**Bigint Computing Journal**
Volume 4, Issue 1, 2026, pp. 40-47
ISSN: 3032-5374
DOI: https://doi.org/10.55537/bigint.v4i1.1370
https://journal.aira.or.id/bigint/index

# Design and Development of a Web-Based Project Management Application Using the Kanban System for Team Collaboration

Perancangan dan Pengembangan Aplikasi Manajemen Proyek Berbasis Web Menggunakan Sistem Kanban untuk Kolaborasi Tim

**Fakhri Alauddin Tarihoran*, Muhammad Irwan Padli Nasution**

Department of Information Systems, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia

**ABSTRACT**

Lack of coordination and transparency often pose significant challenges in managing project teams. This study aims to develop a web-based project management application that utilizes the Kanban method to improve team efficiency and collaboration. The application was developed using Node.js for the backend, React.js for the frontend, Tailwind CSS for the interface design, and MySQL as the database system. The development process followed the Software Development Life Cycle (SDLC) based on the Waterfall model. The results of black-box testing indicate that all core features — including board, list, and card creation, team management, and task assignment — operate successfully. The application significantly improves team coordination, transparency, and overall work performance.

**KEYWORD:** Project Management, Kanban Method, Team Collaboration, Web-Based Application, SDLC

* Corresponding author:
Fakhri Alauddin Tarihoran
Department of Information Systems, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia
Email: fakhrialauddin44@gmail.com
Article History: Received: 2025-10-25; Accepted: 2025-11-30

## 1. INTRODUCTION

The development of information technology and internet networks has brought significant changes to various aspects of life, including the way individuals and organizations work and collaborate. In the digital era, web-based systems enable communication, coordination, and information management processes to be carried out more efficiently without limitations of time and location [1]. The utilization of such technology also provides opportunities for organizations to manage projects in a more structured manner through digital project management systems [2].

In practice, many organizations still face several challenges in project management, such as difficulties in task allocation, monitoring work progress, and coordinating among team members. The use of manual methods, such as spreadsheets or communication through instant messaging applications, often leads to duplicated work, delayed task completion, and low transparency in project progress reporting [3]. Therefore, a project management system that can facilitate effective team collaboration and provide clear workflow visualization is required [4].

One approach widely used in modern project management is the Kanban method. This method visualizes the workflow through boards, lists, and cards, enabling teams to understand task priorities and monitor project progress in real time [5]. In addition, the Kanban method is known for its high flexibility compared to other project management methods such as Scrum, especially in managing information technology projects that require rapid updates and intuitive visualization of progress [6]. The implementation of Kanban has also been proven to improve task management efficiency and support collaboration within digital working environments [7].

Several previous studies have developed web-based project management systems using various modern technologies. Web-based project management systems can increase work transparency and facilitate coordination among team members [8]. Furthermore, the use of modern frontend technologies such as React.js enables the development of responsive and interactive single-page application interfaces [9]. React.js has also been widely

implemented in web-based project management applications due to its capability in efficiently managing interface components [10].

On the other hand, the use of CSS frameworks such as Tailwind CSS provides flexibility and convenience in designing responsive user interfaces [11]. Some studies also indicate that the Kanban method is more effective than Scrum in certain information technology project management scenarios [12]. In addition, the implementation of the Kanban method in web-based systems has been widely adopted to support data management and team collaboration in various information system applications [13].

Other studies also demonstrate that the application of the Kanban method in web-based systems can improve the efficiency of organizational workflows [14]. Moreover, the development of web-based management information systems using modern frameworks can assist organizations in managing project activities in a more structured and systematic manner [15]. Nevertheless, there is still a need to develop project management applications that not only provide task visualization but also offer more integrated collaboration features within a single workspace.

Based on these issues, this study aims to design and develop a web-based project management application using the Kanban method with the Software Development Life Cycle (SDLC) approach based on the Waterfall model. The application is developed using Node.js as the backend runtime environment, React.js as the frontend framework, Tailwind CSS for building a responsive user interface, and MySQL as the primary database system. By integrating member assignment features, task distribution through list structures, and an internal notification mechanism, the proposed system is expected to serve as a practical, efficient, and user-friendly project coordination platform for small to medium-sized teams.

## 2. METHODOLOGY

This study employed a software engineering approach guided by the Software Development Life Cycle (SDLC) methodology. This model was chosen because it provides systematic stages in application development, ranging from requirements analysis to system maintenance. The SDLC approach is used to ensure that each development process is carried out in a structured manner and produces a system that meets user requirements.

