International Journal Software Engineering and Computer Science (IJSECS)
6 (1), 2026, 378-390
Published Online April 2026 in IJSECS (http://www.journal.lembagakita.org/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i1.7132.

RESEARCH ARTICLE Open Access

Comparative Performance Analysis of Integrated Monitoring Engine for Electric Energy Transaction Data Gateway Infrastructure to Accelerate SLA Incident Resolution

Yosua Christian Prasetio [1*], **Yeremia Alfa Susetyo** [2]

[1*,2] Universitas Kristen Satya Wacana, Salatiga City, Central Java Province, Indonesia.

**Corresponding author*: 672022102@student.uksw.edu.

Received: March 27, 2026; Accepted: April 25, 2026; Published: April 30, 2026.

Abstract: The increasing demand for mental health services among young people in Indonesia is not matched by adequate accessibility due to constraints related to time, cost, and social stigma. This study aims to implement a "Mental Health Care" website as an online counseling and psychological education platform by integrating the Spring Boot framework on the backend and Tailwind CSS on the frontend. The development method used is the Waterfall model, which includes the stages of problem identification, system analysis and design, implementation, and testing. The system is designed using a three-layer architecture consisting of a frontend layer based on Vanilla JavaScript and Tailwind CSS, a backend layer based on Spring Boot with Spring Security for authentication and authorization, and a database layer using MongoDB. The main features implemented include real-time online counseling using WebSocket (SockJS & StompJS), an artificial intelligence assistant powered by Gemini AI through Spring AI, MBTI psychological tests, access to psychological educational e-books, a per-session payment system, and authentication via Google OAuth2. Testing is conducted using the Black-box Testing method and interface responsiveness testing to validate system functionality and compatibility across various devices. The results of this study indicate that all 12 functional test scenarios passed with no failures detected, and interface responsiveness testing confirmed full compatibility across Desktop (1920×1080), Tablet (768×1024), and Mobile (375×667) devices. These findings demonstrate that the integration of Spring Boot and Tailwind CSS can produce a functional, secure, responsive, and accessible platform, making it a potential technological solution to improve the accessibility of mental health services and psychological literacy in the digital era.

Keywords: Mental Health; Online Counseling; Spring Boot; Tailwind CSS; WebSocket; MongoDB; Artificial Intelligence; MBTI.

1. Introduction

The current phenomenon in Indonesia reveals a stark contradiction within the national healthcare landscape. On one hand, the demand for mental health services among the younger generation continues to escalate significantly. On the other hand, access to these vital services remains profoundly limited. Nelyahardi *et al*. (2023) discovered that young individuals tend to welcome the use of digital media — such as Instagram, WhatsApp, and websites — as counseling tools, because they are perceived as more flexible and accessible compared to traditional face-to-face consultation. This shift underscores a substantial potential for developing web-based services to reach those who face geographical or social barriers in obtaining direct assistance. The trend is further supported by a systematic literature review indicating that digital counseling is increasingly favored due to its "anytime, anywhere" convenience, although it still faces challenges when dealing with severe psychological disorders (Prayoga *et al*., 2025). Furthermore, the implementation of web-based cyber-counseling has been proven to increase students' interest in seeking help, which has long been hindered by pervasive social stigma (Prasetya *et al*., 2020). Consequently, websites now play a role not merely as information portals but as inclusive platforms for psychological education and intervention.

Despite this technological potential, the digital mental health landscape is still overshadowed by critical issues that necessitate urgent research. One primary obstacle is the low professional help-seeking intention among students, even when they possess foundational knowledge of mental health issues (Azedarach & Ariana, 2022). Research conducted at Airlangga University warned that low mental health literacy carries the risk of leaving psychological conditions undiagnosed; if left unaddressed, such conditions can trigger prolonged stress or even suicidal behavior (Azedarach & Ariana, 2022). From a technical perspective, many existing platforms are suboptimal due to a lack of system integration, overly complex user interfaces, and accessibility barriers that hinder service effectiveness (Dewi *et al*., 2022). A study by the Bali Institute of Technology and Health illustrated this clearly — despite high depression literacy, students' attitudes toward seeking help remained poor because available digital platforms failed to provide structured intervention features (Dewi *et al*., 2022). Additionally, a digital literacy gap exists across different student backgrounds, where non-nursing students tend to have lower digital literacy, making it difficult for them to navigate and utilize web-based services effectively (Andalasari & Safitri, 2024).

