

CodeSync - A Web-Based Version Management System for Collaborative Development

¹Aamir Shabir Lohar, ²Monish Parvez Sheikh, ³Rajan Anil Patel, ⁴L.M Kuwar

^{1,2,3,4}Department of Computer Engineering, D. N. Patel College of Engineering, Shahada, Maharashtra, India

Abstract - Code Sync is an advanced web-based version management system designed to help developers and teams collaborate, manage, and track changes in their project files efficiently. Unlike traditional platforms like GitHub, CodeSync focuses on simplicity, real-time collaboration, and cloud-based version tracking tailored for small teams and academic users. The system allows users to: Create repositories and upload files, Track and manage file versions, generate commit messages automatically, Restore or compare previous versions visually. Built with React.js for a modern interface and Node.js + Express.js backend, CodeSync stores all data in MongoDB and is deployed securely on AWS cloud. The project ensures scalable performance and automated testing using Jest framework.

Keywords: CodeSync, Version Control, Collaboration, File Tracking, Repository, React.js, Node.js, Express.js, MongoDB, AWS, Cloud, Jest.

I. INTRODUCTION

Software development projects require continuous updates, teamwork, and proper organization of files. As multiple team members work on the same project, maintaining different versions of files becomes difficult without a structured system. Traditional tools like GitHub provide version control but can be complex for beginners, academic users, and small teams. To solve this problem, CodeSync is developed as a simple and efficient web-based version management system. CodeSync provides an easy platform where users can create repositories, upload project files, and automatically track different versions. It helps users view previous versions, restore older files, and manage project history without confusion. The system also supports smooth collaboration, allowing multiple members to work on the same project without conflicts. With a clean user interface built using React.js and a secure backend using Node.js, Express.js, and MongoDB, CodeSync ensures fast performance and reliable cloud storage. Overall, this project aims to offer an accessible version control solution designed especially for students, developers, and small teams who need a lightweight and effective tool to manage code and project files effortlessly.

1.1 Introduction to project domain

Version control systems are essential tools used in software development to manage changes in source code. They help teams track modifications, maintain multiple versions, collaborate efficiently, and reduce the chances of data loss. Platforms like GitHub and GitLab have transformed the way developers work by offering cloud-based repositories, issue tracking, and collaboration tools. The domain of this project focuses on software development lifecycle, cloud-based version management, and collaborative coding environments.

1.2 Problem Definition

While existing version control platforms are powerful, many beginners struggle with their complexity. They require command-line knowledge, advanced configuration, and sometimes paid features for private repositories. Small teams and students often need a simple and user-friendly solution to store their code, track versions, and collaborate without technical barriers. The problem is to build a lightweight, easy-to-use version control and collaboration platform that simplifies code management for both beginners and developers.

II. LITERATURE SURVEY

According to Beckman et al. (2020), version control systems such as Git and GitHub have become important learning tools in statistics and data science education. The authors explained that integrating version control into academic environments helps students understand collaborative software development, project tracking, and reproducible workflows. Their study highlighted that Git-based collaboration improves teamwork and coding practices among students and developers while simplifying project management activities [1].

Meyer, Chlipala, and Shenker (2014) discussed the importance of file synchronization and version management in cloud storage systems. The authors explained that cloud-based systems require efficient synchronization mechanisms to maintain data consistency among multiple users. Their research emphasized that proper file versioning techniques

help prevent conflicts, ensure reliable backup management, and improve collaboration in distributed computing environments [2].

Mehmood *et al.* (2025) proposed a Large Language Model (LLM) based content classification approach for GitHub repositories using README files. The study demonstrated how artificial intelligence can automatically classify and organize repositories based on project descriptions and textual information. The authors highlighted that AI-based repository classification improves repository discoverability, project organization, and collaborative software management processes [3].

Facebook Inc. (2024) described React.js as a modern frontend JavaScript library designed for building scalable and interactive user interfaces. The documentation explained that React uses a component-based architecture and virtual DOM technology to improve rendering efficiency and application performance. React.js supports reusable UI components and simplifies the development of responsive web applications commonly used in collaborative platforms and enterprise systems [4].

Kalliamvakou *et al.* (2014) examined the opportunities and challenges associated with mining GitHub repositories for software engineering research. The authors identified several limitations, including inactive repositories, incomplete project data, and inconsistent contribution patterns. However, the study also highlighted that GitHub remains one of the most valuable platforms for analyzing collaborative software development practices and open-source ecosystems [5].

Gousios *et al.* (2016) analyzed the pull-based development model widely used in distributed software engineering projects. The research explained that pull requests provide a structured mechanism for reviewing, discussing, and integrating code contributions from multiple developers. The authors concluded that pull-based workflows improve

collaboration, enhance code quality, and support efficient project management in large-scale software systems [6].

Spinellis (2012) discussed the significance of version control systems in modern software engineering. The study explained that version control systems help developers manage source code history, maintain software consistency, and support collaborative development activities. The author emphasized that these systems provide rollback functionality, change tracking, and efficient project maintenance for complex software applications [7].

