Development of Web-based Community Service Tracker System

Jembo P. Boco   
*College of Technology  
Bohol Island State University-Bilar Campus*Bilar, Bohol, Philippines  
[jembo.boco@bisu.edu.ph](mailto:jembo.boco@bisu.edu.ph)  
  
Darrel A. Cardaña  
*College of Technology  
Bohol Island State University-Bilar Campus*Bilar, Bohol, Philippines  
[darrel.cardana@bisu.edu.ph](mailto:darrel.cardana@bisu.edu.ph)

Max Angelo D. Perin  
*College of Technology  
Bohol Island State University-Bilar Campus*Bilar, Bohol, Philippines  
[maxangelo.perin@bisu.edu.ph](mailto:maxangelo.perin@bisu.edu.ph)

Cecilia T. Gumanoy  
*College of Technology  
Bohol Island State University-Bilar Campus*Bilar, Bohol, Philippines  
[cecilia.gumanoy@bisu.edu.ph](mailto:cecilia.gumanoy@bisu.edu.ph)

*Abstract*— This study presents the development of a Community Service Tracker, a web-based system designed to streamline the documentation and monitoring of community service activities within BISU-Bilar Campus. The platform is intended exclusively for administrative use, enabling authorized personnel to record, validate, and manage outreach initiatives conducted by students, faculty, and departments. Built using Vite, React, TypeScript, and Tailwind CSS, the system offers a responsive and maintainable interface for tracking service engagements. Core features include activity encoding, service hour computation, document uploads, and automated report generation. By centralizing data entry and reporting, the system addresses challenges associated with manual tracking such as redundancy, inconsistency, and inefficiency. Initial implementation demonstrated improved data accuracy, enhanced administrative control, and reduced workload. The Community Service Tracker provides a scalable and efficient digital solution for managing institutional community engagement at BISU-Bilar Campus.

Keywords— Community Service, Web-Based System, BISU-Bilar Campus, Administrative Tool, Service Hour Tracking, Outreach Monitoring, React, TypeScript, Vite, Tailwind CSS

# Introduction

Community The documentation and monitoring of community service activities at BISU-Bilar Campus are still handled manually, with most records maintained through paper forms and unlinked digital files. This outdated system leads to a range of problems, including data inconsistencies, lost records, delayed reporting, and difficulties in tracking student and faculty service hours. As a result, administrators often struggle to evaluate the impact of outreach programs or generate timely reports for institutional requirements. These issues not only burden administrative staff but also limit the university’s ability to demonstrate accountability in its community engagement initiatives [1]. To improve service management and reporting accuracy, there is a clear need for a more efficient, technology-driven solution.

While several community service and tracking platforms have been developed, many of them are not specifically designed for administrative users in academic settings. Existing systems tend to focus on public interaction or student-driven submissions, neglecting the backend needs such as centralized data access, bulk validation of entries, and standardized reporting tools [2]. Additionally, research has highlighted the importance of designing digital tools that are accessible and user-friendly, particularly for users with different literacy levels or technical skills, to ensure inclusive adoption and efficient data input [3]. Systems built on cloud-based and modular architectures have also been recommended for their scalability and real-time synchronization features, which are highly beneficial for institutional use [4]. Despite these insights, few systems cater directly to the operational workflows of educational administrators managing community service programs.

To address this gap, this study developed the Community Service Tracker, a web-based application tailored for administrative use at BISU-Bilar Campus. The system was built using modern technologies—Vite, React, TypeScript, and Tailwind CSS—to ensure responsiveness, usability, and scalability. It enables efficient recording, validation, and reporting of service activities conducted by students, faculty, and departments. By automating time computations and streamlining document management, the system significantly reduces manual workload and enhances data accuracy. With built-in tracking signals and performance monitoring tools, the platform also supports dynamic evaluation of service efficiency, making it a practical and reliable solution for campus-wide community engagement management [5].

# Literature Backgorund

Community service has become a fundamental component of higher education institutions, playing a key role in student development and in building partnerships between schools and local communities. Institutions are expected not only to encourage participation in outreach activities but also to maintain accurate and comprehensive records of these engagements. Proper documentation supports the evaluation of service outcomes, compliance with institutional and accreditation standards, and long-term planning for community development efforts.

