আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম
الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

Department of Computer Science and Engineering

Project Proposal

Course Code: CSE-3632

Course Title: Operating Systems Lab

Project Title: Smart OS Simulator

Members:

1. Name: Abdul-Al-Muheet
   ID: C223198
2. Name: MD. Shahariat Hossen
   ID: C231051
3. Name: Mohammed Sami Hasan
   ID: C231053
4. Name: Afrin Hasan Safin
   ID: C231062

Supervisor: Mohammed Arfat
Lecturer.
Department of CSE, IIUC.

Date of Submission: 05/11/2025

Remarks:

Title: Smart OS Simulator

Introduction:

Operating System (OS) concepts such as process scheduling and memory management are often difficult for students to understand through theory alone. Although classical algorithms explain how decisions are made, students frequently struggle to identify which algorithm performs best under different workloads and system conditions.

To address this issue, this project proposes an enhanced Smart OS Simulator, a game-based educational web application that allows users to act as an operating system while also receiving Machine Learning (ML)–based guidance. The simulator enables users to manage CPU scheduling and memory allocation using classical algorithms such as FCFS, SJF, SRJF, Round Robin, Priority, and RR+Priority, along with memory allocation strategies like First Fit, Best Fit, and Worst Fit.

In addition to manual selection, the system incorporates an ML-based recommendation module that analyzes process characteristics (arrival time, burst time, priority, memory size, etc.) and suggests the most suitable scheduling or memory allocation strategy for a given scenario. This hybrid approach combines traditional OS algorithms with intelligent decision support, making the learning process more adaptive and insightful.

Objectives:

1. To design and develop an interactive simulator that visualizes CPU scheduling and memory management operations.
2. To allow users to apply, compare, and evaluate different scheduling and memory allocation algorithms.
3. To integrate a Machine Learning model that recommends optimal algorithms based on system workload and process characteristics.
4. To provide real-time feedback, performance metrics, and scores to assess user decisions and ML recommendations.
5. To enhance conceptual understanding of OS principles through a game-based and intelligent learning experience.

- ## Proposed Machine Learning Features

The Machine Learning component will act as an intelligent assistant, not a replacement for classical algorithms.

ML Capabilities

- Analyze input features such as:
    - Number of processes

    - Arrival time and burst time
    - Priority values
    - Time quantum
    - Memory requirements
- Predict or recommend:
    - The most efficient CPU scheduling algorithm
    - The best memory allocation strategy for minimal fragmentation

Learning Approach

- Supervised learning using historical simulation data
- Possible algorithms:
    - Decision Tree
    - Random Forest
    - K-Nearest Neighbors (KNN)

Educational Value

- Helps students understand why one algorithm performs better than another
- Encourages experimentation by comparing manual choices vs ML suggestions

Materials

Hardware Requirements

- Computer or laptop with modern browser support
- Minimum 4 GB RAM
- Dual-core processor or higher

Software & Tools

- Frontend: React.js, Tailwind CSS, JavaScript
- Visualization: Chart.js / HTML Canvas
- Machine Learning: Python, scikit-learn (model training)
- Backend (Optional): Flask API for ML predictions
- Development Tools: VS Code, Node.js, GitHub

Conclusion:

The **Smart OS Simulator** aims to bridge the gap between theoretical OS concepts and practical understanding. By transforming abstract scheduling and memory management ideas into an interactive, gamified simulation, this project provides a hands-on learning experience that encourages experimentation, critical thinking, and engagement. The outcome will be a visually intuitive, educational, and self-contained tool that helps students grasp the inner workings of an operating system in a fun and effective way.