

SaaS-Based Sprints Management Platform – Bug Tracking Tool

ARPIT GUPTA¹, VANDANA CHOUDHARY², NAMITA GOYAL³

^{1, 2, 3} Department of Information Technology, Maharaja Agrasen Institute of Technology, GGSIP University, Delhi, India

Abstract- The software industry has been constantly evolving, and the demand for project management systems has been increasing at a fast pace. These systems are essential for managing projects efficiently, tracking progress, and delivering quality products on time. In this research paper, we present a SaaS-based project management system similar to Jira, built using AngularJS, NodeJS, ExpressJS, MongoDB, and AWS. The proposed system is designed with a multitenant architecture to cater to multiple clients, making it highly scalable. The use of NoSQL schema design enhances the scalability of the system. It provides a comprehensive set of features, including project management, sprints management, task and bug tracking, and progress analysis using charts, weekly progress, overall completion, most active users, and sprints analysis. The system's user interface is aesthetically pleasing and user-friendly, enhancing the user experience. Our study focuses on the design, development, and deployment of the system. We also analyze the features and compare them with other project management systems available in the market.

I. INTRODUCTION

The project at hand is a SaaS-based project management system that provides software development teams with a set of tools and techniques to manage their projects, track progress, and deliver quality products on time. The proposed system is designed using AngularJS, NodeJS, ExpressJS, MongoDB, and AWS, making it highly scalable, secure, and efficient. It offers a comprehensive set of features, including project management, sprints management, task and bug tracking, and progress analysis. In the software development industry, project management systems are critical for efficiently managing projects, tracking progress, and delivering quality products on time. There are several project

management systems available in the market, such as Jira, Trello, Asana, and Basecamp. These systems provide software development teams with a set of tools to manage their projects, track progress, and collaborate with team members. However, these systems often lack certain features, such as a multitenant architecture and a NoSQL schema design, which make them less scalable and secure. Additionally, these systems often lack the ability to provide a comprehensive set of progress analysis tools, which are essential for tracking project progress.

The proposed system is designed to cater to the needs of software development teams, providing a unique set of features and tools to manage their projects efficiently. The system's multitenant architecture and NoSQL schema design make it highly scalable and secure, enabling multiple clients to use the system simultaneously. The system's comprehensive set of progress analysis tools, including charts, weekly progress, overall completion, most active users, and sprints analysis, provide software development teams with the ability to track project progress effectively. Moreover, the proposed system's user interface is aesthetically pleasing and user-friendly, enhancing the user experience. In comparison to existing project management systems in the market, our system offers a unique set of features, scalability, security, and a user-friendly interface.

In conclusion, the proposed project management system is a viable alternative to other project management systems available in the market, with its unique features, scalability, and user interface. The proposed system can be used by software development teams, startups, and businesses to efficiently manage their projects, track progress, and deliver quality products on time.

II. LITERATURE REVIEW

In recent years, the demand for project management software has significantly increased as businesses look for ways to streamline their project management processes. Several studies have been conducted on the effectiveness of project management software in improving project success rates and reducing project failures. In this literature review, we will review some of the existing literature on project management software and its effectiveness.

- According to a study conducted by [1] Kim and Kim (2015), project management software can significantly improve project performance and success rates. The study found that project management software enhances project communication, collaboration, and coordination, which are essential for project success. Furthermore, project management software enables project managers to track project progress effectively, identify potential problems, and take corrective actions before they escalate.
- In another study conducted by [2] Crawford and Pollack (2004), project management software was found to improve project planning, communication, and coordination. The study found that project management software improves the accuracy of project planning, enabling project managers to identify potential problems and take corrective actions before they occur. Additionally, project management software enhances project communication and coordination, enabling project teams to work more efficiently and effectively.
- In a study conducted by [3] Kulkarni and Bhattacharjee (2017), project management software was found to enhance project performance by improving team collaboration and coordination. The study found that project management software enables project teams to work more efficiently and effectively by providing them with a set of tools and techniques to manage their projects, track progress, and collaborate with team members.

