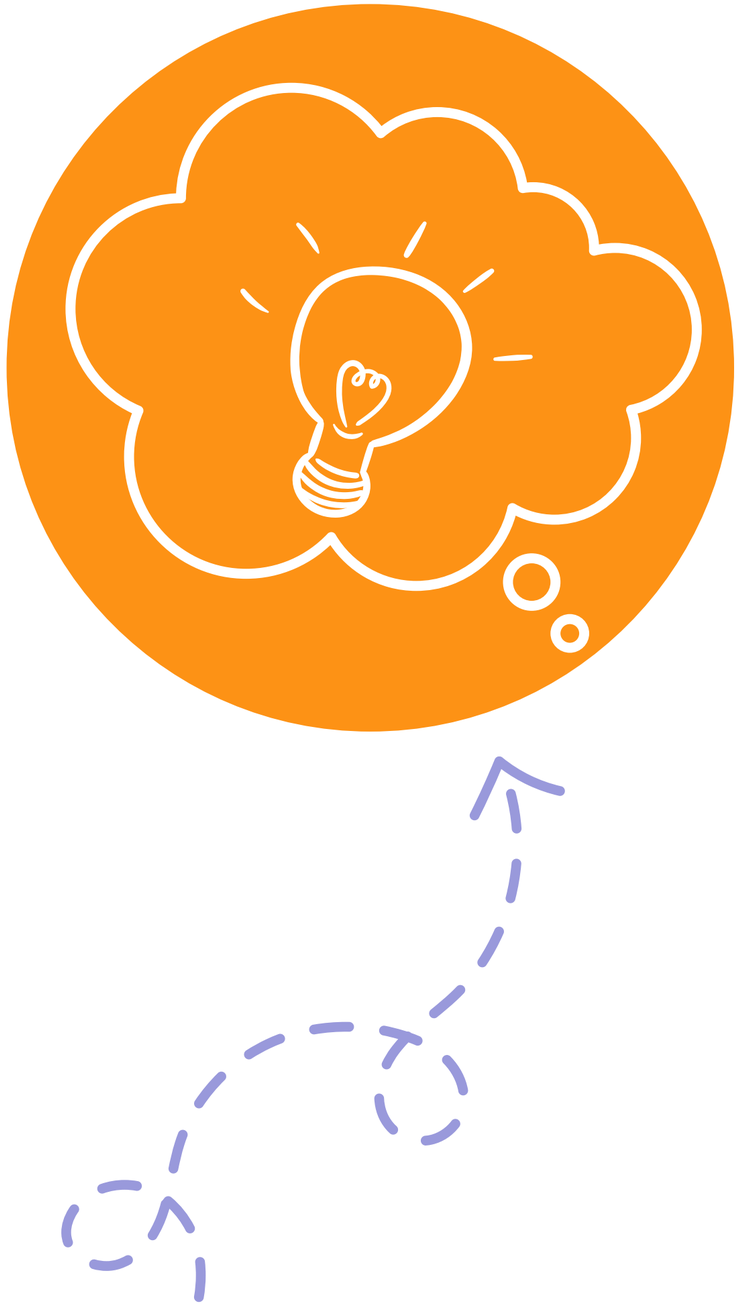



Login/Register - Restaurant
127.0.0.1:5000
Login / Register
Name
tejas
Phone Number
e.g. 9876543210
tejas
duvan
janani
himnish
kauhsik
ananthu
Address
Your address
Account Type
Regular Customer
VIP customers get priority order processing
Submit
LPGA
Round 2 Live
Search
ENG
IN
10:42
09-05-2025

Conclusion

- This project successfully demonstrates the application of core Operating System concepts in the development of a Restaurant Web Application.
- By implementing scheduling algorithms such as First-Come-First-Serve (FCFS) and Priority Scheduling, the system efficiently manages customer orders and task queues in a manner analogous to OS-level process scheduling.
- Process handling logic ensures smooth operation coordination between different modules like order processing and kitchen management, while resource allocation algorithms simulate the optimal usage of limited resources like kitchen stations or delivery slots.
- Overall, the project provides a functional and educational platform that bridges theoretical OS concepts with practical software development.



Future work

- To transform this project from a conceptual and educational prototype into a real-world application usable by restaurants and customers

- Introduce more advanced CPU scheduling algorithms to better simulate real-time and multitasking environments.

THANK YOU