

MODHUB - A Project Management Dashboard

Mr. MEGHARAJ U¹, Mr. SENTHIL MURUGAN S²

¹ Student, 4th Semester MCA, Department of MCA, TJIT, Bengaluru

² Assistant Professor, Department of MCA, TJIT, Bengaluru

Abstract—it’s basically a project management dashboard that sorta takes a page outta Jira’s playbook, but with its own twist. The main goal? Make dealing with tasks, teams, and all that project chaos way less painful. You get those drag-and-drop boards, clean tables, and the kind of UI that doesn’t make you want to cry—thanks to a Next.js frontend jazzed up with Tailwind CSS, plus a dash of Material UI’s Data Grid for that spreadsheet vibe. Oh, and Redux Toolkit is wrangling all the app’s state and API stuff under the hood, so it doesn’t busy.

On the backend, it’s all Node.js with Express—nothing groundbreaking, but super reliable and easy to scale. Prisma’s handling the database talk, so you get nice, type-safe communication with PostgreSQL (no more wild west SQL injections, please). And, if you’re the kind of person who likes to poke around the database, PgAdmin’s there for all your snooping needs.

Now, here’s where it gets seriously “enterprise.” The whole thing lives on AWS. The backend chills on EC2, the database is managed by RDS (Postgres, naturally), and the frontend? AWS Amplify’s got it covered. Want serverless magic? API Gateway and Lambda step in for that—meaning you don’t have to babysit servers 24/7. File uploads land safely in S3 buckets, and AWS Cognito keeps your logins locked down, so randoms can’t just wander in.

Bottom line: you get this cloud-powered, secure, and pretty dang scalable platform that actually helps teams work together in real time, without the headaches of old-school tools. It’s like Jira, but maybe a little less soul-sucking.

I. INTRODUCTION

These days, IT teams are at once—think scattered squads, impossible deadlines, and workflows that look like someone dropped spaghetti on the floor. If you’re still managing projects with spreadsheets or random tools, honestly, good luck. That stuff just doesn’t cut it anymore, especially when everyone’s working from different corners of the world and needs things done, like, yesterday.

So, enter this Project Management Dashboard I’ve been putting together. Picture a web app that actually keeps everyone for once. It’s Jira, but without the sticker shock—or the “why-is-this-so-complicated” learning curve. You’ve got project managers, admins, regular team folks—everybody gets a spot in this one-stop shop. No more endless email chains or, “Wait, where’s that file again?” Just real-time updates and smoother teamwork. Hallelujah.

Tech-wise, it’s built with some of the best tools out there right now. Frontend? Next.js, with Tailwind CSS so it doesn’t look like Windows 95, and Redux Toolkit to

keep your data drama-free. Material UI Data Grid makes the tables and lists way less painful. Backend’s rocking Node.js and Express, plus Prisma ORM, which keeps the PostgreSQL database humming along. All the important stuff—tasks, files, users—gets stuffed in Postgres and managed through PgAdmin, which is surprisingly not terrible.

Everything lives on AWS, so you don’t have to worry about your app going down the minute more than five people log in. We’re talking EC2, RDS, Amplify, S3, Lambda, API Gateway, Cognito—the whole alphabet soup. That means you get real-time updates, strong security, and the kind of uptime that won’t make you look bad in front of your boss.

Modernizing routine administrative tasks.

II. RELATED WORK

In today’s fast-paced software development and organizational environments, effective project management is critical for ensuring timely delivery, optimized resource allocation, and team collaboration. Existing project management tools, such as Jira, often

come with **complex configurations, high subscription costs, and steep learning curves**, for startups, educational institutions, and small to medium enterprises.

Additionally, many traditional systems lack **real-time collaboration, seamless cloud integration, and customizable dashboards**, which limits their ability to adapt to the diverse workflows of different organizations. With teams increasingly distributed across geographies, there is a pressing need for a **cloud-native, scalable, and dashboard** that provides secure access, real-time updates, and efficient task tracking.

The problem, therefore, is to design and implement a **cost-effective, customizable, and secure project management application** that replicates core Jira-like functionalities—such as task creation, assignment, tracking, and reporting—while leveraging modern cloud infrastructure and full-stack technologies for performance, scalability, and ease of use.

III. METHODOLOGY

The proposed system is a **cloud-native Project Management Dashboard** inspired by Jira, designed to provide **task management, team collaboration, and workflow automation** in a cost-effective and user-friendly manner.

Frontend: Developed with **Next.js, Tailwind CSS, Redux Toolkit, and Material UI Data Grid** to provide a responsive, dynamic, and intuitive interface for easy task tracking and data visualization.

The system's backend is constructed using Node.js and Express, which offer a scalable and safe method of managing APIs. So, Prisma ORM basically does the heavy lifting when it comes to talking to the PostgreSQL database—no more wrestling with weird SQL errors at 2 a.m. And we're not just tossing this thing on any old server; it's chilling on AWS. Why? 'Cause you want it to scale when your app blows up, and you don't want it crashing every five minutes. We've got EC2 handling the hosting, RDS with PostgreSQL for the database stuff, S3's there for all your files and uploads, Lambda steps in for those "hey, just run this little thing real quick" moments, API Gateway makes sure the APIs aren't just out in the wild west, and Amplify's got your frontend covered. Oh, and security? AWS Cognito's on it, so only the right folks get in.