### 2.1. System Development Model

This study adopts the Waterfall development model within the SDLC, as each stage follows a logical sequence with strong interconnections.

#### 2.1.1. System Specification Analysis

This stage focuses on identifying the application users' requirements, specifically project teams that need a web-based management system. The primary requirements include the ability to create Kanban boards, add cards and lists, invite users, and assign team members to each task list. Additionally, functional requirements were analyzed, such as user authentication, project data storage, and a responsive user interface.

#### 2.1.2. System Design

This stage involves designing the system architecture, database diagrams, and user interface. The application is built using a client–server architecture, where the frontend employs React.js to provide an interactive and dynamic interface, while the backend is developed with Node.js and Express.js to handle business logic and database connectivity. Tailwind CSS is applied to ensure a consistent and responsive appearance across various devices, with MySQL used as the primary data storage system. System design validation is also conducted to ensure that the design aligns with user requirements and the planned business process flow.

The validation process was conducted through a review of the Use Case Diagram, Activity Diagram, and Entity-Relationship Diagram (ERD). The validation results indicated that the system design meets the functional requirements and is ready to proceed to the implementation stage.

#### 2.1.3. Implementation

After the design phase, the coding process was carried out based on the system design. The frontend components were developed using React.js following a component-based architecture, while the API server was built with Node.js and Express. The implementation process also included the integration between the frontend and backend via RESTful APIs, as well as initial testing of core functionalities such as board creation, card management, and user administration.

#### 2.1.4. System Testing

The system testing process was conducted to verify that all application functions operate as intended. Black-box testing was employed in this process to evaluate outputs based on inputs without examining the internal

program structure. The tested features included login, board creation, card management, user invitations, and team member assignments.

2.1.5. Maintenance

After the system was tested and confirmed to function properly, the maintenance stage was carried out to enhance performance. Maintenance also includes optimizing application performance and ensuring the security of user data.

### 2.2. Tools and Technologies Used

The application was developed using the following tools and technologies:

Table 1. Technology Specifications

| Component | Technology Used | Description |
|---|---|---|
| Programming Language | Javascript (Node.js, React.js) | Main language for frontend and backend development |
| Backend Framework | Express.js | Handles API routes and business logic |
| Frontend Framework | React.js | Builds interactive and dynamic user interfaces |
| CSS Framework | Tailwind CSS | Creates responsive and consistent styling |
| Database | Mysql | Relational database for structured data storage |
| Testing Method | Black-box Testing | Tests system functionality from user perspective |

## 3. RESULTS AND DISCUSSION

The final product of this study is a web-based project management application developed to support efficient team collaboration through the implementation of the Kanban method. The application was designed following a software development approach based on the Software Development Life Cycle (SDLC) using the Waterfall model, ensuring that each development stage is carried out systematically and well-documented. The primary objective of this system development is to create a digital platform that enables teams to plan, monitor, and complete projects in a structured and transparent manner.

During the requirements analysis stage, the system actors and functionalities were identified using a Use Case Diagram. This diagram illustrates the relationships between primary users—such as Project Admin and Team Members—and the main system features, including login, board creation, adding cards and lists, inviting users, and assigning members to tasks. Through this modeling, a clear understanding of system requirements and interactions between components was obtained, facilitating the design and testing in subsequent stages.

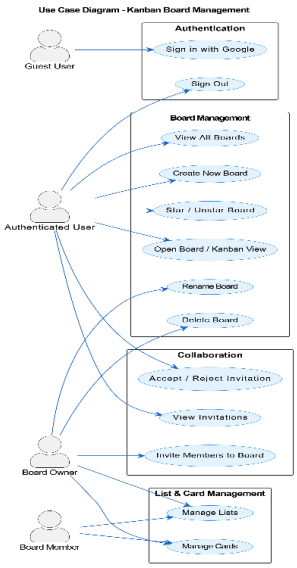


Figure 1. Use Case Diagram

The system design stage was complemented with an Activity Diagram illustrating the user workflow within the application. For instance, the process of creating a Kanban board begins with user login, selecting the project menu, creating a new board, and adding lists and cards according to the work stages. This activity diagram helps ensure that each operational step proceeds logically and efficiently in accordance with the established usage scenarios.