While project management software has been found to be effective in improving project performance and success rates, some challenges exist with current project management systems. As highlighted in the introduction section, current project management

systems often lack scalability, efficient progress tracking features, user-friendly interfaces, and affordability. The proposed sprint management system in this research paper aims to address some of these challenges, providing businesses with a scalable, efficient, and user-friendly solution to manage their sprints effectively. By incorporating a multitenant architecture, a NoSQL schema design, and a comprehensive set of progress tracking features, the proposed sprint management system can enhance project performance, improve team collaboration and coordination, and ultimately increase project success rates.

III. METHODOLOGY

During the planning phase of the sprints management platform, several key factors were carefully considered. Firstly, it was crucial to define the purpose and objectives of the platform, including task management, sprint tracking, and data analysis capabilities. Identifying the target audience, such as software companies, managers, and engineers, was also an important consideration. Technical requirements played a significant role in the planning process. This involved selecting the appropriate platform and technologies, as well as making decisions regarding hosting and security measures. Emphasis was placed on ensuring a scalable and secure architecture to handle multiple clients efficiently. User experience and design were given careful thought to provide a seamless and intuitive interface. Efforts were made to prioritize user-friendliness, incorporating logical navigation and clear instructions to enhance usability. The features and functionality of the platform were determined based on the identified requirements. This encompassed critical aspects such as multilevel authentication, user role management, and comprehensive sprints and tasks management capabilities. Additionally, the platform aimed to provide detailed data insights to facilitate project analysis and decision-making. To streamline the development process, it was divided into two main parts to ensure efficiency and maintain a clear focus.

A. Design and prototype

UI design encompasses the creation of visual and interactive elements for software applications. Its primary objective is to enhance usability, aesthetics,

and intuitiveness for users. Effective UI design significantly influences the overall user experience and the overall success of an application. In the case of the sprints management platform, high-fidelity wireframes were developed using the Figma application. Presented below are a selection of wireframes that exemplify the design considerations made during the UI design process.

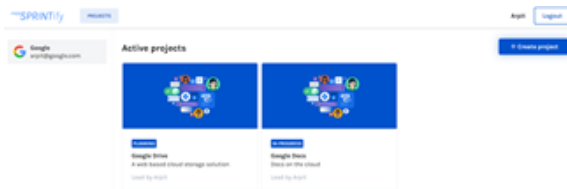


Fig 1. Page displaying active projects in an organisation

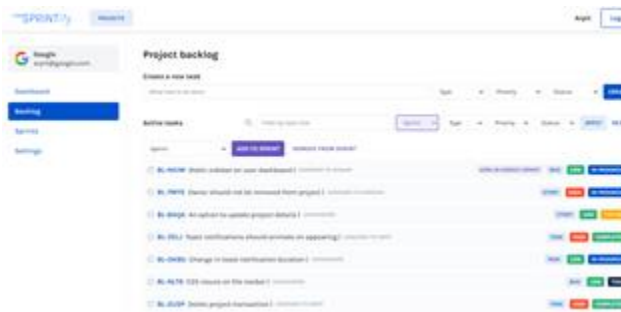


Fig 2. Page displaying active backlog of a project

B. Development

For the development of this hiring platform, I chose MEAN stack (MongoDB, Express.js, AngularJS, Node.js) and AWS. The MERN stack is a collection of technologies used in the development of web applications. It consists of four main components:

1. MongoDB: [4] MongoDB is a versatile document-oriented database system that operates seamlessly across different platforms. It falls under the category of NoSQL databases, meaning it does not adhere to the conventional table-based relational database model. In contrast, MongoDB organizes data in adaptable, JSON-like documents that can have flexible schemas.
2. Express.js: [5] Express.js is a web application framework specifically created for Node.js. Its purpose is to simplify the development and administration of web applications. Express.js

offers a comprehensive set of features and utilities that facilitate the building of server-side applications.

