CodeSync
Team 1 - Product Backlog
Arsh Batth, Jayden Cheung, Arshnoor Randhawa, Adrien Qi

Problem Statement:
In the rapidly evolving field of software development, there exists a significant need for an advanced, real-time collaborative coding platform. Traditional coding environments lack real-time collaboration features, integrated AI assistance, and open-source code, impeding efficiency, modulation, and innovation for coding applications. There exist competitors such as repl.it and VSCode Live Share that provide AI and collaborative features, as well as Kate and ACE code editor that are open-source, but don't provide higher functionality. Our project is the first to combine these features. It will be free and open-source, provide AI augmentation, and support live pair programming, helping users stay connected and write better code.

Background Info:

AI research is advancing rapidly, with new models emerging almost weekly, yet there's a noticeable delay in integrating these innovations into production technologies. At the same time, software development is evolving into a highly collaborative field, as modern software complexity often exceeds the capacity of individual developers. However, traditional Integrated Development Environments (IDEs) have not fully adapted to this shift towards teamwork and shared coding spaces, creating a gap in the market for tools that cater to these evolving needs.

Domain:
The project introduces an innovative online IDE that combines collaborative editing with AI enhancement, addressing the evolving needs of software development. This cloud-based platform facilitates real-time collaborative coding, augmented by AI-driven assistance to boost productivity and enhance code quality. Additionally, it extends beyond coding functionalities, incorporating an online chat feature, and an integrated task and project management system akin to JIRA for streamlined team coordination. Additionally, the platform may feature a reference manual generator and viewer, utilizing NLP models for efficient production and access to pertinent documentation, aligning with the project's focus at the intersection of software development, collaborative tools, and AI-enhanced programming environments.

Targeted Users:

1. Software Developers in Companies: These professionals often work in distributed teams and require tools that support seamless real-time collaboration. The integrated AI enhancement aids in improving productivity and code quality, which is crucial in a professional setting.
2. Students and Educators in Computer Science: This group needs an environment that is not only conducive to learning and teaching coding but also provides exposure to the latest technology and collaborative methodologies.
3. Open-Source Contributors: As open-source projects grow in complexity and contributor numbers, there's an increasing need for tools that facilitate smooth collaboration on common codebases. The proposed platform's focus on real-time collaboration and AI assistance makes it an attractive option for this community.

Similar Platforms:
Platforms like GitHub Codespaces, Gitpod, and Replit offer various levels of real-time collaboration and cloud-based development environments. Other popular platforms like IDE's including IntelliJ and VSCode, as well as open-source options like CodeMirror and Neovim, have additional collaborative coding capabilities. However, most lack a fully integrated AI coding assistant, or they may not be open source and customizable to the extent that a diverse development community requires. Our idea is to combine the best parts of existing platforms while filling critical gaps that affect efficiency and development.

Limitations:

1. **Scalability and Performance:** Existing platforms may not scale efficiently to support real-time collaboration for large teams or complex projects, leading to latency and performance issues.
2. **AI Integration:** While some platforms incorporate AI features, they often lack deep integration with the coding environment or do not offer customizable AI assistance tailored to the project or the user's coding style.
3. **User Experience:** Many collaborative coding environments have steep learning curves or cluttered interfaces, making them less accessible to beginners or those with specific usability needs.
4. **Open Source and Customizability:** Few platforms offer the openness and flexibility required for extensive customization, limiting the ability of users to tailor the environment to their specific needs or contribute improvements to the community.

Requirements:

Functional:

1. As a developer, I would like to register an account so that I can access the platform's features.
2. As a user, I would like to log in to my account so that I can start working on my projects.
3. As a user, I would like to reset my password if I forget it to regain access to my account.
4. As a developer, I would like to create a new project so that I can start coding on a fresh workspace.
5. As a team member, I would like to invite others to collaborate on my project so that we can work together in real time.

6. As a collaborator, I would like to join a project I was invited to so that I can contribute to its development.
7. As a developer, I would like to see changes made by others in real time so that I can collaborate effectively.
8. As a user, I would like to have a basic text editor for coding with syntax highlighting for easier code readability.
9. As a user, I would like to save my project progress so that I can continue working on it later.
10. As a developer, I would like to text chat with my collaborators remotely in real time so that we can discuss the project as we work.
11. As a developer, I would like to voice chat with my collaborators remotely in real time so that we can discuss the project as we work if time persists.
12. As a developer, I would like to video chat with my collaborators remotely in real time so that we can discuss the project as we work if time persists.
13. As a team member, I would like to share files within the platform so that my collaborators and I can easily access them.
14. As a user, I would like to customize the theme of my coding environment so that it fits my personal preference.
15. As a developer, I would like to use basic AI for code completion so that I can code more efficiently.
16. As a user, I would like to receive suggestions for debugging from the AI so that I can fix errors more easily.
17. As a developer, I would like to search within my code so that I can find and navigate to parts of it quickly.
18. As a user, I would like to undo and redo actions so that I can easily correct mistakes.
19. As a user, I would like to view the history of changes made to the project so that I can track progress [Git Integration]
20. As a user, I would like to stage files, enter commit messages, and commit those files in a dedicated view [Git Integration]
21. As a user, I would like to be able to view and compare different branches.
22. As a user, I would like to compare files from previous commits so that I can revert individual files if necessary or revert the entire commit [Git Integration]
23. As a user, I would like to have special IDE support for merge controls which let me view each conflict and accept them so that I can resolve them with ease [Git Integration]
24. As a user, I would like to perform an interactive rebase, which displays a progress bar for how many commits have been applied, and presents each merge conflict so that I can make sure they are resolved and the code has been rebased [Git Integration]
25. As a developer, I would like to run tests on the platform so that I can ensure the code runs as expected.
26. As a team leader, I would like to set permissions for collaborators so that I can control who can edit or view the project.
27. As a user, I would like to receive notifications within the platform (in app) for important updates or messages.
28. As a user, I would like to be able to clear notifications so that I do not have to see many messages
29. As a user, I would like to have notifications be clickable so that I can go and view them separately