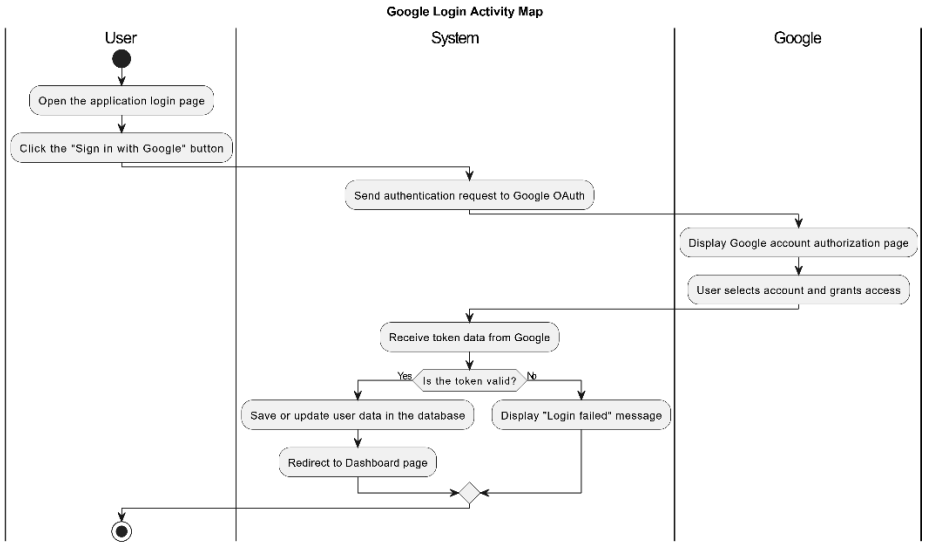


Figure 2. Google Login Activity Diagram

In the Activity Diagram for Google login, the flow begins when the user selects the "Sign in with Google" option on the main page. The system then redirects the user to the Google OAuth authentication service. After the user grants access permission, Google sends a verification token to the server. The server validates the token, then grants access and displays the main dashboard. This process ensures secure and fast user authentication without the need to create a manual account.

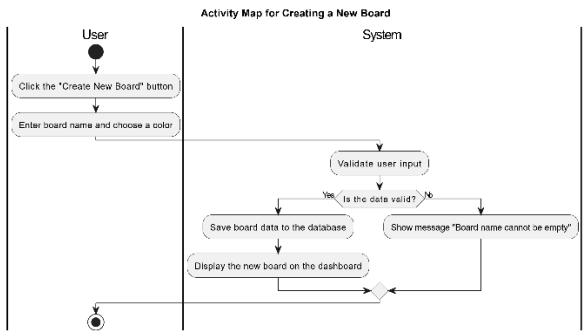


Figure 3. Activity Diagram for Creating a New Board

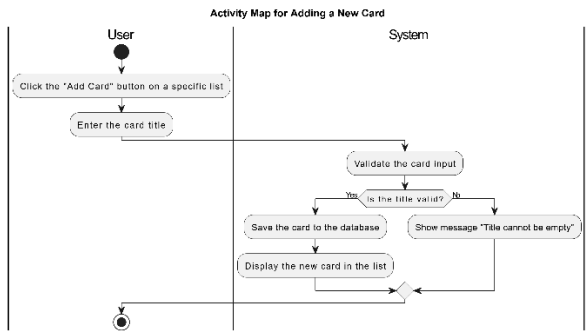


Figure 4. Activity Diagram for Creating a New Card

In the Activity Diagram for creating a new board, the flow begins with a logged-in user selecting the "Create Board" menu. The system displays a form for creating a new project, where the user enters the board name and a brief description. After the data is submitted, the server stores it in the MySQL database and displays the new board on the dashboard page. This activity illustrates the direct interaction between the user and the system in creating a project, which serves as the main container for all collaborative activities.

In the Activity Diagram for creating a new card, the process begins when a user opens a board and selects the list where the task will be created. The user clicks the "Add Card" button and fills in task details such as the name and description. The system stores this data in the database and displays it directly under the corresponding list. This activity allows team members to add and manage tasks according to the Kanban structure being used.

