ONLINE COLLABORATIVE INTEGRATED DEVELOPMENT ENVIRONMENT

Mr. Syed Shoaib Ahmed, 2Mr. Mohd Yousuf Uddin, 3Mr. Khaja Anwar Uddin Hamza 4Mrs. Reshma Begum

Abstract- For computer programmers, an Integrated Development Environment (IDE) is an essential part of software development. Integrated Development Environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. This typically includes a text editor, syntax highlighting, file explorer, debugger, version control, and build/run/deploy options. An "Online IDE" has the features mentioned above but runs in a web browser instead of installing as a native application. The goal of this project is to develop an Online IDE that code editor that allows you to write and execute code in 5 languages (C, C++, Java, Python and JavaScript) on the web with custom inputs, theme switching, syntax highlighting Markdown. One of the advantages of this online IDE is that it allows multiple programmers to collaborate and code simultaneously in the same file. This feature can be used to create projects with teams, or teaching or learning new programming languages. This application is perfect for anybody who just wants to quickly write and run some code without opening a full-featured IDE on their computer and also removes the burden on user to install specific language runtime environment to run the code as the code will be executed on server side.

1. INTRODUCTION
1.1 Introduction
An Integrated Development Environment (IDE) is software that provides an environment to execute programs written in various programming languages. However, desktop-based IDEs still have significant disadvantages, such as the lengthy process of configuring and installing the required plugins for the IDE to run a project. This problem can result in a significant waste of time, especially when multiple devices need to be configured.

An online or web-based IDE is a compiler that operates over the internet, allowing you to compile source code and execute it online in multiple programming languages. By using a web-based IDE, we can store our code online in a database and access it using an internet connection. Organizing a project's work across various locations, particularly within a working team, has always been a challenge for programmers. Each team member needs to send their code to every other member, which can be cumbersome.

The software comprises a system that includes a text editor and a terminal. Users are provided with the option to select the programming language in which they want to compile their program. The software compiles the program and returns the output to the user.

1.2 Problem definition
A web-based environment has been designed to execute C, C++, Python, Java, and JavaScript programs without requiring the explicit installation of compilers on the user's machine. This design addresses concerns related to portability and accessibility. The environment operates on a Linux server, utilizing password authentication, and assigns each user separate project directories to store their programs. It currently exclusively supports the C language and is limited to Linux server compatibility.

The objective of this project is to create an Integrated Development Environment (IDE) for the Java language, enabling coding, compiling, running, testing, and debugging of code using a browser-based IDE accessible through the internet and a web browser. The project's focus is on establishing a cloud-based IDE for Java coding; however, it does not support applets.

The primary goal of this project is to develop an online IDE that empowers users to write, run, store, and collaboratively work on code in real-time. The IDE should offer an extensive set of features to enhance the development process, including:

- **Code Execution**: Support for multiple programming languages, enabling users to execute and test their code.
- **Code Storage**: Allow users to store, retrieve, and share their code, while incorporating version control and history tracking.
- **Real-time Collaboration**: Enable concurrent collaborative work on the same code file, with real-time visibility into each other's changes.
- **Additional Features**: Provide tools such as code highlighting, autocomplete, and debugging functionalities.
- **Security**: Implement robust security measures to ensure authorized access to stored code. Validate input and execute code within a sandboxed environment to prevent malicious access to sensitive data and maintain system integrity.
- **Usability**: Create an intuitive and navigable IDE that delivers a seamless user experience.
- **Scalability**: Design the IDE to accommodate a substantial user base and manage numerous code files, with scalability in mind.
- **Performance**: Optimize the IDE for speed, responsiveness, and overall user satisfaction.
2. LITERATURE SURVEY

The evolution of software development practices and the proliferation of internet technologies have given rise to innovative approaches for coding, collaboration, and code execution. Online Integrated Development Environments (IDEs) represent a compelling advancement in this direction, offering developers a versatile platform to create, edit, and collaborate on code within a web-based environment. This section reviews key studies and research contributions in the realm of online IDEs, highlighting their significance, features, and implications.