Zhu *et al.* (2019) explored cloud-based collaboration and version management systems designed for distributed development environments. The authors explained that cloud technologies improve scalability, accessibility, and synchronization among multiple users working on shared projects. Their research highlighted that efficient cloud-based version management systems enhance collaboration performance and support real-time project access [8].

Node.js Foundation (2024) described Node.js and Express.js as important backend technologies for developing scalable and high-performance web applications. The documentation explained that Node.js uses an event-driven and non-blocking architecture suitable for real-time applications, while Express.js simplifies server-side routing and API development. These technologies are widely adopted in collaborative systems requiring efficient request handling and server communication [9].

According to Martin Fowler (2018), enterprise application architecture patterns provide structured solutions for designing scalable and maintainable software systems. The study highlighted the importance of modular architecture, layered design, and reusable components in enterprise applications. Fowler emphasized that proper architectural patterns improve maintainability, scalability, and performance in large collaborative software platforms [10].

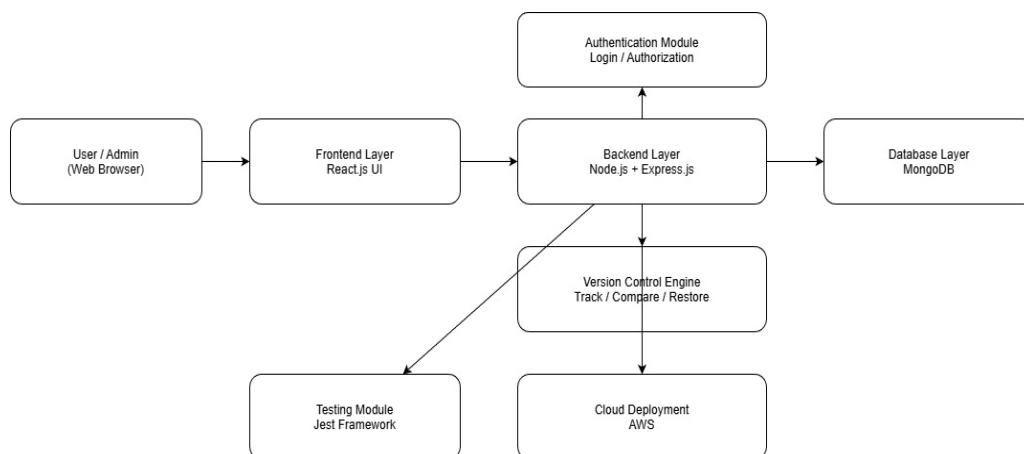


Figure 1: CodeSync Architecture

III. SYSTEM ANALYSIS

The analysis phase focuses on understanding the requirements, workflow, and overall structure of CodeSync. It includes identifying system behavior, understanding user needs, and outlining resources needed for development. This section ensures that the system is designed efficiently to meet functional and non-functional goals.

3.1 System Development Requirements

This section describes the hardware and software resources needed to develop and run CodeSync. These requirements ensure smooth performance, fast processing, and reliable cloud-based version management

3.2 Functional Requirements

- Users can create their repositories and upload files.
- The system will track and manage file versions.
- Commit messages will be automatically generated.
- Users can restore previous versions or visually compare them.
- User authentication will ensure that only authorized users have access.

3.3 Non-Functional Requirements

- Security: Data encryption and login authentication.
- Performance: Fast response and smooth performance for multiple users.
- Reliability: Cloud backup and minimal downtime.
- Scalability: The system can support large teams and repositories in the future.
- Usability: User-friendly interface suitable for beginners and academic users.

3.4 Functional Modeling

- Main system modules: Users, Repositories, and Version Control.
- Users' interactions with repositories and file operations are clearly defined.
- The functional flow shows how file upload, commit, version restore, and comparison work.
- Ensures smooth team collaboration with minimal conflicts.

IV. PROJECT PLANNING

4.1 Software Process Model

The CodeSync project follows the Agile Software Development Model, which is well-suited for projects that require flexibility, frequent updates, and user feedback. In this

model, development is carried out in small, iterative cycles called sprints. Each sprint focuses on completing a small set of functionalities, which are then tested and evaluated. This approach allows developers to identify and fix errors early, incorporate feedback from users or stakeholders promptly, and adapt to changing requirements without affecting the overall project schedule. The Agile model promotes strong collaboration within the team, ensures continuous progress, and results in a higher quality product as the system gradually evolves with each iteration.

4.1.1 Model Description

In the Agile methodology, the project is divided into several short sprints, each lasting for a fixed duration, typically one to two weeks. During each sprint, a subset of features or modules is developed, integrated, and tested. After the sprint ends, the team evaluates the progress and gathers feedback to make improvements in the next iteration. This step-by-step approach ensures that development is incremental, risks are minimized, and any changes in requirements can be easily incorporated. For CodeSync, this model helps in focusing on critical modules like repository management, version control, and real-time collaboration first, while other features are added progressively.

