Project Proposal: Cloud-Based Programming IDE for Computing Class

Introduction

The goal of this project is to develop a cloud-based Integrated Development Environment (IDE) for a computing class. Traditional IDEs often require local installations, leading to inconsistencies in development environments. This IDE will eliminate the need for local installations, provide a seamless programming experience, and integrate collaboration and AI-driven features to enhance learning. By leveraging cloud technology, real-time tools, and AI-driven problem-solving, the IDE aims to streamline the programming experience and foster a collaborative learning environment.

Nature of the Problem:

Computing classes require an efficient, accessible, and collaborative programming environment. Students face difficulties with local software installations, collaboration, real-time code sharing, and GitHub integration.

Major Functions Required:

- Cloud-based IDE to facilitate programming from any device.
- User authentication (log in/sign up) to manage projects.
- Project creation, saving, and execution.
- Error messaging for debugging.
- GitHub integration for project management.
- Collaboration tools for real-time coding and screen sharing.
- AI-powered programming challenges and auto-testing.

Key Information and Data:

- User authentication details.
- Project files and code history.
- Error logs and test results.
- GitHub repository details for integration.

Organization Details:

This project is developed for a computing class to provide students and instructors with a seamless and interactive coding environment. The goal is to simplify the learning process by reducing software setup overhead and enabling effective collaboration.

Stakeholders:

- **Computing class instructors** – Ensure a smooth learning experience and provide assessments.
- **Students enrolled in the course** – Gain hands-on coding experience with real-time feedback and collaboration.
- **IT administrators** – Manage the platform and ensure smooth operation.
- **Developers working on the IDE** – Implement features and maintain the system.

Project Goals:

- Develop a cloud-based IDE to enable coding from anywhere.
- Provide a seamless, consistent development environment.
- Integrate real-time collaboration features for teamwork.
- Ensure GitHub integration for project version control.
- Implement AI-driven programming challenges for self-assessment.

Major Functions and Attributes:

General Features:

- **User Authentication**: Sign up/log in functionality to access projects.
- **Project Management**: Create, save, and run projects.
- **Error Messaging**: Provide debugging assistance.

Collaboration Features (Team 1):

- **Real-time editing**: Multiple users can work on the same project.
- **Screen sharing**: Integrated tool for peer collaboration.

GitHub & AI Features (Team 2):

- **GitHub Integration**: Save and load projects from GitHub.
- **AI-Powered Coding Challenges**:
    - Assign programming tasks.
    - Allow users to accept and start coding.
    - Automated testing and submission.

Data Management:

- **User Data**: Credentials, permissions.
- **Project Data**: Code files, execution history.
- **Error Logs**: Debugging insights.

- **GitHub Repository Data**: Linked projects, commit history.
- **AI Challenge Data**: Assigned problems, user submissions, test results.

Proposed Technologies:

- **IDE Platform**: Eclipse Che (Cloud-based, Kubernetes-native).
- **Backend**: Node.js & Firebase for real-time collaboration.
- **Frontend**: React.js for UI/UX.
- **Database**: MongoDB for user and project data.
- **Version Control**: GitHub integration for project management.
- **Collaboration Tools**: WebRTC for screen sharing.
- **AI & Testing**: Python-based auto-testing system.

Team Responsibilities & Milestones:

- **Team 1** (Editing & Collaboration):
    - Develop real-time code editing.
    - Implement screen-sharing functionality.
- **Team 2** (GitHub & AI Features):
    - Develop GitHub integration for project storage.
    - Implement AI-driven coding challenges and auto-testing.

Potential Challenges:

- Ensuring smooth real-time collaboration without lags.
- Managing secure and efficient GitHub authentication.
- Designing an intuitive AI challenge workflow.

Expected Outcomes & Conclusion

- A fully functional cloud-based IDE tailored for a computing class.
- Enhanced collaboration through live coding and screen sharing.
- GitHub integration for seamless project version control.
- AI-assisted programming challenges for interactive learning.

Team members:

1. Godwin Idowu
2. Yaw Oppong-Krampah
3. Wisdom Akanwe
4. Braeden Singleton
5. Calvin Deka
6. Connor Moss
7. Nick Nelson