International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 02 | Feb - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Interactive Animation-Based Learning Platform for Operating System and AI/ML Concepts

Author 1: Damayanti D Pawar

Information Technology, K.K Wagh Polytechnic,Nashik

Author 2: Shirin Atul Gaikwad

Information Technology, K.K Wagh Polytechnic,Nashik

Author 3: Antara Vijay Raut

Information Technology, K.K Wagh Polytechnic,Nashik

Author 4: Samruddhi Pankaj Pawar

Information Technology, K.K Wagh Polytechnic,Nashik

ABSTRACT - This project focuses on creating an interactive animation-based learning platform that helps students understand tough topics in Operating Systems and Artificial Intelligence/Machine Learning in an easy and fun way. The platform uses visual animations, short interactive lessons, and quizzes to explain difficult ideas in a more engaging format.With this, learners can watch how algorithms, processes, and models work instead of reading about them from textbooks. The platform uses animations and simulations to show how real-time tasks like OS scheduling and memory management, as well as AI/ML concepts like neural networks and data processing, work in practice.The platform is made for self-paced learning and aims to help both beginners and advanced learners gain a clear understanding of the topics. In the future, this platform plans to make technical education more interactive, easier to access, and more enjoyable for everyone.

Key Words: Algorithm Visualization, Interactive Pedagogy, Operating Systems, Machine Learning, Real-Time Simulation, Process Scheduling, Neural Networks, Memory Management, Self-Paced Learning, Educational Technology

1.INTRODUCTION

In today's fast-moving technology world, it's important for IT students and professionals to understand complex technical topics like Operating Systems and Artificial Intelligence/Machine Learning algorithms. However, traditional ways of learning, such as using books and lectures, usually don't provide the kind of visual and interactive understanding these topics need. Learners often find it tough to see the link between the theory and real-life applications, which can reduce their interest and understanding.To address this, this project proposes an Interactive Animation-Based Learning Platform that uses animations, simulations, and interactive assessments to simplify complex OS and AI/ML concepts. The platform is designed so students can learn by seeing and interacting, making education more engaging and learner-focused. Teachers can also use this platform to demonstrate difficult topics in a more visual and productive way during classroom sessions.Overall, this project aims to modernize education by using technology to provide an innovative, interactive, and learner-centered solution for understanding Operating Systems and AI/ML topics. It tries to make learning not only more effective but also more enjoyable and suited for the growing needs of digital education.

2.EXISTING SYSTEMS AND THE LEARNING GAP

The standard methods for teaching complex technical subjects like Operating Systems and Artificial Intelligence/Machine Learning rely mostly on textbooks and traditional lectures. These methods often present processes and algorithms, like memory management or neural network execution, as abstract theories that are hard to visualize in real time. Although some small-scale web games, like "Flexbox Froggy," use friendly characters and game-like designs to teach coding skills, other technical topics still lack similar interactive approaches. As a result, learners often struggle to connect theoretical knowledge with practical applications, leading to reduced interest and confusion.

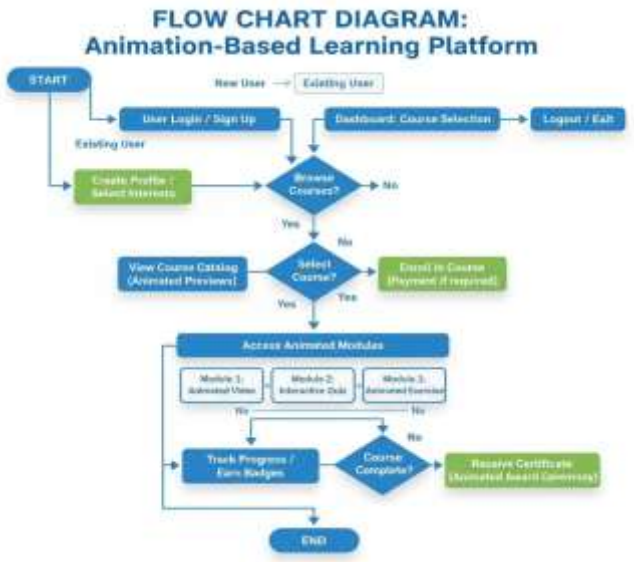
3.THE PROPOSED SOLUTION

This project, the "Interactive Animation-Based Learning Platform," addresses this gap by moving from static learning to dynamic visualization. Unlike traditional systems that focus on reading theory, this platform lets learners see how algorithms and processes work through visual animations and simulations. The main difference is the interactivity — it moves beyond passive learning to an approach where students control how algorithms work. It includes short interactive lessons, quizzes, and level-based learning, similar to modern educational games, to make technical education not only effective but also fun and accessible for both beginners and advanced learners.

4.SYSTEM WORKING :

This project works through a structured six-step process. It starts by identifying which OS and AI algorithms students find hardest to imagine. Once those core concepts are selected, the team creates visual storyboards to show how each step of the algorithm will be displayed graphically. During the development phase, programming tools like JavaScript or Python are used to build the logic and animation modules, making the interface interactive. The system is then tested to ensure the animations accurately show how the algorithm works. Finally, user feedback is gathered to improve the visuals and controls, ensuring the platform delivers maximum educational value.

Fig: Flow Chart Diagram



Flow Chart Explanation:

The process of this platform starts at the "Start" stage, where users either sign up as new members or log in as existing users. New users are guided to create a profile and choose their interests to personalize their experience. Once logged in, users reach a central dashboard where they can explore the course catalog, which includes animated previews of the available modules. If a user selects a course and joins, they gain access to the "Animated Modules" section. This part includes three main components — animated videos for teaching concepts, interactive quizzes for assessment, and animated exercises for practical work. As users interact with these modules, the system tracks their progress and provides badges. Upon completing the course successfully, the user receives a certificate during an "Animated Award Ceremony," and the process ends there.