3. AngularJS: [6] AngularJS is a popular JavaScript framework used for building dynamic web applications. It provides a structured and efficient approach to web development by extending HTML with additional functionalities and incorporating the Model-View-Controller (MVC) architectural pattern. AngularJS simplifies the process of creating complex single-page applications by offering features such as data binding, dependency injection, and modularization. With its two-way data binding, AngularJS automatically keeps the user interface in sync with the underlying data, reducing the need for manual updates.
4. Node.js: [7] Node.js is a runtime environment that enables developers to build server-side applications using JavaScript. It leverages the V8 JavaScript engine, the same engine used in Google Chrome, to execute JavaScript code on the server. With Node.js, developers can utilize their existing JavaScript skills to create efficient and scalable server-side applications. By providing a runtime environment for JavaScript outside of the browser, Node.js opens up possibilities for building fast and dynamic web servers, APIs, and other server-side applications.
5. AWS: [8] AWS, short for Amazon Web Services, is a comprehensive cloud computing platform that offers a wide range of services to help individuals and organizations build and deploy various types of applications and infrastructure in a flexible and scalable manner. One of the key services provided by AWS is Amazon S3 (Simple Storage Service). S3 is a highly reliable and secure object storage service that allows users to store and retrieve vast amounts of data over the internet. It provides virtually unlimited storage capacity and is designed to deliver high durability and availability of stored objects.

The platform's development is centred around a client-server architecture, which relies on APIs (Application Programming Interfaces) to retrieve data from the database. The obtained data is then presented on the frontend, resulting in a smooth and cohesive user interface.

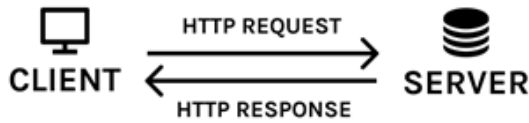


Fig 3. Client server model

[9] Multitenant architecture in SaaS applications refers to a design approach where a single instance of the application serves multiple clients, or tenants. It ensures data isolation, customization options, and efficient resource utilization, offering cost-effectiveness and scalability for SaaS providers.

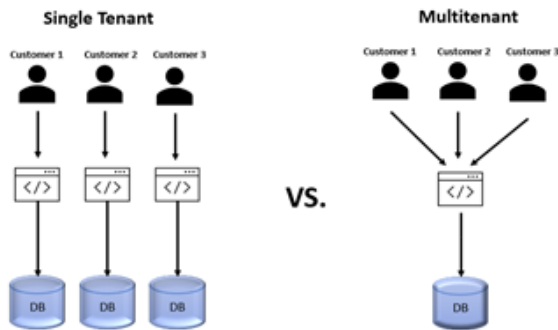


Fig 4. Single tenant vs Multi tenant architecture

To implement the multitenant architecture, the first step involved designing a scalable NoSQL database schema. The schema included a "users" collection responsible for managing user roles and permissions. Each user was categorized as either a superadmin, brand admin, or brand user, with specific functionalities accessible based on their assigned role. This schema design ensured efficient management of user access and permissions within the application. Following is a sample object from the "users" collection.

```
{
  "_id": {
    "$oid": "64228ab3f18515661ac62e3b"
  },
  "name": "Arpit",
  "email": "arpit@google.com",
  "password":
"$2a$10$qG1/sKSn0bjGH3Qgvbu54uiDUYe5Odtu6sE.lOnZWNnJ.NYAzFAU2",
  "roles": [
```

```
"brand_user",
  "manager"
],
  "permissions":
["v_pjt","c_pjt","e_pjt","v_tsk","c_tsk","e_tsk","v_spr","c_spr","e_spr"],
  "isSuperAdmin": false,
  "isBrandAdmin": false,
  "isBrandUser": true,
  "brand": {
    "name": "Google",
    "logoUrl":"https://img.freepik.com/free-icon/search_318-265146.jpg",
    "brandId": "64228a43f18515661ac62e2b"
  },
  "isDeleted": false,
  "didPasswordReset": false,
  "createdAt": {
    "$date": "2023-03-28T06:35:31.396Z"
  },
  "updatedAt": {
    "$date": "2023-03-28T06:35:31.396Z"
  },
  "__v": 0
}
```