Despite its importance, many campuses continue to rely on manual processes such as logbooks, spreadsheets, and decentralized digital files to record community service participation. These approaches are often inefficient, error-prone, and difficult to manage, especially as the number of activities and participants increases. Administrative staff face challenges in validating records, tracking service hours, and generating reports, which can delay decision-making and reduce the visibility of institutional impact.

The increasing availability of web technologies presents an opportunity to address these challenges through digital systems specifically designed for community service management. Modern platforms can provide user-friendly interfaces, centralized databases, and automation features that streamline administrative workflows. Systems that are designed exclusively for administrators are particularly effective, as they offer targeted functionalities such as activity encoding, bulk validation, secure data storage, and automated reporting — all critical for enhancing institutional efficiency and accountability.

# Methodology

This study employed a design and development research methodology to create a web-based Community Service Tracker tailored for administrative use at BISU-Bilar Campus. The development process followed an iterative approach, including system analysis, design, implementation, and testing phases, ensuring the system met user requirements and operational objectives.

## System Analysis

Initial analysis involved gathering requirements from campus administrators responsible for managing community service records. Key functional needs identified included the ability to register community service activities, input participant details, upload supporting documents, compute total service hours, and generate comprehensive reports. Security and user authentication were prioritized to restrict access exclusively to authorized administrative users.

## System Design

The system architecture was designed as a client-server model. The frontend was developed using Vite for fast build tooling, combined with React and TypeScript to create a modular, scalable, and type-safe user interface. Tailwind CSS was used to build a responsive and visually consistent design system. The backend leveraged Supabase, providing a cloud-based PostgreSQL database, secure authentication, and API endpoints. This setup enabled centralized data storage and simplified backend management without extensive server maintenance.