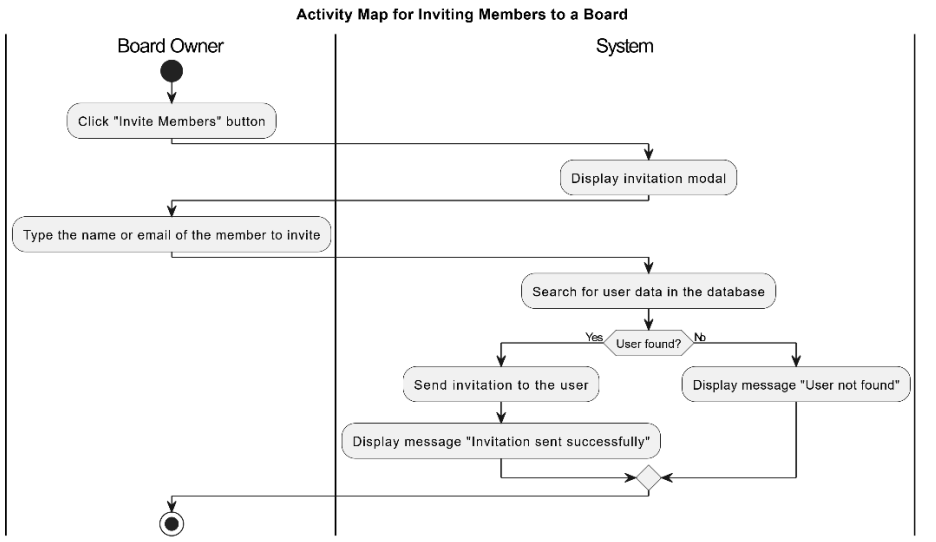


Figure 5. Activity Diagram for Inviting Members to a Project

In the Activity Diagram for inviting members to a project, the flow begins when the project admin opens a board and selects the "Invite Member" option. The user enters the email addresses of the members to be invited, and the system verifies this data in the MySQL database. If the email is found, the system sends an invitation notification to the respective user account. The invitation is considered successful only if the user accepts it through the received notification. Once confirmed, the system adds the user to the project member list and grants access according to their role.

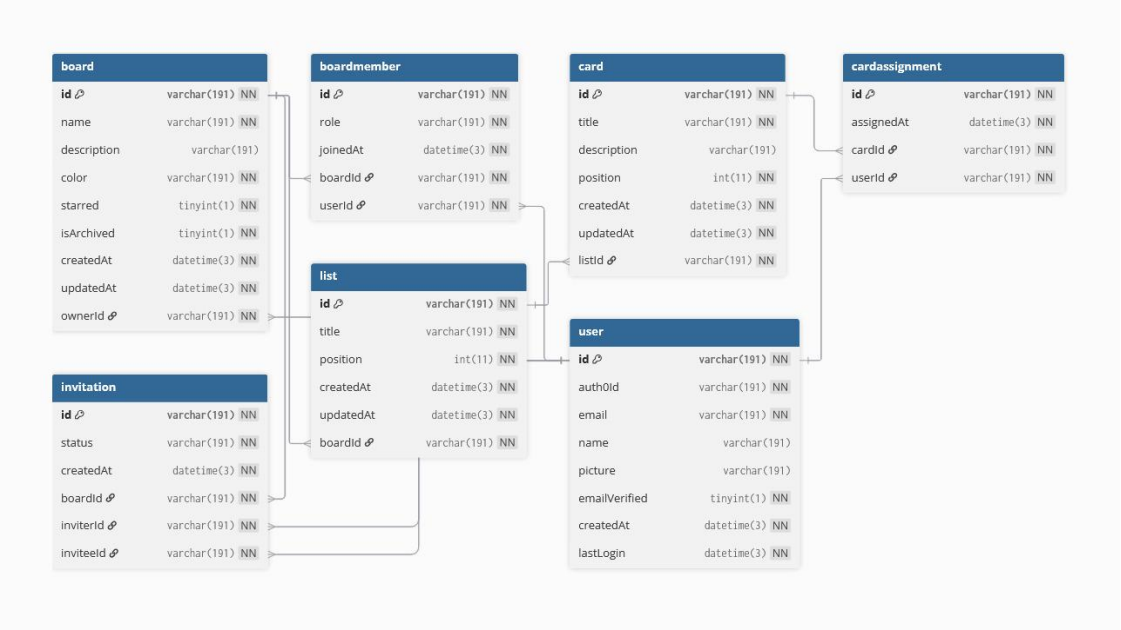


Figure 6. Entity-Relationship Diagram (ERD)

The ERD of this web-based project management application consists of seven main entities: User, Board, BoardMember, List, Card, CardAssignment, and Invitation. This structure illustrates the relationships between data within the Kanban system, where users (User) can create multiple project boards (Board), invite team members through Invitations, and assign them via BoardMember. Each Board contains multiple Lists that hold Cards representing tasks, while CardAssignment records the users responsible for specific tasks. This design supports structured team collaboration and maintains data consistency in project management.

The system implementation was carried out using Node.js as the backend runtime environment, React.js as the frontend framework, and Tailwind CSS as a utility-first CSS framework. The system architecture follows a client–server pattern, with data communication conducted through RESTful APIs. The database structure is managed using MySQL due to its relational nature and stability in handling structured data transactions such as users, projects, task lists, and system activities.