30. As a developer, I would like to have a split-screen feature so that I can view multiple files side by side.
31. As a user, I would like to download my project as a .zip so that I can share it or deploy it outside the platform.
32. As a team member, I would like to schedule coding sessions within the platform so that we can plan collaboration times.
33. As a developer, I would like to have a feature to comment on code so that I can leave notes for my team.
34. As a user, I would like to have a quick access toolbar so that I can easily reach frequently used features.
35. As a developer, I would like to customize keyboard shortcuts so that I can work more efficiently.
36. As a user, I would like to have a dashboard to view my active projects so that I can quickly access them.
37. As a team member, I would like to view the profiles of my collaborators so that I can learn more about their skills and roles.
38. As a developer, I would like to have access to a library of code snippets so that I can reuse common patterns easily.
39. As a user, I would like to filter and sort my projects so that I can organize my workspace.
40. As a developer, I would like to compile and run scripts online for supported languages.
41. As a team member, I would like to have a whiteboard feature for brainstorming and planning so that we can visualize ideas together.
42. As a user, I would like to have access to documentation and tutorials so that I can learn more about the platform's features.
43. As a developer, I would like to receive performance insights for my code so that I can optimize it for better efficiency.
44. As a user, I would like to have a feedback option so that I can report issues or suggest improvements.
45. As a team leader, I want to set, view, and adjust deadlines for individual tasks within the platform enabling us to track our progress against the project timeline effectively. This feature allows for the easy identification of upcoming, overdue, and completed tasks based on their deadlines.
46. As a team leader, I want the ability to assign team members to each task ensuring clear responsibility and accountability. This functionality should support multiple team members in a task if necessary and allow reassignment as the project evolves.
47. As a team member, I would want to view a personalized list of tasks assigned to me, complete with their deadlines. This feature should provide an overview of my immediate priorities and help me manage my workload effectively.
48. As a team leader, I need to categorize tasks by priority, type (bug, improvement, feature), and add tags (frontend, backend, etc.). This will facilitate organized task management, allowing the team to filter and prioritize work based on these categories.
49. As a team member, I want to add detailed descriptions to tasks, including the ability to embed code snippets for tasks related to bugs. This feature should support rich text formatting to enable clear communication of task requirements and details.

50. As a team member, I need to view tasks organized by priority in a Kanban-like interface. This visualization should allow for easy identification of high-priority tasks and support drag-and-drop functionality to update task statuses as they move through different stages of completion.

Non-Functional:

Architecture, Performance & Response Time:

1. **Scalable Architecture:** The platform will be built with a microservices architecture to ensure scalability and maintainability. This approach allows individual features or services (e.g., real-time editing, chat, AI suggestions) to scale independently based on demand.
2. **Response Time:** The system will be designed to ensure a maximum response time of 500 milliseconds for all user interactions within the platform, ensuring a seamless experience for end-users during real-time collaboration and AI interactions.
3. **Backend Technologies:** For the backend, we will use Node.js and Socket.IO for real-time communication capabilities. This combination is chosen for its efficiency in handling concurrent connections and its low latency in real-time data transmission.
4. **Frontend Performance:** The frontend will be developed using React to ensure a responsive and dynamic user interface. React's virtual DOM will help in minimizing the performance cost of frequent UI updates during collaboration.

Scalability:

1. **Cloud Infrastructure:** The platform will be hosted on a cloud infrastructure (e.g., AWS, Google Cloud) to leverage auto-scaling capabilities. This allows the system to automatically adjust computing resources based on real-time demand, supporting a large number of concurrent users without degradation in performance.
2. **Load Balancing:** To distribute traffic evenly across servers and prevent any single point of failure, the platform will implement load balancing techniques. This ensures high availability and reliability even under peak loads.
3. **Database Scalability:** The platform will use a combination of SQL and NoSQL databases (e.g., PostgreSQL for structured data and MongoDB for unstructured data), optimized for both performance and scalability. Database sharding or partitioning strategies will be employed to manage large datasets efficiently.

Usability:

1. **Intuitive User Interface:** The design of the user interface will prioritize simplicity and ease of use to ensure that new users can quickly become proficient with the platform without extensive tutorials or training.
2. **Accessibility Compliance:** The platform will adhere to WCAG 2.1 guidelines at the AA level to ensure it is accessible to users with disabilities, including those who use screen readers, keyboard navigation, and other assistive technologies.
3. **Cross-Platform Compatibility:** The platform will be designed to be fully responsive and functional across various devices and screen sizes, including desktops, tablets, and smartphones, as well as compatible with all major browsers.

Security:

1. **Data Encryption:** All data transmissions between the client and server, including real-time collaboration data and personal user information, will be encrypted using TLS/SSL to protect against eavesdropping and man-in-the-middle attacks.
2. **Authentication and Authorization:** The platform will implement JWT (JSON Web Tokens) for secure and efficient user authentication and authorization. This ensures that users can only access and modify their projects and personal information.
3. **Regular Security Audits:** To maintain high-security standards, the platform will undergo regular security audits to identify and mitigate potential vulnerabilities, including SQL injection, cross-site scripting (XSS), and other common web security risks.