5.METHODOLOGY

5.1. Problem Identification: Identify that learners find it hard to grasp how OS algorithms work internally because of the lack of visual teaching methods.

5.2. Data and Algorithm Selection: Choose core algorithms (like scheduling, memory management, or AI-based algorithms) that align with the learning goals.

5.3. Design Phase: Create a visual storyboard and decide how each algorithm's step will be shown graphically.

5.4. Development Phase: Use programming tools such as Java, JavaScript, or Python to build the logic and animation features. Every algorithm has an interactive interface, allowing users to control how the execution works.

5.5. Testing Phase: Check that the animations correctly illustrate how the algorithm operates and help learners gain clarity.

5.6. Evaluation and Improvement: Collect user feedback to improve the visuals, controls, and explanations for better educational impact.

6.FEATURES/KEY POINTS

6.1. Overcoming Abstract Concepts: Students often find it hard to imagine how operating systems and AI/ML work. Our platform uses visual animations and simulations to make these complex ideas easier to understand.

6.2. Connecting Theory to Reality: Many students find it hard to apply what they learn to real situations. Our system includes real-time execution modules that show how OS scheduling and memory management work in action, helping learners see exactly how these processes function.

6.3. Enhancing Engagement: Traditional learning methods can be boring. Our project counters this by using a game-like layout inspired by tools like Flexbox Froggy. It includes a friendly mascot and level-based progress to keep learners motivated.

6.4. Empowering Educators: Teachers often struggle to explain technical topics using traditional methods. This platform acts as a visual aid, making classroom sessions more productive and helpful for students.

7.SYSTEM ARCHITECHTURE

7.1. Frontend (The Visual Interface):

1. Technologies: HTML, CSS, and JavaScript.

2. Animation Engine: Special libraries like GSAP or Framer Motion are used to create smooth and efficient transitions for algorithm steps.

3. Design Philosophy: The user interface has a clean, minimal style with playful icons and a friendly mascot, helping to reduce the intimidating feel of technical subjects.

7.2. Backend (The Logic and Server): The backend manages the core functions of the platform, including user data and module progression.

1. Technologies: Node.js or Python (using frameworks like Flask or Django).

2. Functionality: It handles user login and sign-up, tracks course enrollment, and saves user progress like earned badges.

7.3. Database (Data Storage) :

1. Technologies: MySQL (Relational) or MongoDB (NoSQL).

2. Role: It stores user profiles, quiz results, and the database structure that maps students to their courses.

7.4. Core Features and User Flow:

The platform follows a clear learning flow to help users learn effectively:

1. Interactive Dashboard: Once logged in, users can browse a course catalog with animated previews of lessons.

2. Learning Modules: Each lesson includes 4 parts:
1.Animated Videos: To explain concepts.
2.Interactive Exercises: Let users control how algorithms work.
3. Interactive Quizzes: To test understanding
4.. Gamification: As users complete modules, they track their progress and earn badges, with an animated ceremony for certification.

8. CONCLUSIONS

This interactive, animation-based learning platform addresses a major issue in traditional technical education: the challenge of visualizing abstract and real-time system processes. By replacing static theory with dynamic simulations, the platform helps users understand complex algorithms like OS process scheduling and neural network workflows.Through self-paced modules, interactive quizzes, and visual storytelling, the project creates a more engaging and effective learning environment for both new and experienced students. Ultimately, this initiative helps modernize digital education, making it easier, more enjoyable, and better aligned with industry needs.

9.ACKNOWLEDGEMENT

We would like to thank everyone who contributed to the intellectual foundation and support for our project, "Interactive Animation-Based Learning Platform for Operating System and AI/ML Concepts." We are deeply grateful to the researchers whose work formed the basis of our literature review, especially Francisco S. Melo, Samuel Mascarenhas, and Ana Paiva for their insights on embedding machine learning in interactive serious games. The studies by Huang Li and colleagues on the use of visualization in ML learning were key in shaping our approach to data processing modules. Additionally, the research on the effectiveness of animations in teaching data structures and the self-regulated learning models from Project OSCAR provided a proven framework for our technical education approach.

10.REFERENCES

10.1. Francisco S. Melo, Samuel Mascarenhas, Ana Paiva, 2018: This paper discusses how machine learning can be embedded into interactive learning systems (serious games, tutoring systems). Link: https://doi.org/10.17083/ijsg.v5i3.256

10.2. Huang Li, Shiaofen Fang, Snehasis Mukhopadhyay, Andrew J Saykin, Li Shen: This study shows how visualization helps in ML learning, especially with limited data. Link: PMC version: https://pmc.ncbi.nlm.nih.gov/articles/PMC6499624/

10.3. "The efficacy of animation and visualization in teaching data structures: a case study": This research looks at using animations and visual tools to teach technical computing topics like data structures.

10.4. "Project OSCAR: Open-Source Animations Repository To Foster Self-Regulated Learning" :This article describes an open web repository of animations and simulations for learning.

10.5. "Design of AI-Based Personalized Interdisciplinary Animated Micro-Courses" (2024): This paper covers AI and animated micro-courses for personalized learning.

10.6. "Interactive online learning method for students based on artificial intelligence" :A recent article about an AI-driven interactive learning platform.