MINI PROJECT

Team:-

Vaishnavi Rahamtkar
(AM21014)
Team Leader

Aboli Wankhade
(AM21004)
Member

Tanushree Sarode
(AM21032)
Member

Sakshi Mantri
(AM21038)
Member

CONTENT:-

- Problem Statement
- Introduction
- Objective
- Methodology
- Expectations
- Challenges
- Outcome
- Research Paper

PROBLEM STATEMENT

Ideate and implement a system to enhance the quality of education in Emerging Technology.

Introduction

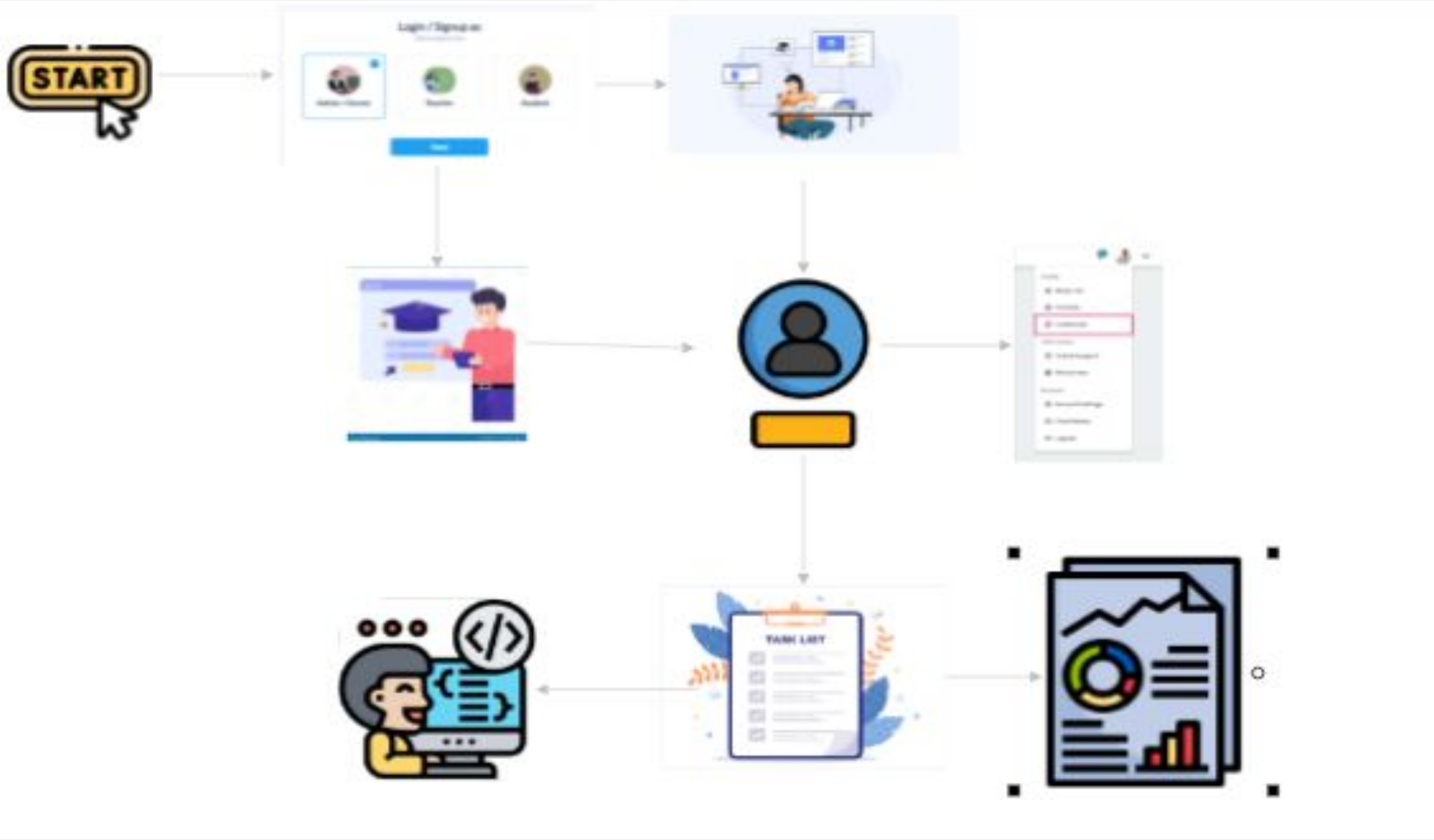
- Our goal is to enhance the quality of education in emerging technology by implementing a comprehensive system that includes a user-friendl online compiler for both students and faculty.
- This system will have distinct roles for admin and users, creating a seamless platform.
- Introducing an advanced online compiler for students and facult to elevate education in emerging technology.
- Admins have control over system management, while users benefit from a online compilation experience. This initiative aims to efficient learning.