Moving on to the backend development, Node.js was utilized to build the server-side of the application. API endpoints were created to handle various functionalities, and robust authorization and authentication mechanisms were implemented using [10] Passport.js JWT (JSON Web Token) local strategy. This ensured secure access to the application's resources. The backend was seamlessly connected to a MongoDB cluster database, allowing data storage and retrieval. By leveraging MongoDB clusters, the application achieved high scalability and availability, accommodating the demands of multiple tenants efficiently. With the backend infrastructure in place, the focus shifted to frontend development using AngularJS. User interfaces were crafted, incorporating responsive and visually appealing designs. The AngularJS framework facilitated the integration of APIs, enabling seamless communication between the frontend and backend components of the application. This integration ensured real-time data updates and interactive user experiences. To provide detailed and in-depth data insights, MongoDB's aggregation framework was employed. Utilizing this feature,

additional APIs were developed on the Node.js backend to calculate and fetch comprehensive information. These APIs enabled the retrieval of data insights such as weekly progress, most active users, backlog status, sprint prediction, and more. By leveraging the aggregation capabilities of the MongoDB database, the application provided valuable analytics to enhance project management and decision-making processes.

Following the completion of development, the entire application code, including both the backend and frontend components, was successfully deployed on web hosting services. Specifically, the backend was deployed on cyclic.sh, while the frontend was deployed on vercel. These hosting services provided the necessary infrastructure and resources to ensure the application's availability and accessibility on the web. By leveraging these hosting services, the developed application was made accessible to users, enabling them to benefit from its features and functionalities.

IV. RESULT

The result of this research paper presents the successful development and deployment of a SaaS-based project management system with a multitenant architecture. Built using AngularJS, NodeJS, ExpressJS, MongoDB, and AWS, the system offers an extensive range of features comparable to popular platforms like Jira. The use of a NoSQL schema design enhances scalability, enabling the system to effectively cater to multiple clients. The system provides comprehensive project management functionalities, including sprints management, task and bug tracking, and progress analysis using various charts and metrics. The aesthetically pleasing and user-friendly user interface enhances the overall user experience. The study encompasses the entire process, covering the design, development, and deployment of the system. Additionally, a comparative analysis of the system's features against other project management systems available in the market is included. Overall, the proposed system demonstrates its capability to address the increasing demand for efficient project management in the evolving software industry.

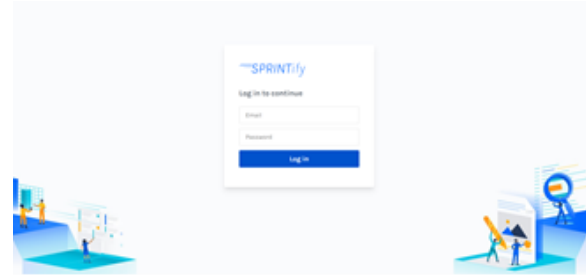


Fig 5. Application login screen

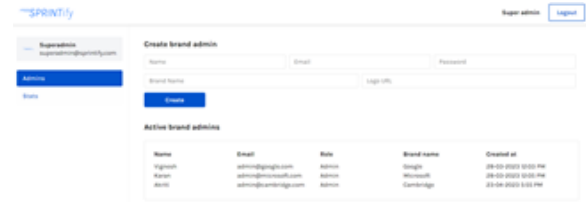


Fig 6. Superadmin dashboard to create brand admins

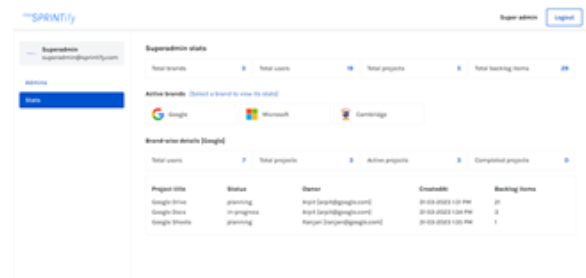


Fig 7. Superadmin dashboard stats to view all clients using this application and track their activities

