



Bconnect using Mern stack

**Prof. Amit Meshram^{*1}, Payal Khawse², Yogesh Kamunkar³, Salif Sheikh⁴,
Bhushan Kotiyan⁵, Swati Kove⁶**

Professor Department of Computer Science and Engineering, Nagarjuna Institute of Engineering Technology And Management, Nagpur, Maharashtra, India¹

UG Student, Department of Computer Science and Engineering, Nagarjuna Institute Of Engineering Technology & Management, Nagpur, Maharashtra, India²⁻⁶

Abstract: This project, Bconnect, is a web-based platform developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js) aimed at facilitating seamless communication and collaboration within an academic or organizational environment. BConnect provides users with features such as real-time messaging, group discussions, announcements, and file sharing, making it an ideal solution for enhancing productivity and connectivity among students, faculty, and staff.

I. INTRODUCTION

Overview

In today's digital world, fast and efficient communication within academic or organizational settings is essential. Traditional tools like emails and notice boards often lead to delays and miscommunication. This motivated the development of BConnect, a centralized platform built using the MERN stack (MongoDB, Express.js, React.js, Node.js).

Purpose

The purpose of BConnect is to provide a real-time, user-friendly web application where users can chat, share files, post announcements, and collaborate in groups. It aims to enhance connectivity and productivity within institutions.

Motivation

BConnect offers a responsive interface, secure login with JWT authentication, and a scalable backend. The platform demonstrates how modern web technologies can solve real-world communication challenges effectively

II. METHODOLOGY

The development of Bconnect followed an Agile-based approach with the following key phases:

1. **Requirement Analysis** – Identified user needs like messaging, group chats, and announcements.
2. **Technology Selection** – Chose **MERN stack** (MongoDB, Express.js, React.js, Node.js) for full-stack development.
3. **System Design** – Created architecture diagrams, database schema, and UI wireframes.
4. **Frontend Development** – Built a responsive interface using React.js.
5. **Backend Development** – Developed RESTful APIs with Express.js and Node.js, and used JWT for authentication.
6. **Integration** – Connected frontend and backend using Axios and ensured real-time functionality.
7. **Testing** – Performed unit and integration testing to ensure reliability.
8. **Deployment** – Deployed the app using platforms like Netlify (frontend) and Render/Heroku (backend).

III. MODELING & ANALYSIS

To design and structure **Bconnect** effectively, the following models and analysis techniques were used:

- **System Architecture:** 3-tier model with React.js (frontend), Node.js/Express (backend), and MongoDB (database).
- **Use Case Diagram:** Showed main actions like login, messaging, group creation, and file sharing.
- **ER Diagram:** Defined relationships between Users, Messages, Groups, and Announcements.



- **DFD (Data Flow Diagram):** Visualized how data moves between frontend, backend, and database.
- **Activity Diagram:** Mapped user flows (e.g., login → chat → logout).
- **Component Diagram:** Outlined key React components and their interactions.

IV. CONCLUSION

The **BConnect** platform successfully utilizes the **MERN stack** to provide a real-time, scalable communication tool for academic or organizational environments. It meets the goals of enhancing communication, offering a secure and user-friendly interface. The project provides valuable experience in full-stack development and can be expanded with features like video calls or mobile integration in the future.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project guide **Prof. Amit Meshram**, for their invaluable guidance and support throughout this project. I also thank the faculty members of Computer Science and Engineering for their encouragement. Special thanks to my friends and family for their constant support and motivation during this journey.

REFERENCES

- [1]. Express.js. (2024). Express - Node.js web application framework. Retrieved April 13, 2025, from <https://expressjs.com/>
- [2]. Meta (Facebook). (2024). React – A JavaScript library for building user interfaces. Retrieved April 13, 2025, from <https://reactjs.org/>
- [3]. Open JS Foundation. (2024). Node.js Documentation. Retrieved April 13, 2025, from <https://nodejs.org/en/docs>
- [4]. Cloudinary Ltd. (2024). Cloudinary Documentation. Retrieved April 13, 2025, from <https://cloudinary.com/documentation>
- [5]. Socket.IO. (2023). Socket.IO Documentation. Retrieved April 13, 2025, from <https://socket.io/docs/v4/>
- [6]. Asur, S. and Huberman, B., 2010. "Predicting the future with social media." In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. IEEE
- [7]. W3Schools. (2024). React and Node.js Tutorials. Retrieved April 13, 2025, from <https://www.w3schools.com/>