The limitations identified in previous studies highlight a significant research gap that this study aims to fill. While recent studies have demonstrated that website-based approaches can overcome physical barriers in mental health service delivery, many of these systems still relied on conventional architectures that struggle with modern scalability and data security requirements. Akbar *et al*. (2024) developed a web-based e-counseling program with educational features but did not integrate real-time communication capabilities, and Hafidz *et al*. (2024) built a progressive web application for mental health education without incorporating interactive counseling functionality. To address the issues of responsiveness and data management frequently reported in earlier systems, this study proposes a solution through the utilization of the Spring Boot framework for the backend and Tailwind CSS for the frontend. Spring Boot provides a high level of data security and strong authentication mechanisms, which are critical for handling sensitive psychological data (Abdurrahman, 2025). This aligns with the necessity of robust data encryption and privacy protection in transforming online counseling services into professional and secure platforms (Rahmadhea, 2024). Meanwhile, Tailwind CSS offers a utility-first approach that produces highly responsive interfaces, outperforming traditional frameworks like Bootstrap in terms of customization and performance (Santoso, 2025). By combining these technologies, technical obstacles such as confusing navigation or ambiguous scheduling can be minimized, leading to more efficient and reliable service delivery (Hanapi & Harjono, 2025).

The novelty of this research lies in its integrated approach to building a "Mental Health Care" platform that balances technical rigor with user-centric design. Unlike previous models that focused solely on information dissemination (Hafidz *et al*., 2024), this system was built for structured psychological intervention within a secure, high-performance environment. The choice of Spring Boot ensures that the application is scalable and maintainable, while Tailwind CSS ensures that the platform is accessible across various devices — essential given the varying digital literacy levels among students (Pamungkas *et al*., 2025). Maharani (2025) demonstrated that aligning a modern frontend framework with a structured backend improves service accessibility, while Ulum (2022) showed that the absence of such integration leads to manual data handling and reduced reliability in counseling systems. This research builds on both findings by examining how specific backend-frontend pairings can directly affect the accessibility and reliability of mental health services in a digital-first society. Based on the problems and gaps identified above, the primary research objectives of this study are as follows:

1) To design and develop a backend system using the Spring Boot framework that ensures secure data handling and user authentication for mental health services.
2) To implement a responsive and intuitive user interface using Tailwind CSS to enhance the accessibility and user experience of the platform.

3) To evaluate the functional integrity of the developed system through Black-box testing to ensure that all features meet the specified requirements.
4) To analyze the effectiveness of the platform as an alternative mental health service aimed at reducing social stigma and increasing mental health literacy.

This research provides both practical and theoretical contributions. Practically, the "Mental Health Care" website is expected to become a viable alternative for mental health support, helping to reduce social stigma and raise help-seeking intentions. Theoretically, this study serves as a technical reference for future developers and researchers in creating technology-driven solutions relevant to the mental health needs of the digital era. By addressing both the psychological barriers and the technical limitations of current platforms, this work contributes to the broader fields of health informatics and computer science. The subsequent sections of this paper move through the theoretical grounding of the work, the architectural and methodological decisions that shaped the system, the implementation and testing outcomes, and finally a critical reflection on what the results reveal — and what they leave open for future inquiry.

2. Related Work

2.1 State-of-the-art Research

Several studies have explored digital counseling platforms as a means to improve mental health service accessibility, with findings that are both encouraging and instructive about what remains unresolved. Nelyahardi *et al*. (2023) found that integrating social media and website-based tools for counseling at Universitas Jambi improved accessibility and was well received by students — though reliance on third-party platforms such as Instagram and WhatsApp limited control over data security and privacy. A systematic literature review by Prayoga *et al*. (2025) confirmed that digital counseling is increasingly preferred for its flexibility, while acknowledging persistent challenges in managing severe psychological cases. Prasetya *et al*. (2020) demonstrated that web-based cyber-counseling can raise students' help-seeking interest by reducing social stigma barriers. On the mental health literacy side, Azedarach and Ariana (2022) found that university students exhibited low help-seeking intentions despite possessing basic mental health knowledge, warning that undiagnosed conditions may lead to prolonged stress. Dewi *et al*. (2022) showed that even students with high depression literacy held poor help-seeking attitudes because available digital platforms lacked structured intervention features — a finding that points less to user failure and more to platform inadequacy. Andalasari and Safitri (2024) further identified a digital literacy gap between nursing and non-nursing students, underscoring the need for intuitive and accessible platform design.

