WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Universal Ide Using Mern Stack

***Sonali Karape*[1]*, Mr.Pritish Bisne*[2]**

[1]Master of Computer Application,Trinity Academy of Engineering, Pune.
[2]Professor (Dept of MCA),Trinity Academy of Engineering,Pune.

ABSTRACT

In the rapidly evolving landscape of software development, developers require efficient, responsive, and accessible tools to streamline coding, debugging, and collaboration processes. This project presents the design and implementation of a Universal Integrated Development Environment (IDE) built using the MERN stack—MongoDB, Express.js, React, and Node.js. The IDE provides a web-based, cross-platform solution that supports multiple programming languages, real-time code collaboration, syntax highlighting, and execution environments. The frontend, developed with React, offers a responsive and user-friendly interface that supports modular design, live previews, and code auto-completion. The backend, powered by Node.js and Express.js, manages user sessions, project storage, real-time communication (via WebSockets), and language execution using containerized environments. MongoDB is used as a scalable NoSQL database to store user data, project files, and collaboration logs. This Universal IDE aims to simplify development workflows by integrating tools such as terminal access, version control (Git), and cloud-based project storage. Its real-time collaboration feature allows multiple users to code simultaneously, making it suitable for remote teams, educational environments, and pair programming. The project demonstrates the MERN stack's capability to deliver a full-featured, scalable, and accessible IDE, opening doors to future enhancements such as AI-assisted coding and plugin-based extensibility.

INTRODUCTION

In today's fast-paced software development ecosystem, developers seek tools that are not only powerful and versatile but also easily accessible across different platforms. Traditional desktop-based Integrated Development Environments (IDEs), while feature-rich, often face limitations related to platform dependency, installation overhead, and lack of real-time collaboration. To address these challenges, web-based IDEs have gained popularity, offering portability, ease of use, and cloud integration. This project proposes the development of a Universal IDE using the MERN stack (MongoDB, Express.js, React, Node.js)—a powerful combination for building full-stack web applications. The Universal IDE aims to provide developers with a seamless coding experience directly from their web browser, supporting multiple programming languages and tools in a unified interface. By leveraging React, the frontend delivers a responsive and interactive user experience, including features like syntax highlighting, file management, and real-time code editing. The Node.js and Express.js backend handles server-side logic, user authentication, file storage, and code execution, while MongoDB stores user data, project files, and configurations. This web-based IDE is designed not only to support individual development but also to enable real-time collaboration, where multiple users can code together in the same environment. The cloud-native approach ensures that projects can be accessed and edited from any device, anytime, without requiring complex setups or installations. Overall, the Universal IDE using the MERN stack represents a modern, scalable, and collaborative solution tailored to the evolving needs of developers, educators, and teams.

LITERATURE SURVEY/ BACKGROUND

The concept of web-based Integrated Development Environments (IDEs) has evolved significantly in recent years with the growing demand for platform-independent development tools, real-time collaboration, and cloud integration. Various tools and frameworks have been developed to address these requirements. This literature survey explores existing systems, technologies, and research related to web-based IDEs and how the MERN stack can be effectively utilized to build such a solution.Our project bridges this gap by combining YOLOv5s with a Flask web framework to deliver a responsive, intelligent surveillance platform that supports both live feed and uploaded video analysis. This integration addresses both usability and performance, creating a more practical solution for real-world surveillance challenges.

Background and Motivation

Traditional desktop IDEs often require complex installations and lack real-time collaboration features, making them less suitable for modern, cloud-centric development workflows. With the rise of remote work and online education, there is a growing need for accessible, browser-based coding platforms. The MERN stack provides a powerful, full-stack JavaScript solution to build such a scalable and interactive web-based IDE. This project aims to create a universal IDE that simplifies coding, enhances collaboration, and runs seamlessly across devices.

System Architecture

Overview

The Universal IDE using the MERN stack is a web-based development platform designed to support coding in multiple programming languages directly from the browser. Built with MongoDB, Express.js, React, and Node.js, it offers features like real-time collaboration, syntax highlighting, and code execution. The system ensures accessibility, scalability, and ease of use without the need for local setup. It is ideal for developers, students, and teams working remotely.

Technologies Used

- **MongoDB** – Database for storing user and project data
- **Express.js** – Backend framework for APIs
- **React.js** – Frontend for interactive UI
- **Node.js** – Server-side runtime
- **WebSockets** – Real-time collaboration
- **Code Editor (e.g., Monaco)** – In-browser code editing

Features and Functionalities

The Universal IDE offers real-time code editing, multi-language support, and collaborative programming directly in the browser. It includes features like syntax highlighting, file management, code execution, and user authentication.

Implementation

Data Preprocessing

User inputs and code files are sent from the React frontend to the Node.js/Express backend, where they are processed and stored in MongoDB. Real-time updates and collaboration are managed via WebSockets for seamless synchronization.

Model TrainingModel Training

The Universal IDE can integrate with external AI/ML services or APIs to train models on user-provided code or data, with training tasks managed on the backend. Model updates and predictions are communicated back to the frontend for real-time assistance.

Deployment

The MERN-based Universal IDE is deployed on cloud platforms using services like Heroku, AWS, or DigitalOcean for scalable, accessible web hosting.

RESULT

The Universal IDE built using the MERN stack successfully delivers a responsive, web-based development environment accessible from any device with a browser. It supports multiple programming languages, real-time collaboration, and seamless code execution, enhancing productivity for individual developers and teams. The system efficiently manages project data with MongoDB, while React provides a smooth user interface. Real-time features powered by WebSockets ensure synchronized coding sessions. Overall, the project demonstrates that a full-featured, cloud-based IDE can be effectively built using the MERN stack, meeting modern development needs without the overhead of traditional desktop IDEs.

CONCLUSION

collaboration and multi-language support. The integration of MongoDB, Express.js, React, and Node.js provides a scalable and efficient architecture suited for both individual developers and teams. This project highlights the potential of cloud-based IDEs to enhance productivity and foster collaborative coding environments. Future enhancements could include AI-powered code assistance and expanded language support to further improve the user experience. The Universal IDE developed with the MERN stack offers a powerful, flexible, and accessible platform for modern software development. By leveraging web technologies, it eliminates the need for complex local setups while enabling real-time

REFERECNES

1. **MongoDB Documentation**
   MongoDB, Inc. (n.d.). *MongoDB Manual*. Retrieved from https://docs.mongodb.com/
2. **Express.js Official Documentation**
   Express.js. (n.d.). *Express - Node.js web application framework*. Retrieved from https://expressjs.com/
3. **React Official Documentation**
   ReactJS. (n.d.). *React – A JavaScript library for building user interfaces*. Retrieved from https://reactjs.org/

4. **Node.js Documentation**
   Node.js Foundation. (n.d.). *Node.js Documentation*. Retrieved from https://nodejs.org/en/docs/
5. Krill, P. (2020). *Real-time Collaboration in Web Applications*. ACM Computing Surveys, 53(1), 1–28. https://doi.org/10.1145/3379533
6. Rani, P., & Kumar, S. (2020). *Cloud-Based IDEs: Architecture and Implementation*. International Journal of Computer Science and Engineering, 15(2), 100-110.
7. Microsoft. (n.d.). *Monaco Editor – The code editor that powers VS Code*. Retrieved from https://microsoft.github.io/monaco-editor/