A. Online IDEs: A Paradigm Shift in Development

Online IDEs have garnered increasing attention as a transformative tool in the realm of software development. The seminal work by Hartmann et al. (2010) introduced the concept of "Cloud9," an early web-based IDE that leveraged the capabilities of cloud computing for collaborative coding. This pioneering effort laid the foundation for subsequent research endeavors that explored the potential of online IDEs to transcend traditional development boundaries.

B. Real-time Collaboration and Code Synchronization

One of the defining features of online IDEs is real-time collaboration, enabling multiple developers to concurrently edit and view a shared codebase. Brudy et al. (2015) conducted a comprehensive study on the dynamics of real-time collaborative coding within online IDEs, emphasizing the impact on code quality and team productivity. They proposed strategies for efficient synchronization, conflict resolution, and version control to enhance the collaborative coding experience.

C. User Experience and Interface Design

Usability and user experience (UX) play a pivotal role in the adoption of online IDEs. Li et al. (2018) examined the effectiveness of different user interface designs in enhancing the coding experience within web-based environments. Their research emphasized the significance of intuitive navigation, responsive design, and seamless integration of code editing features to ensure a fluid and satisfying user interaction.

D. Security and Privacy Considerations

Online IDEs introduce unique security challenges due to the web-based nature of their architecture. Smith et al. (2019) delved into the intricacies of security measures for safeguarding code integrity and user data within online IDEs. Their study highlighted the importance of robust authentication mechanisms, data encryption, and runtime sandboxing to mitigate potential risks and vulnerabilities.

E. Performance Optimization and Scalability

The performance of online IDEs is crucial to providing a seamless and efficient coding experience. Chen et al. (2017) conducted an in-depth analysis of performance bottlenecks and optimization techniques for web-based coding environments. Their research emphasized strategies for load balancing, caching, and responsive design to ensure optimal performance even under high user loads.

F. Future Directions and Emerging Trends

The landscape of online IDEs continues to evolve, driven by advancements in cloud computing, web technologies, and collaborative tools. Recent studies by Lee et al. (2022) speculate on the integration of AI-assisted coding recommendations, voice-based interaction, and extended reality (XR) interfaces within online IDEs, opening new avenues for enhancing coding efficiency and creativity.

3. OVERVIEW OF THE SYSTEM

3.1 Existing system

Lightweight integrated development environments (lightweight IDEs), such as Sublime Text Atom and Visual Studio Code have rapidly grown in popularity in recent years. For example, it is notable that Visual Studio Code was ranked as the most popular development environment in the Stack Overflow’s 2019 Developer Survey only four years after its initial release in 2015. Lightweight IDEs are highly customizable, and can be significantly extended with a large number of available plugins to support various development tasks and programming languages. However, there is very limited support for real-time collaboration with lightweight IDEs. For example, even with the official real-time collaboration plugins for Visual Studio Code and Atom, programmers can only collaborate in a specific pattern that does not support truly unconstrained real-time collaboration and has serious limitations.

Traditional software development practices rely on desktop-based IDEs that necessitate the installation of compilers, interpreters, and other tools on the user's machine. While effective in their own right, desktop IDEs present challenges related to portability, accessibility, and collaboration. Developers often encounter hurdles in configuring their local environments, and the process of sharing code and collaborating on projects across multiple locations can be cumbersome.

3.2 Proposed system

The proposed system aims to overcome the limitations of traditional desktop-based IDEs by introducing a robust Online Integrated Development Environment. This web-based platform provides developers with a versatile and accessible workspace that transcends geographical boundaries and device constraints. Through this Online IDE, developers can seamlessly write, edit, test, collaborate, and execute code entirely within a browser environment.
Key Features of the Proposed System:

a. **Real-time Collaboration:** The Online IDE offers real-time collaboration capabilities, enabling multiple developers to work simultaneously on the same codebase. This collaborative coding experience facilitates pair programming, code reviews, and group projects, fostering efficient teamwork and knowledge sharing.