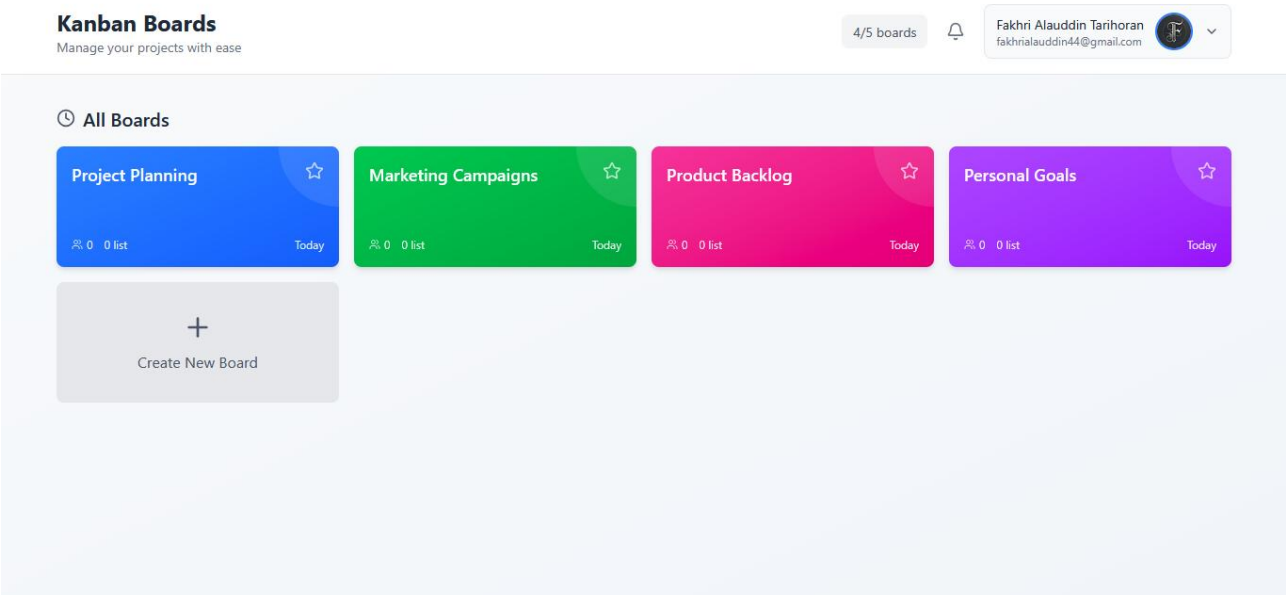


Figure 7. Dashboard Page Interface

On the user interface (UI) side, the dashboard page is designed with a clean and organized layout to facilitate project management. The dashboard displays all boards in a grid of cards, each showing the board name, activity indicators, and timestamps. Users can select an existing board or create a new one via the "Create New Board" button, with intuitive navigation ensuring an efficient user experience.