From a technical standpoint, the picture is similarly uneven. Akbar *et al*. (2024) developed a web-based e-counseling program with educational features but without real-time communication. Ulum (2022) identified the risks of manual data management in counseling systems and proposed a web-based solution, though without a layered architecture or advanced security mechanisms. Hanapi and Harjono (2025) developed a cyber-counseling model for secondary schools but did not implement structured authentication despite handling sensitive user data. Hafidz *et al*. (2024) focused on mental health education through a progressive web application, again without interactive counseling capabilities. Regarding frontend technologies, Pamungkas *et al*. (2025) and Maharani (2025) validated Tailwind CSS for building responsive interfaces in student registration and company profile contexts, respectively, while Santoso (2025) reported that Tailwind CSS achieves a 15% reduction in CSS code compared to Bootstrap. On the backend side, Abdurrahman (2025) confirmed the effectiveness of Spring Boot for building stable RESTful APIs, though his implementation was confined to basic CRUD operations with a relational MySQL database.

2.2 Comparison with Previous Studies

Table 1 presents a comparative summary of related studies.

Table 1. Comparison of Related Studies

Study	Key Features	Real-time Chat	AI Integration	Structured Authentication
Nelyahardi *et al*. (2023)	Multi-platform counseling access	—	—	Third-party dependent
Akbar *et al*. (2024)	E-counseling with education	—	—	Not specified
Ulum (2022)	Digital counseling records	—	—	Basic
Hanapi & Harjono (2025)	Cyber counseling model	—	—	Not implemented

Hafidz *et al.* (2024)	Mental health education (PWA)	—	—	Not specified
Abdurrahman (2025)	Spring Boot RESTful API	—	—	Basic
This Study	Counseling, education, MBTI, e-books, payment	WebSocket	Gemini AI	Spring Security + OAuth2

The comparison makes one point difficult to ignore: no prior study combined real-time communication, AI-powered assistance, and structured authentication within a single mental health platform. Most earlier systems addressed either counseling or education in isolation, and applications of modern frontend frameworks like Tailwind CSS were confined to considerably simpler domains.

2.3 Positioning This Research

This study addresses four gaps identified in the literature. First, existing platforms lack real-time communication; this study implements WebSocket-based counseling (SockJS + StompJS) with a room-based architecture for direct client-psychiatrist interaction. Second, no prior study integrated AI for mental health support; this study incorporates a Gemini AI assistant via Spring AI to support user literacy. Third, previous systems exhibited weak architectures and inadequate security; this study adopts a three-layer architecture consisting of a frontend layer built with Tailwind CSS, a backend layer powered by Spring Boot with Spring Security — covering session cookies, CSRF protection, and Google OAuth2 authentication — and a database layer using MongoDB for flexible data storage. Fourth, Tailwind CSS usage in prior research was confined to simple applications; this study applies it to a complex interactive platform involving real-time chat, MBTI testing, and session-based payment workflows. Taken together, these four dimensions define a contribution that is not merely incremental — they represent a qualitatively different approach to what a digital mental health platform can and should do.

2.4 Review of Technologies, Frameworks, and Algorithms

Spring Boot is a Java-based framework that simplifies the creation of production-ready applications through auto-configuration and embedded server deployment. It supports RESTful API construction with built-in dependency injection and request mapping (Abdurrahman, 2025; Hartono & Mailoa, 2024), and recent studies confirm that it delivers faster and more stable response times when handling REST API requests compared to conventional PHP frameworks (Yulianto *et al.*, 2024). Spring Security extends this foundation by providing authentication and access control mechanisms, including CSRF protection, session management, and OAuth2 integration. Tailwind CSS complements the backend by offering a utility-first approach to frontend styling — granular control through low-level utility classes that enable responsive design with greater efficiency than component-based frameworks, achieving a 15% reduction in CSS code compared to Bootstrap (Santoso, 2025). MongoDB serves as the database layer, storing data in flexible JSON-like documents that accommodate the diverse data types required by the platform, including user profiles, chat messages, payment records, and MBTI results. WebSocket with SockJS and StompJS enables full-duplex real-time communication between clients and the server: SockJS provides fallback transport compatibility, while StompJS adds structured message routing for the room-based counseling chat. Finally, Spring AI with Gemini AI provides an abstraction layer for integrating artificial intelligence into the Spring Boot application, enabling the AI assistant to deliver on-demand mental health information to users. The Waterfall model was adopted as the development methodology for its sequential structure and well-defined phases — problem identification, analysis and design, implementation, and testing — which align with the scope and constraints of this academic research project.