Objectives Of The Project

Develop and implement an online education system.

1. Develop an online compiler for students and faculty to enhance practical coding skills in emerging technologies.
2. Create a user-friendly website with distinct roles for admin and users.
3. Enable seamless interaction between students and faculty through the online compiler platform.
4. Facilitate real-time code compilation, execution, and debugging for a variety of programming languages.
5. Provide a collaborative learning environment by allowing users to share code snippets and solutions.

Methodology

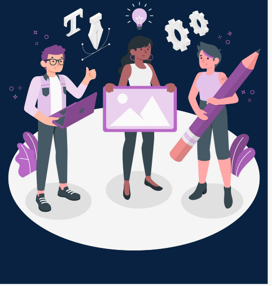


- **Login Page**:
  - Faculty and students log in using their respective credentials.
  - Separate login interfaces for faculty and students.
- **Faculty Dashboard**:
  - Faculty can create new tests, manage existing tests, and view test results.
  - Interface for assigning tests to specific students or groups of students.
- **Test Creation**:
  - Faculty can create tests by specifying questions, options, and correct answers.
  - Ability to set time limits, assign test to specific classes or groups, and customize test parameters.
- **Student Dashboard**:
  - Students can view assigned tests and access them for solving.
  - Clear interface displaying test instructions and questions.

- **Test Solving**:
  - Students can attempt tests within the specified time limit.
  - Interface should allow for easy navigation between questions and submission of answers.
- **Automatic Result Generation**:
  - System automatically evaluates student responses against correct answers.
  - Generates test scores and provides immediate feedback to students upon completion.
- **Score Submission**:
  - After completing the test, students can submit their answers for evaluation.
  - Scores are automatically recorded and submitted to the faculty.
- **Result Viewing**:
  - Faculty can view test results for each student, including scores and detailed breakdown of answers.
  - Ability to export results for record-keeping or further analysis.

Expectations

- Students expect an online compiler to provide a user-friendly interface.
- Support for multiple programming languages.
- Efficient code execution, helpful error messages and integration with version control systems.
- Faculty expect features for code review, plagiarism detection, and classroom management, fostering a collaborative and productive learning environment for programming courses.



Outcome

For Students:

- **Practice and Skill Enhancement:** Access to diverse coding challenges improves coding skills and problem-solving capabilities.

- **Instant Feedback:** Real-time feedback on code submissions aids learning and iterative improvement.