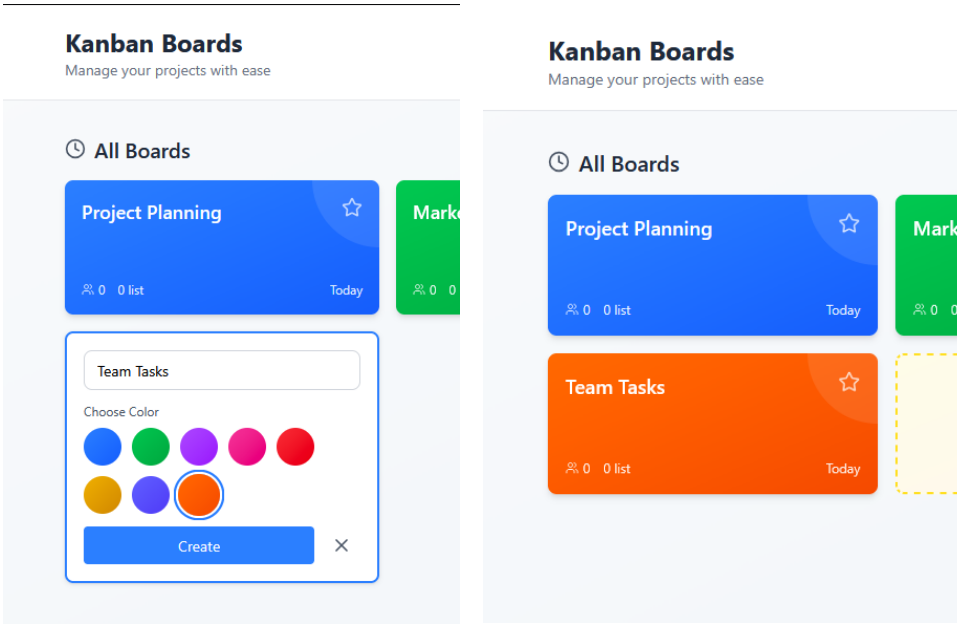


Figure 8. Interface for Adding a New Board

On the user interface (UI) side, the process of creating a new board is designed to be simple and intuitive. Users only need to click the "Create Board" button, which opens a pop-up form to enter the board name. Once the data is submitted, the system immediately displays the new board on the dashboard without requiring a page reload, providing a fast and interactive user experience.

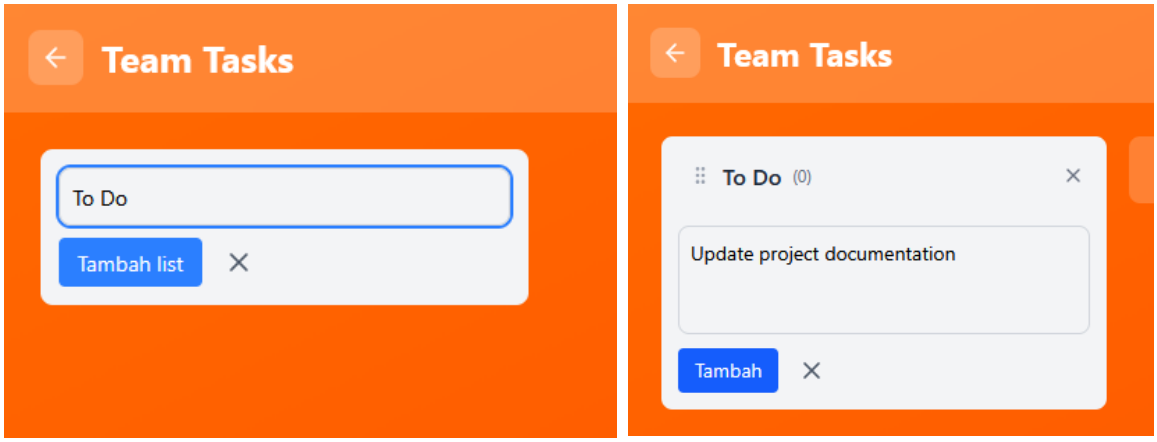


Figure 9. Interface for Adding Lists and Cards

On the user interface (UI) side, the process of creating cards and lists is designed to be easy to understand and quick to use. Users can add lists to represent work stages such as To Do, In Progress, and Done. Within each list, users can create cards containing specific task details. Any changes made, such as adding or moving cards, are updated in real-time without requiring a page reload.

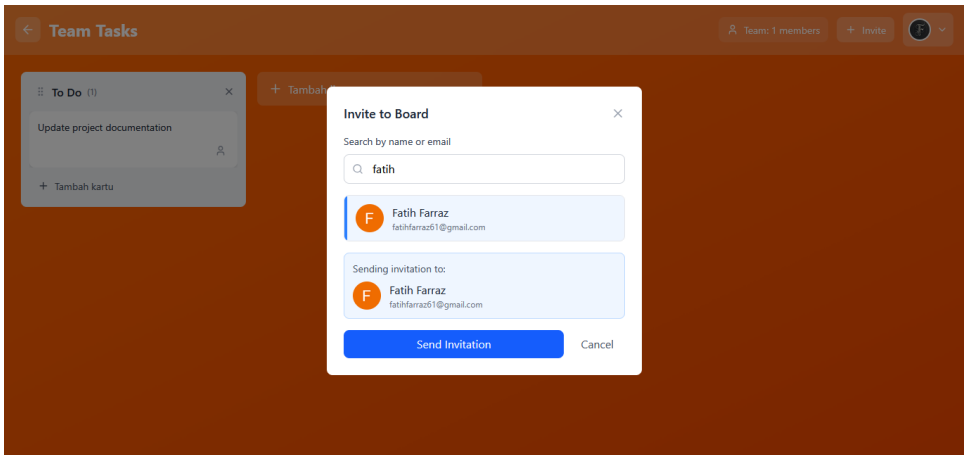


Figure 10. Interface for Adding or Inviting Users to a Project

The system also supports multiuser collaboration through the invite user feature, where the primary user can invite team members to a specific project, assign roles, and ensure that each member has access according to their responsibilities.