4.1.2 Model Selection Criteria

The Agile model was selected for CodeSync due to several key reasons. First, the project requires frequent updates and modifications based on user feedback, which Agile supports effectively. Second, the development team is small, making iterative development and close collaboration practical. Third, CodeSync is intended for academic and small team users, so requirements may change during the development process, and Agile allows those changes to be implemented without causing major delays. Additionally, Agile ensures that each module is tested thoroughly before moving to the next, reducing bugs and improving overall system quality.

4.2 Estimations

Project estimation is essential to plan the resources, time, and cost required for successful completion. For CodeSync, careful estimation helps in allocating work to team members efficiently, scheduling development activities in a realistic timeframe, and avoiding unnecessary delays. Estimations also provide a reference to monitor progress and ensure that the project stays within the planned scope and budget.

4.2.1 Historical Data

Historical data from previously developed version control and collaboration systems were studied to estimate the development effort for the CodeSync project. Past academic and industry projects similar to GitHub-like platforms were analyzed in terms of development time, software modules, and estimated lines of code (LOC). This analysis helped in understanding the project complexity, identifying major functional components, and making realistic development estimates

V. CONCLUSION

The CodeSync project successfully provides a simple and efficient web-based version management system for small teams and academic users. It allows users to create repositories, upload files, track multiple versions, compare changes, and restore previous versions easily. The system improves collaboration by maintaining proper version history and reducing conflicts during file updates. With a user-friendly interface developed using React.js and a reliable backend using Node.js, Express.js, and MongoDB, CodeSync ensures smooth performance and secure data handling. Overall, the project demonstrates how a lightweight version control system can simplify code management and enhance teamwork compared to traditional manual methods

ACKNOWLEDGEMENT

The Acknowledgement is just like a drop in the ocean of the deep sense of gratitude within our heart for people who helped us out of the most embarrassing part of our life when we were standing at the most difficult step towards our dream of life. Many people have contributed to the success of this. Although a single sentence hardly suffices, we would like to thank Almighty God for blessing us with His grace. We extend our sincere and heartfelt thanks to Prof. V.S. Mahajan, Head of Department, Computer Engineering, for providing us with the right ambience for carrying out this work. We are grateful and sincerely appreciate the efforts of our respected in charges, Prof. V. S. Mahajan and Prof. A. P. Khan, who acted as a fulcrum for us and supported us during the ups and downs of our project. We are profoundly indebted to our project guide, Prof. L.M. Kuwar, for his innumerable acts of timely advice and encouragement, and we sincerely express our gratitude to him. We express our immense pleasure and thankfulness to Principal Prof. N.J. Patil and all the teachers and staff of the Department of Computer Engineering for their cooperation and support.

Mr. Aamir Shabir Lohar #1

Mr. Monish Parvez Sheikh #2

Mr. Rajan Anilbhai Patel #3

REFERENCES

- [1] M. D. Beckman, M. C. etinkaya-Rundel, N. J. Horton, C. W. Rundel, A. J. Sullivan, and M. Tackett, Implementing version control with Git and GitHub as a learning objective in statistics and data science courses, *Journal of Statistics and Data Science Education*, vol. 29, no. S1, pp. S132–S144, 2020.
- [2] D. Meyer, A. Chlipala, and S. Shenker, Managing file versioning and synchronization in cloud storage systems, *Proceedings of the IEEE International Conference on Cloud Computing*, pp. 301–308, 2014.
- [3] M. U. Mehmood, S. Hussain, W. L. Wang, and M. U. Malik, LLM based Content Classification Approach for GitHub Repositories by the README Files, *arXiv:2507.21899*, 2025.
- [4] Facebook Inc. (2024) React.js Official Documentation, available at <https://react.dev/>, accessed in 2025.
- [5] E. Kalliamvakou et al., The promises and perils of mining GitHub, *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR)*, pp. 92–101, 2014.
- [6] G. Gousios et al., Pull-based development model, *Proceedings of the 38th International Conference on Software Engineering (ICSE)*, pp. 345–355, 2016.
- [7] D. Spinellis, Version control systems, *IEEE Software*, vol. 29, no. 4, pp. 66–69, 2012.
- [8] L. Zhu et al., Cloud-based collaboration and version management systems, *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 845–857, 2019.
- [9] Node.js Foundation (2024) Node.js and Express.js Documentation, available at <https://nodejs.org/>, accessed in 2025.
- [10] Martin Fowler (2018) “Patterns of Enterprise Application Architecture,” *IEEE Software*, vol. 35, no. 3, pp. 89–95.
- [11] Ajahar Ismailkha Pathan, Liladhar M. Kuwar, Rijavan A. Shaikh, Dheeraj Basant Shukla, “Removing Duplicate Data in Cloud Environment using Secure Inverted Index Method”, *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056, Volume-05 Issue: 09, Sep 2018.



Citation of this Article:

Aamir Shabir Lohar, Monish Parvez Sheikh, Rajan Anil Patel, & L.M Kuwar. (2026). CodeSync - A Web-Based Version Management System for Collaborative Development. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(5), 390-394. Article DOI <https://doi.org/10.47001/IRJIET/2026.105052>