3. Methodology

3.1 System Architecture

The system was designed using a decoupled, three-layer architecture that separates concerns between the user interface, business logic, and data persistence. Figure 1 presents the overall system architecture.

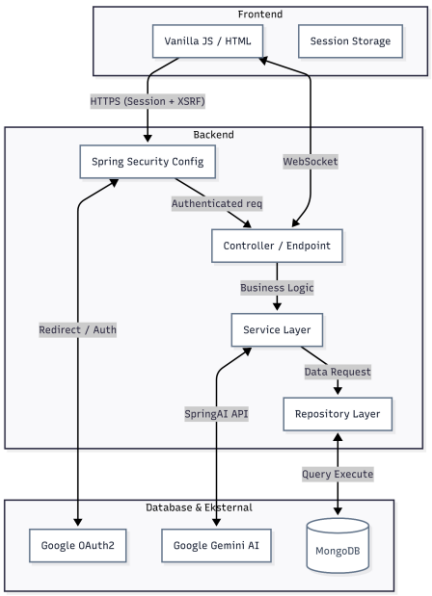

Figure 1. System Architecture Diagram

The frontend layer is built using Vanilla JavaScript (ES6 Modules) combined with Tailwind CSS. This layer handles interface rendering, user interaction, and server communication. The choice of Vanilla JavaScript maintains lightweight browser-side performance, while Tailwind CSS ensures a modern and responsive visual appearance — a capability shown to accelerate frontend development and guarantee responsiveness even in complex enterprise-scale applications (Azhariyah & Mukhlis, 2024; Kesuma *et al.*, 2024). The backend layer serves as the application control center, built on the Java Spring Boot framework. It manages all business logic, including user authentication through Spring Security, session cookie management (JSESSIONID), Cross-Site Request Forgery (CSRF) protection, and AI integration through Spring AI — processing and validating every request received from the frontend. The database layer uses MongoDB as the data management system, selected for its flexibility in handling JSON document-format data. Communication between layers occurs through two primary channels: REST API via HTTP for general data exchange, and WebSocket (SockJS + StompJS) for real-time counseling features. To map system functionality, a use case diagram was designed to illustrate the interaction between the two primary actors: Client (Patient) and Psychiatrist. Figure 2 presents the use case diagram.

Figure 2. Use Case Diagram

Both actors share access to core features including Login/Register, Dashboard, Edit Profile/Avatar, MBTI Test, E-Book access, and AI Chat powered by Gemini AI. Role-specific access restrictions are enforced to maintain the platform's business logic. The Client has exclusive access to the Consultation Payment feature, which serves as a mandatory prerequisite for initiating a professional counseling session, while the Psychiatrist is granted exclusive authority to manage consultation pricing through the dashboard. Feature dependencies through include relations ensure that authentication is required before accessing all primary features, and

MBTI test results are only accessible after completing the full test sequence. The Forgot Password feature operates independently as an account recovery mechanism.

3.2 Development Process

The Waterfall model was adopted as the development methodology for its sequential structure and well-defined phases — problem identification, analysis and design, implementation, and testing — which align with the scope of this academic research project. Figure 3 illustrates the stages of the Waterfall methodology.