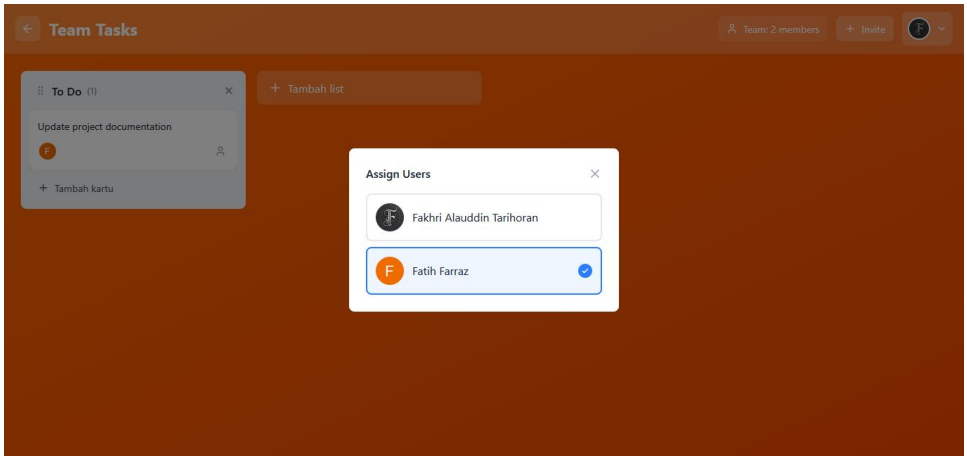


Figure 11. Interface for Assigning Users to Tasks

Furthermore, through the assign user feature, users can assign responsibilities to each card, monitor work progress, and enhance the clarity of the workflow structure as well as the accountability of each team member.

Table 2. Black-Box Testing Results

| No | Feature Tested | Test Description | Input | Expected Result | Result |
|---|---|---|---|---|---|
| 1 | User Authentication | Test login functionality with Google OAuth | Select "Sign in with Google" option | User redirected to dashboard after authentication | Pass |
| 2 | Create New Board | Test creating a new project board | Enter board name | New board appears on dashboard without page refresh | Pass |
| 3 | Create List in Board | Test adding new list to existing board | Enter list name (e.g., "To Do") | New list appears within the board | Pass |
| 4 | Create Card in List | Test adding task card to specific list | Enter card details (title, description) | New card appears in the selected list | Pass |
| 5 | Invite Team Member | Test inviting user to collaborate on board | Enter registered user's email address | System sends invitation notification to registered user | Pass |
| 6 | Assign User to Card | Test assigning team member to specific task | Select user from dropdown in card details | User's name appears as assignee on the card | Pass |
| 7 | Move Card Between Lists | Test drag-and-drop functionality | Drag card from "To Do" list to "In Progress" list | Card moves to new list and Position updates in real-time | Pass |

Based on the results of black-box testing, all features in the application functioned as intended and met user requirements. Features such as login, board creation, card addition, card movement between lists, user invitations, and task assignment operated correctly without causing system errors. The system's performance was also assessed as stable and responsive, with relatively fast page loading times thanks to the optimization of React and Node.js.

Overall, the implementation of the Kanban method and the integration of modern web technologies enhance efficiency and transparency in team project management. UML diagrams assist in designing the system in a structured manner, while the web-based architecture enables cross-device access. Consequently, this application serves as a practical solution for teams that require a simple and functional collaborative system.

The implementation of this application has a positive impact on team collaboration. Workflow visualization through Kanban boards helps members clearly understand task priorities and progress. The invite user and assign task features strengthen coordination, while task status updates enhance work transparency. Overall, the system promotes more efficient, structured, and accountable collaboration in completing team projects.

## 4. CONCLUSION

This study successfully developed a web-based project management application implementing the Kanban method to enhance team efficiency and collaboration. The system was developed using Node.js, React.js, Tailwind CSS, and followed the Waterfall model within the Software Development Life Cycle (SDLC) framework. Testing results indicated that all core features, such as board creation, card and list addition, as well as the invite user and assign user functionalities, operated effectively. The implementation of the Kanban method proved to assist teams in monitoring progress in real-time, improving transparency, and minimizing task duplication.

However, this study has limitations regarding the limited scale of testing and the lack of real-time notification support. Future development can focus on integrating real-time collaboration features and conducting tests on a larger user scale to enhance system performance and user experience.