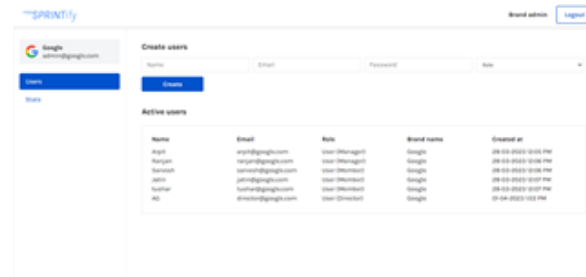


Fig 8. Brand admin dashboard to create brand users

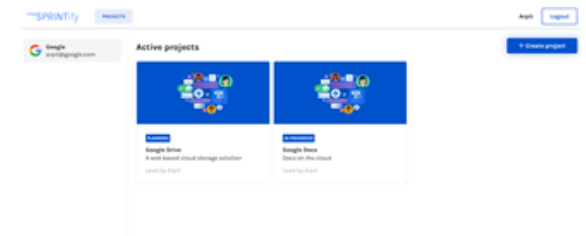


Fig 9. Active projects of an organisation

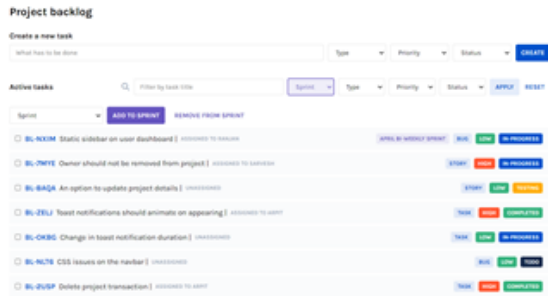


Fig 10. Active backlog of a project

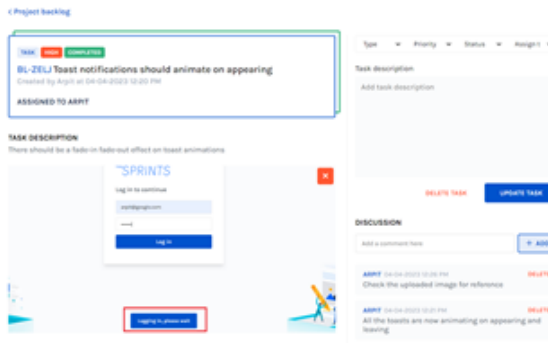


Fig 11. Task details screen

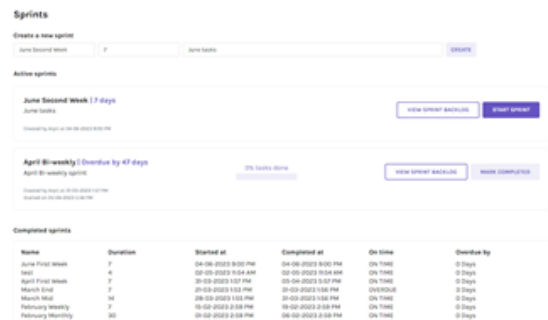


Fig 12. Sprints management screen to view and create active and past sprints

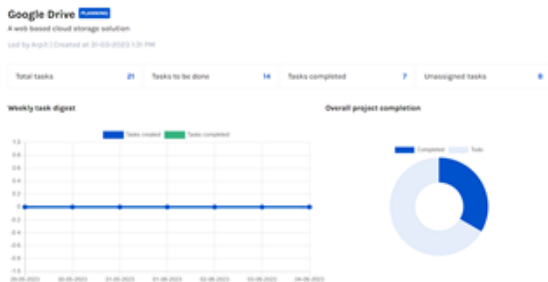


Fig 13. Project dashboard for data insights



Fig 14. Project dashboard for data insights continued

REFERENCES

- [1] Kim, S. H., & Kim, H. J. (2015). The impact of project management software on project performance: An empirical study. *International Journal of Project Management*, 33(3), 655-66.
- [2] Crawford, L., & Pollack, J. (2004). Hard and soft project management: A useful dichotomy? *Proceedings of the PMI Research Conference*, 2004, 1-10.
- [3] Kulkarni, S., & Bhattacharjee, A. (2017). Project management software and project performance: An empirical investigation. *Journal of Organizational Computing and Electronic Commerce*, 27(3), 224-240.
- [4] <https://www.mongodb.com/home>
- [5] <https://expressjs.com/>
- [6] <https://angularjs.org/>
- [7] <https://nodejs.org/en/>
- [8] <https://aws.amazon.com>
- [9] <https://www.gooddata.com/blog/multi-tenant-architecture/>
- [10] <https://www.passportjs.org/packages/passport-jwt/>