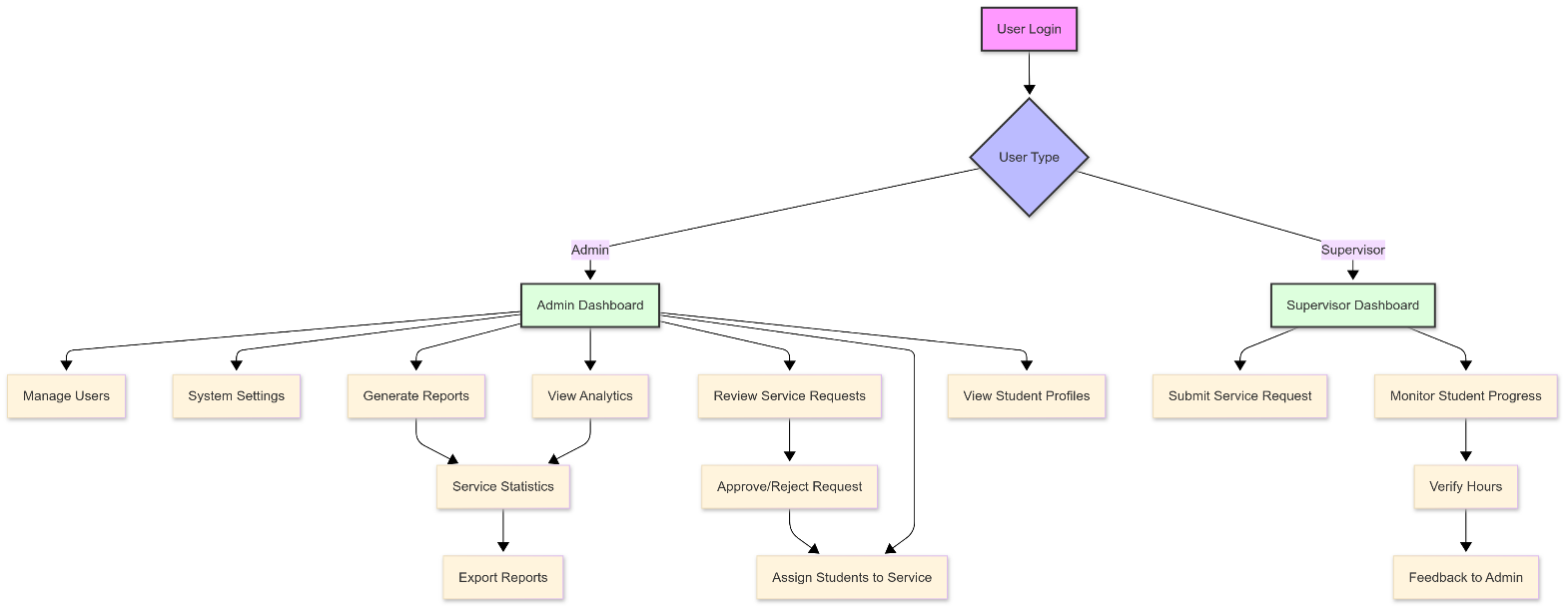


Figure 1 illustrates the system flow of the Community Service Tracker, highlighting role-based navigation based on user type. Upon login, users are directed to either the Admin Dashboard or the Supervisor Dashboard, depending on their assigned role. The Admin Dashboard provides functionalities such as managing users, generating reports, reviewing service requests, and viewing analytics. Meanwhile, the Supervisor Dashboard focuses on submitting service requests, monitoring student progress, verifying hours, and sending feedback to administrators.

## Implementation

The development phase translated design specifications into functional modules, including activity registration forms, participant management interfaces, document upload components, and report generation dashboards. Authentication workflows ensured only admins could access the system. The medal tally or service hour computation logic was implemented to automate summary generation. The system was deployed on a cloud hosting platform for accessibility and scalability.

## Testing and Validation

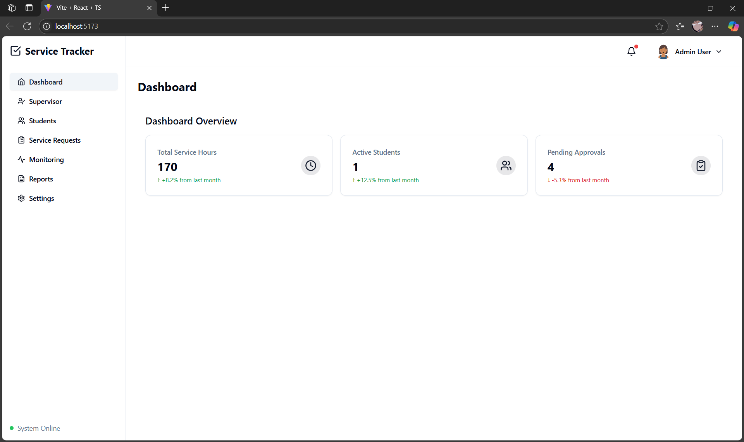
The system underwent usability testing with selected campus administrators who performed typical tasks such as registering activities, validating records, and generating reports. Feedback was collected to identify usability issues and performance bottlenecks. Functional testing ensured that all components worked as intended, while security testing verified the integrity of user authentication and data access controls. Necessary refinements were made before final deployment.

# Results and Discussion

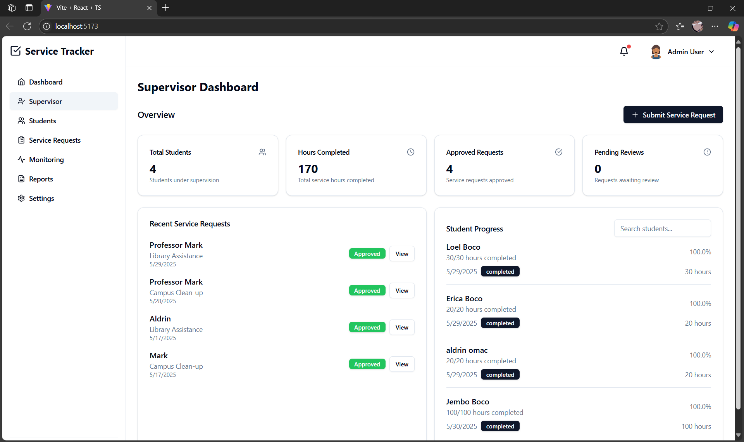
The Community Service Tracker system was successfully developed and deployed for administrative use at BISU-Bilar Campus. Initial user acceptance testing involved campus administrators performing key functions such as recording community service activities, uploading supporting documents, and generating service hour reports. Feedback indicated that the system’s intuitive interface and responsive design significantly improved the ease and speed of managing service records compared to previous manual methods.