## REFERENCES

[1] F. A. Widjaja and B. Hakim, "Sistem manajemen proyek berbasis website dengan metode Kanban," *Comput. Sci.*, vol. 5, no. 2, pp. 77–86, 2025. https://doi.org/10.31294/coscience.v5i2.8864

[2] M. A. Ilmi, F. Pradana, and W. H. Nugraha Putra, "Software project management systems using Kanban method in CV Primavisi Globalindo," *INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf.*, vol. 4, no. 2, pp. 215–231, 2020. https://doi.org/10.29407/intensif.v4i2.14320

[3] D. Ramadhani and M. Fakih, "Analisis sistem informasi manajemen proyek menggunakan metode Kanban," *Comput. Sci.*, vol. 1, no. 1, pp. 91–98, 2025.
[4] M. Danaparamita, Y. A. G. Purwoyudo, and D. Darmawan, "Web-based application development for creating collaborative project using Node.js," *Eng. Math. Comput. Sci. J. (EMACS)*, vol. 6, no. 3, pp. 215–224, 2024. https://doi.org/10.21512/emacsjournal.v6i3.11651
[5] A. D. Putri, Q. B. R. D. Seff, and G. R. Suryoadhiva, "Analisis perbandingan efektivitas metode manajemen proyek TI Scrum dan Kanban: A literature review," *J. Ilmu Komput. dan Informatika (JIKI)*, vol. 4, no. 2, pp. 101–110, 2024. https://doi.org/10.54082/jiki.201
[6] L. T. Y. Sheng, S. A/L Subarmaniyan, and N. A. A. Majid, "Web-based Kanban tool for project management," in *Proc. 3rd Int. Conf. Integrated Intell. Comput. Commun. Security (ICIIC)*, 2021, pp. 512–518. https://doi.org/10.2991/ahis.k.210913.056
[7] Z. Ramadhani, E. Emilia, M. Iqbal, D. Ariska, S. Oktarina, and S. Rahayu, "Sistem manajemen proyek pada startup jasa pembuatan aplikasi," *Klik – J. Ilmu Komput.*, vol. 5, no. 2, pp. 101–110, 2024. https://doi.org/10.56869/klik.v5i2.592
[8] S. R. A. Rojak and D. Erica, "Sistem informasi smart project management (SPM) berbasis web pada PT Andromega Data Nusantara Bogor," *Indonesian J. Comput. Sci.*, vol. 4, no. 1, 2025. https://doi.org/10.31294/h9tkbh70
[9] M. Z. Avlach and Y. Yamasari, "Web-based construction project application development with ReactJS, NestJS, and Agile Scrum," *J. Emerg. Inf. Syst. Bus. Intell.*, vol. 6, no. 2, pp. 216–229, 2025. https://doi.org/10.26740/jeisbi.v6i2.70380
[10] R. Jonathan and Suprihadi, "Development of front-end web applications utilizing single-page application framework and React.js library," *Int. J. Softw. Eng. Comput. Sci. (IJSECS)*, vol. 3, no. 3, pp. 529–536, 2023. https://doi.org/10.35870/ijsecs.v3i3.1943
[11] S. Azhariyah and M. Mukhlis, "Framework CSS: Tailwind CSS untuk front-end website store PT XYZ," *J. Informatika*, vol. 3, no. 1, pp. 30–36, 2024. https://doi.org/10.57094/ji.v3i1.1601
[12] A. D. Putri, Q. B. R. D. Seff, and G. R. Suryoadhiva, "Analisis perbandingan efektivitas metode manajemen proyek TI Scrum dan Kanban: A literature review," *J. Ilmu Komput. dan Informatika (JIKI)*, vol. 4, no. 2, pp. 107–116, 2024. https://doi.org/10.54082/jiki.201
[13] H. E. Dien, P. P. Arhandi, A. E. P., and D. S. A., "Sistem informasi multiplatform repositori data dosen berbasis React menggunakan framework Kanban," *J. Inf. Syst. (JOINS)*, vol. 6, no. 1, pp. 116–130, 2021. https://doi.org/10.33633/joins.v6i1.4543
[14] S. Wahyuni, F. Wadly, and R. A. Pratama, "Application of inventory and service transactions on web-based CV Medan Teknik using the Agile Kanban method," *Int. J. Comput. Sci. Math. Eng. (IJECOM)*, vol. 2, no. 1, pp. 8–15, 2023. https://doi.org/10.61306/ijecom.v2i1.16
[15] F. G. P. Susanto, N. I. Y. Fadlan, and P. Haryani, "Web-based management information system design for student organizations in Kendal Regency using Next.js framework," *Compiler*, vol. 2, no. 3, pp. 145–154, 2023. https://doi.org/10.28989/compiler.v12i1.1616