Figure 3. Waterfall Methodology Diagram

The first phase involved problem identification, where the core issue of limited mental health service accessibility in Indonesia was examined. Observations revealed that existing counseling processes were still largely manual and prone to information loss (Ulum, 2022). This phase was supported by a literature study gathering scientific references on e-counseling systems, real-time communication protocols (WebSocket), and AI integration through Spring AI (Nelyahardi *et al.*, 2023; Akbar *et al.*, 2024). In the analysis and design phase, both functional and non-functional requirements were defined. Functional requirements included online counseling, psychological education articles, MBTI personality testing, and Google OAuth2 authentication. The database was designed using MongoDB, selected for its schema-less nature that provides high flexibility for dynamic data storage. User interface design was also prioritized, drawing on the customization capabilities of Tailwind CSS, which has been shown to offer a more efficient code structure compared to conventional frameworks (Santoso, 2025). The implementation phase translated the system design into working code. The backend was developed using Spring Boot to handle application logic, RESTful API construction, and security management through Spring Security. Real-time counseling chat was implemented via WebSocket with a room-based scheme, while the educational AI assistant was connected to the Google Gemini API through Spring AI. On the frontend, Tailwind CSS was used to build a modern, lightweight, and responsive interface (Pamungkas *et al.*, 2025; Maharani, 2025). Integration between frontend and backend was conducted intensively to ensure that data exchange through REST API and STOMP message mapping operated smoothly in accordance with the established pay-per-session business flow. The final phase involved testing and evaluation to validate system performance and functionality. Black-box Testing was used to verify that each feature functioned correctly without failure, with test cases systematically derived from the use case diagram (Figure 2) and the functional requirements defined during the analysis and design phase. Each test scenario was mapped to a specific use case to ensure comprehensive coverage, resulting in 12 test scenarios covering registration, authentication, counseling, payment, AI chat, and MBTI testing. Interface responsiveness testing was also performed to confirm usability across devices with varying screen sizes (Santoso, 2025). The testing phase was executed internally by the development team within a controlled environment; User Acceptance Testing (UAT) involving real patients and professional counselors was not conducted at this stage, as the priority was technical validation prior to clinical deployment.

3.3 Tools and Technologies

Table 2 summarizes the primary tools and technologies used in this study.

Table 2. Tools and Technologies

Category	Technology	Purpose
Backend Framework	Spring Boot (Java)	Application logic, RESTful API, server-side processing
Security	Spring Security	Authentication, session cookies, CSRF protection, OAuth2
Frontend Styling	Tailwind CSS	Responsive and modern UI design
Frontend Logic	Vanilla JavaScript (ES6)	Client-side interactivity and API communication
Database	MongoDB	Flexible document-based data storage
Real-time Communication	WebSocket (SockJS + StompJS)	Bidirectional real-time counseling chat
AI Integration	Spring AI + Google Gemini API	Intelligent mental health assistant
Authentication	Google OAuth2	Third-party login via Google accounts
Development Methodology	Waterfall Model	Sequential development process
Testing Method	Black-box Testing	Functional and responsiveness validation

4. Result and Discussion

4.1 Results

The implementation phase produced a fully functional "Mental Health Care" web platform based on the three-layer architecture described in the previous section. This section presents the directory structures of both the frontend and backend systems, followed by the key user interface pages.

4.1.1 Frontend Implementation

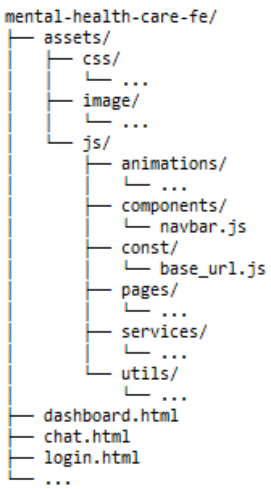

Figure 4. Frontend Directory Structure

The main directory consists of an assets/ folder serving as storage for static resources — including css/ for styling configuration, image/ for visual assets, and js/ as the center of JavaScript-based application logic. Within js/, the service/ subdirectory handles backend communication through API calls, and utils/ provides helper functions for authentication and security token management. The components/ folder stores reusable interface components such as the navigation bar, while const/ maintains global constants including backend URL configuration. HTML files such as dashboard.html, chat.html, and login.html serve as entry points for each page, integrated with their corresponding JavaScript modules. This modular structure allows independent development and testing of each component, improving scalability and code readability.

4.1.2 Backend Implementation

The backend follows the Repository Pattern, implemented through several primary folders. The controller/ folder handles HTTP requests and WebSocket-based communication for user authentication, data management, messaging, and payment processing. The service/ folder serves as the business logic layer, while repository/ manages interactions with the MongoDB database. The model/ folder represents data entity

structures stored as documents, and dto/ (Data Transfer Object) handles structured data exchange between backend and frontend layers. The mapper/ folder performs conversions between model objects and DTOs to maintain data consistency and security. The config/ folder contains system configurations including Spring Security settings, WebSocket configuration, and authentication encoders. The main entry point BackendApplication.java initializes the application. This structure supports modularity and separation of concerns, facilitating continuous development, testing, and maintenance.

```
mental-health-care-be/
└── src/main/java/com/mental/health/care/backend/
    ├── config/
    │   └── ...
    ├── controller/
    │   ├── AuthController.java
    │   ├── ChatMessageController.java
    │   ├── PaymentController.java
    │   └── ...
    ├── model/
    │   ├── BaseUser.java
    │   ├── Client.java
    │   ├── Psikiater.java
    │   └── ...
    ├── repository/
    │   └── ...
    ├── service/
    │   ├── UserService.java
    │   ├── PaymentService.java
    │   └── ...
    ├── dto/
    │   └── ...
    ├── mapper/
    │   └── ...
    └── BackendApplication.java
```