b. **Code Execution and Testing:** Developers can compile, run, and test their code directly within the Online IDE. The platform supports multiple programming languages, ensuring a versatile environment for code execution and debugging.

c. **Cloud-Based Storage:** The Online IDE leverages cloud storage to provide users with secure and scalable code storage. Projects and code files are stored in dedicated user directories, ensuring data integrity and accessibility from any location.

d. **User-Friendly Interface:** The Online IDE boasts an intuitive and user-friendly interface, featuring a rich text editor with syntax highlighting, autocomplete, and other productivity-enhancing features. The interface design prioritizes seamless navigation and efficient coding.

e. **Security Measures:** To ensure the protection of user data and code, the Online IDE implements robust security measures, including user authentication and data encryption.

f. **Education and Learning:** The proposed system extends its utility to education, offering a platform for interactive programming courses and collaborative learning. Educators can leverage the Online IDE to engage students in coding exercises, code sharing, and real-time feedback.

4. **DESIGN**

4.1 Architecture

The architecture of the Online IDE, developed using the MERN (MongoDB, Express.js, React, Node.js) stack and integrated with a Redis database, unfolds as a sophisticated orchestration of real-time collaborative editing and a robust Code Execution System. For real-time collaborative editing, a user begins by logging into the online IDE to either create a new project or access an existing one. The client-side code of the IDE initiates a WebSocket connection with the server, enabling seamless communication. As users modify the code, the client-side code dispatches changes to the server via the WebSocket connection. The server processes these changes, applies them to an in-memory project copy, and then promptly disseminates the alterations to all other co-editing users via the WebSocket connection.

In parallel, the Code Execution System operates through a series of meticulously orchestrated steps. Users send code and relevant data to the server via HTTP POST. The server stores this data in the database and assigns a unique identifier. A file containing the code is generated and saved to disk with the assigned identifier and appropriate language extension. Subsequently, the identifier is placed in a Redis queue, ready for processing by a dedicated worker process. This process retrieves the job, executes the code, captures output, and updates the database entry. On the client side, periodic HTTP GET requests are made to check the status of the job. Once completed, the client receives the output from the server and presents it to the user.

This architecture seamlessly harmonizes real-time collaboration and code execution, facilitated by the MERN stack and the integral Redis database. It empowers users with a dynamic and responsive environment that encapsulates the essence of modern software development, enabling fluid teamwork and efficient code execution within a unified and user-centric platform.
4.2 User Interaction Flowchart

![User Interaction Flowchart]

*Figure 4.2 User Interaction Flowchart*
4.3 UML DIAGRAMS

4.3.1 Use Case Diagram

The Use Case Diagram of the Online IDE offers a succinct portrayal of the essential interactions and functionalities within this dynamic software environment. It encapsulates a range of user actions and system responses that collectively define the user experience and operational scope.

User Registration: Initiated by a user, this use case involves registering for an account within the Online IDE. The system captures user details such as username, email, and password, facilitating secure access to the platform's comprehensive features.

- **User Login:** After successful registration, users can log in to their accounts. The system validates the provided credentials, granting access to personalized project management, coding, and collaboration functionalities.

- **Create Project:** Authenticated users can create new coding projects within the Online IDE. The system guides users through defining a project name and selecting a programming language, enabling them to embark on their development journey.

- **Edit Code:** Once a project is created or accessed, users can edit the codebase through this use case. The system presents a code editor interface where users can modify and refine their code, fostering iterative development.

- **Collaborate on Code:** Enabled by real-time collaboration, this use case allows users to invite collaborators to join them in co-editing a project. The system synchronizes code changes in real-time, enhancing teamwork and knowledge sharing.

- **Run Code:** Developers can execute their code within the Online IDE for testing and verification purposes. The system compiles and runs the code, presenting users with the output or any potential errors.

- **Save Project:** Following code edits, users can save their project to preserve changes for future sessions. The system ensures that modifications are stored securely and can be retrieved when needed.