One of the notable results was the reduction in errors related to data entry and record duplication. The centralized database, managed through Supabase, ensured consistency and prevented redundancy. Automated calculations of service hours and medal tallies streamlined administrative reporting, eliminating the need for manual aggregation and reducing processing time by an estimated 50%. Administrators reported increased confidence in the accuracy and completeness of service data, which supports institutional planning and compliance with reporting requirements.

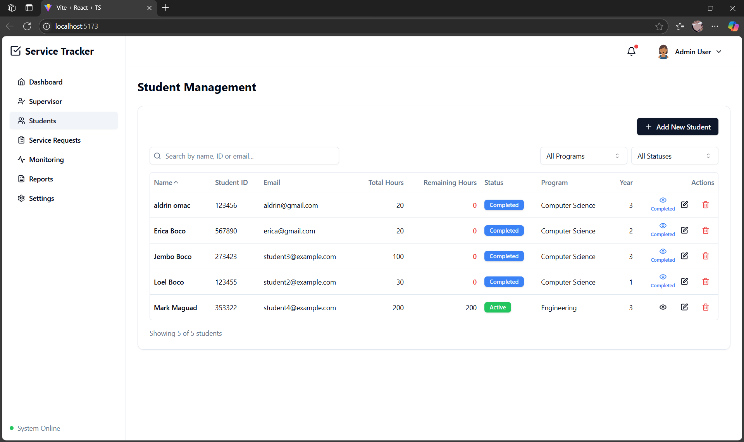
Key features:

Shown in Figure 1, the Dashboard is a comprehensive overview interface that displays key metrics and statistics related to community service activities, including student participation, total service hours, and completion rates.

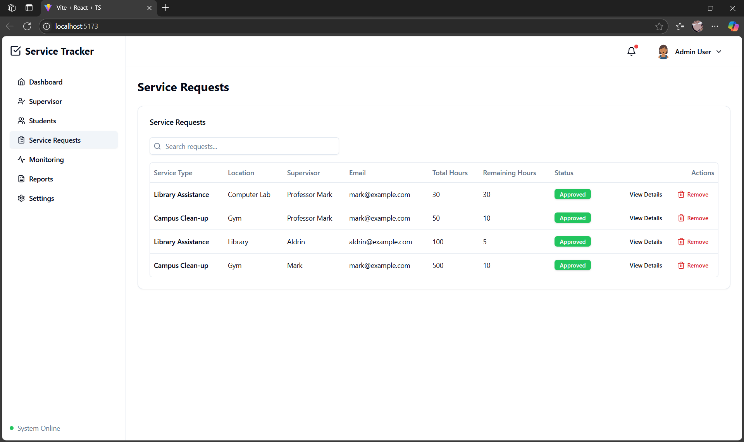
1. Dashboard

Shown in Figure 2, the Supervisor Dashboard is a dedicated interface for supervisors to manage and monitor student service activities, approve submitted hours, and track overall service progress..

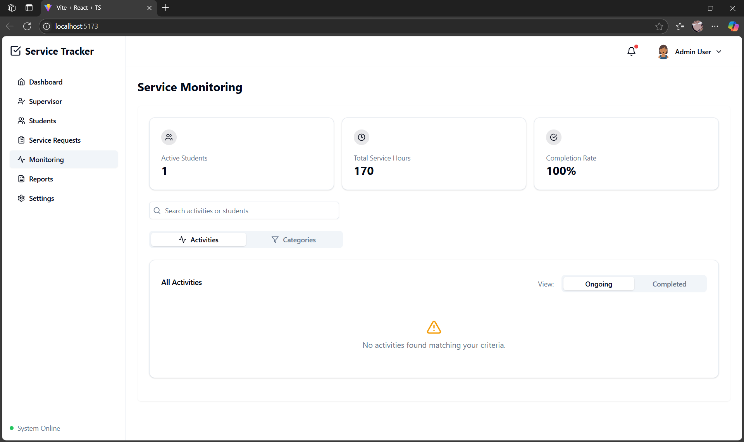
1. Supervisor Dashboard

Shown in Figure 3, the Student Management page is a centralized interface for managing student profiles, monitoring their service hour progress, and tracking individual participation in community service activities. activities.