Figure 5. Backend Directory Structure

4.1.3 User Interface Implementation

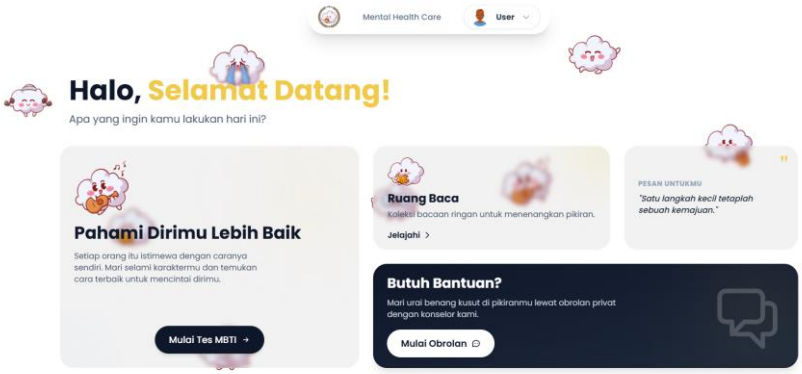

Figure 6. Dashboard Pages

Figure 6 shows the main dashboard page, which functions as the central navigation hub connecting users to various features including the MBTI personality test, E-Book reading room, and private counseling chat with psychiatrists. The interface was built using Tailwind CSS and Vanilla JavaScript to produce a responsive layout, with user profile data dynamically retrieved from MongoDB through REST API endpoints managed by the Spring Boot backend.

Figure 7. E-Book Reading Room Pages

Figure 7 displays the E-Book reading room, which provides a collection of psychological education materials aimed at supporting psychoeducation and raising users' mental health literacy. The E-Book catalog is managed by the Spring Boot backend, which retrieves data from the Google Books API and renders it on the frontend layer.

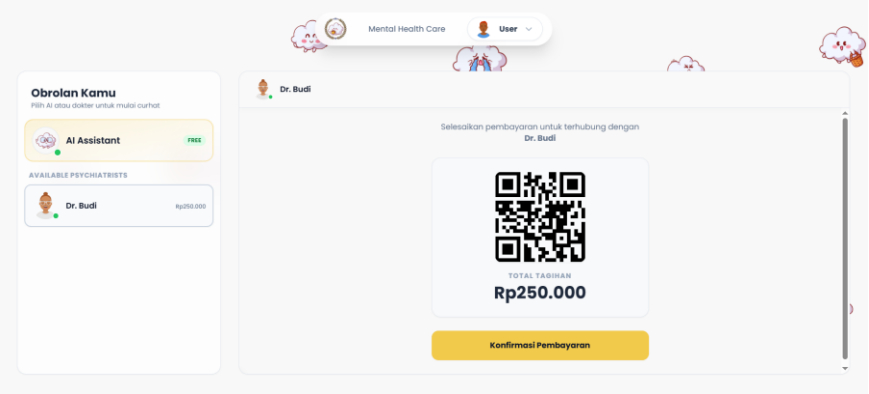

Figure 8. Payment Confirmation Pages

Figure 8 presents the pay-per-session payment confirmation page that users must complete before initiating a counseling session. The page displays a QR code (QRIS) along with the consultation fee previously set by the psychiatrist through the system. The transaction is tied to Spring Security and session cookie management for user access validation, with successful payment status recorded directly into MongoDB.