But wait, there's more—teamwork isn't an afterthought. The app lets you see tasks update in real time, tweak your

dashboard to your heart's content, and actually keep tabs on what's getting done. It's built so startups, smaller businesses, and even schools can actually afford it, instead of coughing up an arm and a leg for Jira. In a nutshell, you're getting a tool that's flexible, not a total resource hog, scales when you need it, and doesn't make your accountant cry. Plus, you still get all the essentials: handing out tasks, watching progress, running reports, and keeping everyone on the same page.

IV. RESULTS AND DISCUSSION

1.Introduction

4.1. Notification and Alert Module
So, here's the deal—AWS Lambda + SNS is your go-to combo for blasting out those "drop everything!" alerts. It's set-and-forget automation for anyone who doesn't want to babysit their notifications all day.

4.2. API and Integration Module
The frontend (Next.js) and backend (Express plus Prisma—pretty standard, right?) talk to each other without drama. You can spin up projects, mess with tasks, or wrangle users, all locked down tight with AWS API Gateway. It's RESTful, it scales, and if you ever want to bolt on Slack, GitHub, or whatever CI/CD flavor-of-the-month, it's ready. Plug and play, baby.

4.3. Cloud Infrastructure and Deployment Module
This bit is all about letting AWS do the heavy lifting—hosting, scaling, the whole nine yards. The frontend? That's chillin' on AWS Amplify, so you can push updates without sweating server stuff.

Backend APIs are hosted on **AWS EC2** and managed through **API Gateway**.

Database (PostgreSQL) is managed through **AWS RDS**.

File storage is handled by **AWS S3**.

Authentication is secured by **AWS Cognito**. This module ensures the application is scalable, secure, and always available with automated backups and disaster recovery.

V. CONCLUSION

The Project Management Dashboard developed in this work provides an efficient, secure, and scalable solution for managing projects, tasks, and teams in a cloud-based environment. By leveraging **modern frontend technologies** such as Next.js, Tailwind CSS, Redux Toolkit, and Material UI Data Grid, the system delivers

a **responsive and interactive user interface** that simplifies project tracking and collaboration. Node.js, Express, and Prisma ORM are used in the system's backend to manage database operations, business logic, and seamless front-end communication. With PgAdmin for administration and PostgreSQL as the primary database, all project data is reliably and systematically stored. The system is set up on to facilitate scalability and security. Because of this, the dashboard can be used by both small and large teams for real-time collaboration.

The dashboard's cost-effectiveness, cloud-nativeness, and customization solve common issues with traditional project management tools. Project managers and team members thanks to modules for authentication, task management, reporting, file sharing, and collaboration.

Testing using unit, integration, system, security, and usability testing techniques to ensure that it satisfies both functional and non-functional requirements. These tests verify that the dashboard is safe, scalable, and user-friendly in a variety of operating environments in addition to functioning as intended.

In conclusion, the Project Management Dashboard serves as a **reliable alternative to commercial tools like Jira**, offering real-time collaboration, improved productivity, and flexibility for modern organizations. The successful implementation demonstrates the potential of combining to build enterprise-level applications. In the future, the system can be further enhanced by incorporating **AI-powered analytics, predictive project risk management, and advanced reporting features**, making it even more powerful for decision-making in project management.

REFERENCES

- [B1] Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional.
- [2] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- [3] Sommerville, I. (2016). *Software Engineering* (10th Edition). Pearson Education Limited.
- [4] Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice*. Addison-Wesley.

- [5] Richardson, C. (2018). *Microservices Patterns: With Examples in Java*. Manning Publications.
- [6] Tanenbaum, A. S., & Van Steen, M. (2017). *Distributed Systems: Principles and Paradigms*. Pearson.
- [7] PostgreSQL Global Development Group. (2023). *PostgreSQL Documentation*. Available at: <https://www.postgresql.org/docs/>
- [8] Prisma ORM. (2023). *Prisma Documentation*. Available at: <https://www.prisma.io/docs>
- [9] Next.js. (2023). *Next.js Documentation*. Available at: <https://nextjs.org/docs>
- [10] Redux Toolkit. (2023). *Redux Toolkit Documentation*. Available at: <https://redux-toolkit.js.org/>
- [11] Amazon Web Services. (2023). *AWS Documentation – EC2, RDS, API Gateway, S3, Cognito, Lambda*. Available at: <https://docs.aws.amazon.com/>
- [12] Tailwind Labs. (2023). *Tailwind CSS Documentation*. Available at: <https://tailwindcss.com/docs>
- [13] Material UI. (2023). *Material UI Data Grid Documentation*. Available at: <https://mui.com/x/react-data-grid/>
- [14] Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th Edition). McGraw-Hill.
- [15] Atlassian. (2023). *Jira Software Documentation*. Available at: <https://support.atlassian.com/jira-software/>
- [16] For Referral of my project deployed use this url address : <https://master.d1eqk0hvwmo0me.amplifyapp.com>
- 1.