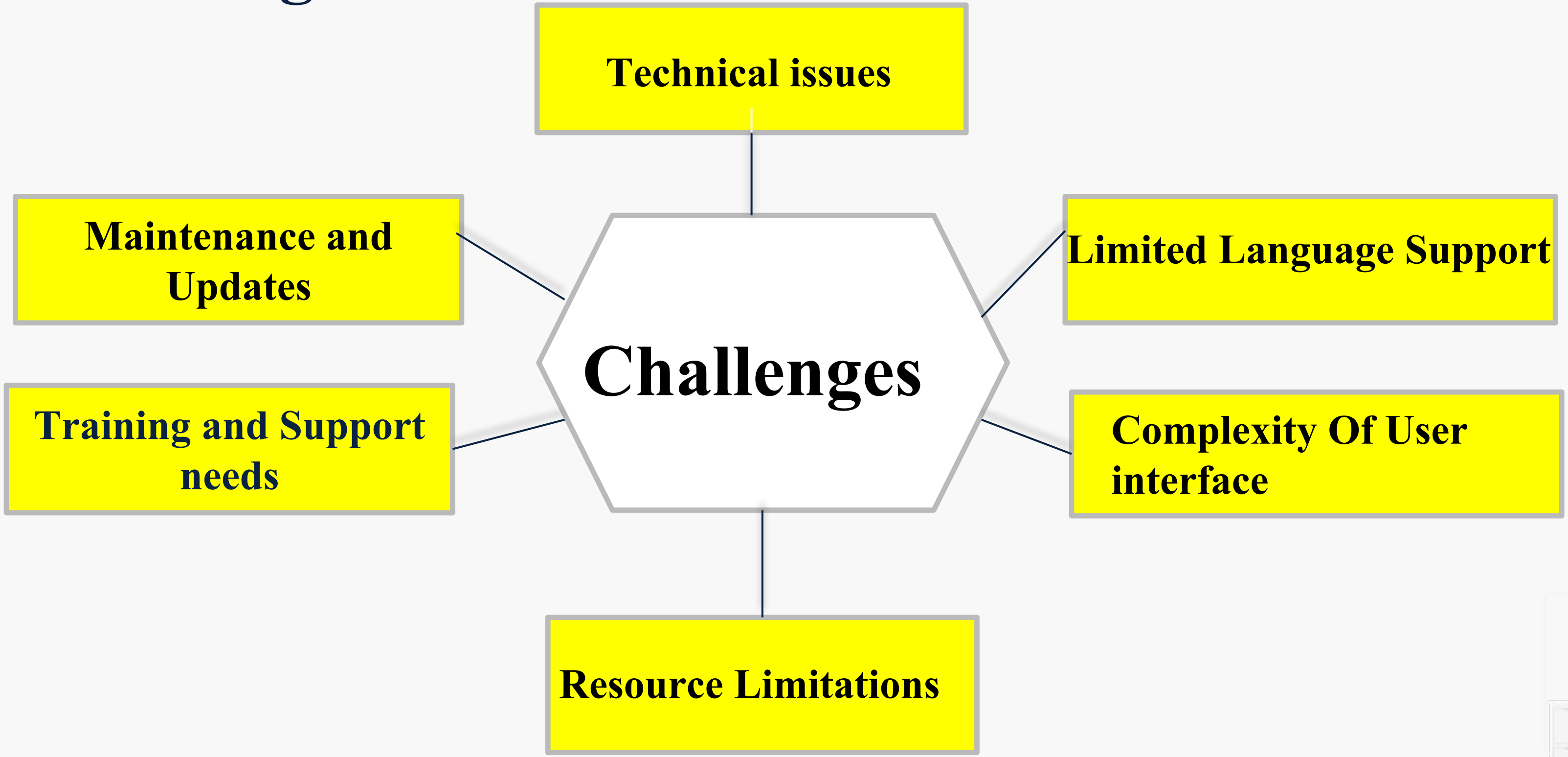
- **Competition Motivation:** Coding contests and challenges foster competition, motivating students to excel.

- **Structured Learning:** Curated learning paths and tutorials facilitate systematic knowledge building in specific programming domains.

For Faculty:

- **Customized Assignments:** Faculty can create tailored coding assignments and assessments.
- **Automated Grading:** Grading tools automate evaluation, providing instant feedback and saving faculty time.
- **Performance Tracking:** Analytics dashboards track student performance, enabling targeted instruction.
- **Community Interaction:** Online forums foster faculty-student engagement for questions and support beyond class hours.

Challenges:-



Challenges

Technical issues

Maintenance and Updates

Limited Language Support

Training and Support needs

Complexity Of User interface

Resource Limitations

Technical issues:

Code execution security, language support, performance optimization, error handling, concurrency, scalability, user interface design, data management,deployment, user management.

Limited language support:

Supporting numerous programming languages poses challenges, requiring prioritization based on user demand and technical feasibility to ensure optimal functionality.

Complexity of user interface: A user-friendly interface is crucial for seamless interaction, prompting the need for intuitive design and clear navigation to mitigate user confusion and frustration.

Resource Limitations:

Resource limitations impact online compiler development. Optimize code execution, manage servers efficiently, and consider cloud solutions for scalability.

Maintenance and updates: Regular maintenance and updates are crucial for ensuring system stability, security, and performance, requiring proactive monitoring, timely patching, and continuous improvement efforts.

Research Paper Related With Project

- https://www.academia.edu/32529973/Review_Paper_on_Online_Java_Compiler
- https://www.researchgate.net/publication/323832187_Java_Code_Engine_for_Online_Code_Compilation
- https://prolangs.cs.vt.edu/rutgers/reading/papers/chen_oopsla.pdf
- https://www.irjet.net/archives/V4/i3/IRJET-V4I3122.pdf

THANK YOU !