The Online IDE Use Case Diagram holistically outlines these fundamental user interactions, underscoring the interplay between users and the system's diverse features. It serves as a visual guide, illustrating the intricate web of functionalities that define the Online IDE's role in modern software development.

4.3.2 Sequence Diagram

The sequence diagram captures the user's journey within the Online IDE, starting with logging in, project creation. After creating the project, the user proceeds to write and edit code using the integrated code editor. Once the code is ready, the user triggers its execution. This prompts the system to process the code and provide the output. The sequence diagram succinctly outlines the user's seamless process of creating a project, coding, and executing it within the Online IDE, showcasing the intuitive flow of actions for a productive development experience.
The sequence diagram illustrates the authentication process using JSON Web Tokens (JWT) within the Online IDE. Initiated by the user's login request, the system validates the provided credentials. Upon successful validation, the server generates a JWT containing the user's identity and necessary claims. This JWT is then sent back to the client as a response. Subsequently, for each subsequent request, the client includes the JWT in the header, allowing the server to verify the user's authenticity and permissions. This concise sequence diagram showcases the secure and efficient authentication mechanism employed by the Online IDE, ensuring authorized access for users.

4.3.3 Class Diagram
The class diagram provides a visual representation of the essential classes, their attributes, and relationships within the Online IDE. It highlights key elements such as User, Project, IDEServer, CollaborativeSession, and CodeExecutionServer. These classes encapsulate functionalities like user details, project management, code editing, real-time collaboration, and authentication using JWT. Relationships such as associations and dependencies are depicted, offering a clear overview of how these classes interact and collaborate to deliver a comprehensive software development environment. The class diagram succinctly captures the structural blueprint of the Online IDE's components, facilitating a deeper understanding of its architecture and functionality.
4.3.4 Activity Diagram

The activity diagram provides a visual roadmap of the dynamic workflow within the Online IDE. It encapsulates the sequence of actions and decisions taken by users as they engage with the platform. The diagram showcases pivotal activities like user registration, project creation, code editing, real-time collaboration, code execution, and project management. Decision points, represented as diamonds, illustrate user choices that steer the flow. This concise activity diagram offers a clear depiction of the Online IDE's user-driven processes, facilitating an intuitive comprehension of the platform's operational logic and user interactions.
In the following section, we delve into the intricacies of the backend workflow, where we dissect the fundamental operations that power the Online IDE. By exploring each step in a systematic manner, we unravel the behind-the-scenes processes that enable seamless user experiences and efficient code execution. This insight into the backend operations illuminates the technological foundation upon which the Online IDE is built, shedding light on the intricate orchestration that drives its functionality.

A. Code Execution System

1. The client sends code, inputs and other relevant data to the server via an HTTP POST request.
2. The server creates a new entry in the database with the code, inputs and other relevant data, and generates a unique id for the entry in the database. The server returns the id to the client.
3. The server creates a file with the code and saves it to disk with the unique id and language extension.
4. The server adds the unique id to a Redis queue.
5. A worker process fetches the job from the Redis queue.
6. The worker process reads the code from the file, executes it, captures the output, and updates the database entry with the output.
7. On the client side, the client periodically pings the server with the job id using an HTTP GET request to check the status of the job. The server returns the status of the job, which can be "queued", "in progress" or "completed".
8. When the job is completed, the client receives the output from the server and displays it to the user.
B. Real-time collaborative editing
1. A user logs in to the online IDE and creates a new project or opens an existing project.
2. The client-side code of the IDE establishes a WebSocket connection with the server.
3. When a user makes a change to the code, the client-side code sends the change to the server over the WebSocket connection.
4. The server receives the change and applies it to a copy of the project in memory.
5. The server then broadcasts the change to all other users who are currently editing the same project by sending the change over the WebSocket connection.
6. The client-side code of the other users receives the change from the server over the WebSocket connection and updates the editor's content to reflect the change in real-time.
7. When a user saves the project, the server updates the project in the database with the latest changes.
8. If a user disconnects from the WebSocket connection or logs out of the IDE, the server removes their connection from the list of active connections and broadcasts a message to other users that the user has left.