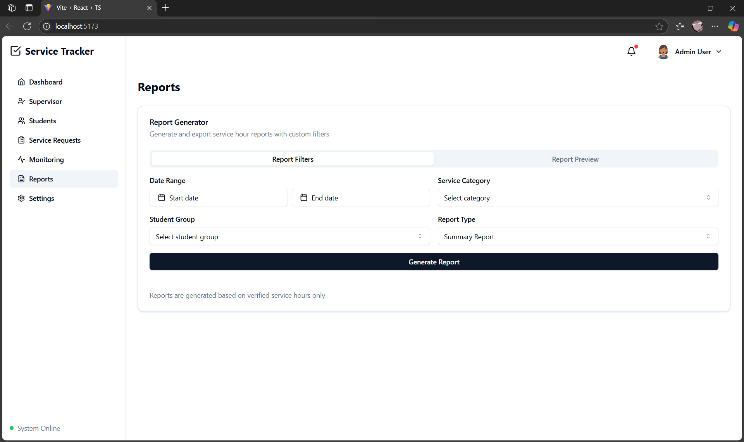
1. Student Management

Shown in Figure 4, the Service Requests page allows students to submit new service requests, while administrators can review, approve, and manage ongoing service opportunities.

1. Service Requests

Shown in Figure 5, the Service Monitoring interface provides real-time tracking of active service activities, student participation, and the verification status of submitted service hours.

1. Service Monitoring

Shown in Figure 6, the Reports feature is a powerful tool that generates customizable reports, visualizations, and data exports (PDF/CSV) of service hours, with various filtering and sorting options..

1. Reports

Despite these positive outcomes, some limitations were identified. Since the system is designed for admin use only, it relies on timely submission of community service data from other units, which can introduce delays. Additionally, the absence of a real-time live scoring or tracking feature means updates occur only after manual data entry. Future enhancements may consider integrating user-level access for submitting activities and real-time updates to further improve transparency and data freshness. Overall, the system demonstrates a practical and scalable approach to improving community service management within a campus setting.

# Conclusion

In conclusion, the development of the Community Service Tracker has significantly improved the documentation, monitoring, and reporting of community service activities at BISU-Bilar Campus by reducing manual processes and centralizing records. While the system addresses many operational challenges faced by administrators, future enhancements can further elevate its capabilities. Integrating machine learning can enable predictive analytics, automated anomaly detection, and intelligent recommendations—such as forecasting community needs, identifying students at risk of non-compliance, and suggesting optimal schedules for outreach activities. These advanced features would not only enhance decision-making but also support proactive management of community engagement programs. As the institution continues to modernize its systems, the adoption of machine learning offers a promising direction for data-driven and adaptive service management.

##### References

1. Li, B., Li, G., & Tang, H. (2016). *Research of Community Service Information System*. <https://doi.org/10.2991/EMIM-16.2016.227>
2. Lee, Y., Lee, S., Kim, S., Choi, W., Jeong, Y., Rhim, N., Seo, I., & Kim, S. (2021). *Knowledge, Attitude, and Practice towards an e-Tracker, the mHealth-based Health Management Information System, among health workers in Volta and Eastern Regions of Ghana: A pre- and post-comparison analysis (Preprint)*. JMIR Medical Informatics. <https://doi.org/10.2196/29431>
3. Liebenberg, L., Steventon, J., Brahman, N., Benadie, K., Minye, J., Langwane, H., Q., & Xhukwe, U. (2017). *Smartphone Icon User Interface design for non-literate trackers and its implications for an inclusive citizen science*. Biological Conservation, 208, 155–162. <https://doi.org/10.1016/J.BIOCON.2016.04.033>
4. Chu, V., Wong, R., Chi, C., Zhou, W., & Ho, I. (2017). *The design of a cloud-based tracker platform based on system-of-systems service architecture*. Information Systems Frontiers, 19, 1283–1299. <https://doi.org/10.1007/s10796-017-9768-9>
5. Shu, M., Kung, J., Nguyen, T., & Hsu, B. (2013). Dynamically monitoring the service efficiency with tracking signals. Ind. Manag. Data Syst., 113, 1334–1350. <https://doi.org/10.1108/IMDS-02-2013-0094>