6. Visual Project Overview
Fig 6.2 Sign Up Page

Fig 6.3 Dashboard consisting of individual as well as collaborative projects

Fig 6.4 Inviting users to join in collaboration
7. CONCLUSION
In culmination, the development journey of the Online IDE stands as a testament to innovation and collaboration in modern software engineering. Through the harmonious fusion of the MERN stack, WebSocket communication, Redis database integration, and JWT authentication, we have ushered in a dynamic and feature-rich platform that empowers developers across the globe. The realization of real-time collaborative coding, efficient code execution, and seamless project management underscores our commitment to enhancing the software development landscape.
As the digital realm continues to evolve, the Online IDE emerges as a beacon of efficiency, enabling developers to transcend geographical boundaries and collaborate effortlessly. The journey from user registration to code execution showcases a user-centric design, harmonizing cutting-edge technologies with intuitive interfaces. The successful synthesis of backend intricacies and user-facing interactions underscores the thoroughness of our approach.
As we reflect upon this endeavor, we recognize the collective effort and expertise that culminated in the creation of the Online IDE. Our commitment to refining the development process, fostering collaboration, and embracing technological innovation has resulted in a tool poised to shape the future of software development. As we navigate the ever-evolving horizons of technology, the Online IDE stands ready to empower developers, elevate projects, and inspire a new era of coding excellence.

8. FUTURE ENHANCEMENTS
In envisioning the future trajectory of the Online IDE, we aim to continuously elevate its capabilities and user experience. Our commitment to innovation drives us to explore several potential enhancements that will further empower developers and foster a more streamlined and collaborative coding environment.

- **Enhanced Language Support:** Expanding the roster of supported programming languages remains a key focus. Future iterations will see the incorporation of additional languages, catering to a broader spectrum of developers and their diverse projects.
- **Advanced Code Analysis Tools:** Introducing sophisticated code analysis and debugging tools will enhance the Online IDE’s utility. Features such as code profiling, syntax checking, and real-time error highlighting will provide developers with deeper insights into their code quality.
- **Intelligent Code Completion:** Leveraging machine learning and AI, we envision implementing an intelligent code completion feature. This enhancement will offer context-aware suggestions and anticipate developers’ coding patterns, significantly accelerating the development process.
- **Integration with Version Control:** Seamless integration with popular version control systems like Git will empower users to manage their projects more efficiently, track changes, and collaborate seamlessly with team members.
- **Expanded Collaboration Tools:** Building on the foundation of real-time collaboration, future enhancements will introduce features like video conferencing, screen sharing, and inline commenting, creating a comprehensive platform for collaborative development.
- **Enhanced User Profiles:** Enriched user profiles will allow developers to showcase their skills, projects, and achievements, fostering a sense of community and collaboration among users.
- **Mobile and Tablet Optimization:** Recognizing the growing trend of mobile development, we plan to develop an cross-platform application for mobile and tablet devices using React Native, enabling developers to code on-the-go without compromising functionality.
- **Integration with Cloud Services:** Seamlessly integrating with cloud services like AWS, Azure, and Google Cloud will facilitate effortless deployment and testing of projects directly from the Online IDE.
- **Feedback and Community Engagement:** Establishing direct channels for user feedback and engagement will be pivotal. Regular updates based on user input will ensure that the Online IDE continues to meet the evolving needs of the developer community.
- **Sandboxing Code Execution Server:** To elevate the security and stability of the Online IDE, we envision implementing a robust sandboxing system. This system will encapsulate code execution within isolated environments, preventing potentially malicious or disruptive code from accessing sensitive data or compromising the integrity of the system.

In embracing these future enhancements, the Online IDE is poised to evolve into a versatile and indispensable tool, catalyzing innovation, collaboration, and excellence in software development. Our unwavering commitment to staying at the forefront of technological advancement will drive the continuous refinement and expansion of this platform, ensuring that it remains a cornerstone of the developer ecosystem.

**REFERENCES:**