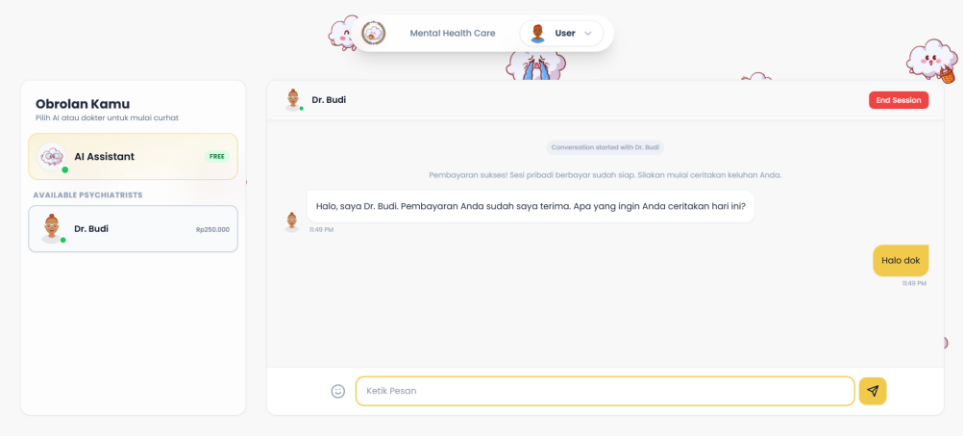

Figure 9. Counseling Chat Page

Figure 9 shows the real-time counseling chat page between Client and Psychiatrist. The private communication room activates automatically after the user confirms their session payment. Bidirectional message transmission is implemented using WebSocket combined with SockJS and StompJS over the Spring Boot framework, facilitating room-based data exchange with minimal latency. This synchronous communication channel is what makes the consultation environment genuinely interactive — not just technically functional.

4.1.4 Testing Results

Table 3 presents the results of interface responsiveness testing conducted across three device categories: Desktop (1920×1080), Tablet (768×1024), and Mobile (375×667).

Table 3. Interface Responsiveness Testing Results

Page	Device	Resolution	Test Result	Status
Dashboard	Desktop	1920×1080	Bento grid layout displays completely, sidebar navigation visible	Valid
Dashboard	Tablet	768×1024	Grid adjusts to 2 columns, navbar remains functional	Valid
Dashboard	Mobile	375×667	Grid becomes single column, hamburger menu active	Valid
E-Book	Desktop	1920×1080	E-book list displayed in grid format, images and text proportional	Valid
E-Book	Tablet	768×1024	E-book grid adjusts to 2 columns, content readable	Valid
E-Book	Mobile	375×667	E-book cards displayed in single column, text and images not cropped	Valid

Payment	Desktop	1920×1080	Payment form and session details displayed clearly	Valid
Payment	Tablet	768×1024	Payment form adjusts to screen width without overflow	Valid
Payment	Mobile	375×667	Payment form in full single column, confirmation button accessible	Valid
Chat	Desktop	1920×1080	Contact list panel and message panel displayed side by side	Valid
Chat	Tablet	768×1024	Panel layout adjusts, messages remain readable	Valid
Chat	Mobile	375×667	Chat panel fills screen, message input easily accessible	Valid

All 12 responsiveness test scenarios produced valid results with no elements cropped or overlapping across any device category. These findings align with Santoso (2025), who reported that Tailwind CSS achieves a 15% reduction in additional CSS lines compared to Bootstrap while providing high customization flexibility for precise cross-screen rendering. Pamungkas *et al*. (2025) showed that Tailwind CSS reduced registration processing time from an average of 10–15 minutes to 5 minutes, and Maharani (2025) confirmed that it accelerates responsive interface design. Both studies, however, applied Tailwind CSS in relatively simple domains — student registration and company profiles. The present study extends that application to a considerably more complex platform requiring interactive features such as real-time chat, MBTI testing, and pay-per-session payment processing, suggesting that the framework's responsiveness advantages hold even under greater functional demand. Table 4 presents the results of Black-box testing conducted to validate the core functionality of the system.

Table 4. Functionality Testing Results (Black-box Testing)

Feature	Test Scenario	Input Data	Expected Output	Status
Registration	Valid data registration	New email, username, password	Account created, welcome email sent	Valid
Registration	Duplicate email registration	Existing email	System rejects with "Email already registered!" message	Valid
Login	Correct credentials	Valid email and password	Session active, user enters Dashboard	Valid
Login	Incorrect password	Correct email, wrong password	"Wrong Password!" message displayed	Valid
Google Login	First-time Google Login	Valid Google account	Account auto-created, welcome email sent, enters Dashboard	Valid
Forgot Password	Reset with registered email	LOCAL account email	Reset password link email sent	Valid
E-Book	Open E-Book page after login	Active login session	Mental health e-book list loaded successfully	Valid
Payment	Simulate consultation payment	Valid patientId and psychiatristNoStr	PAID status active, payment confirmation email sent	Valid
Chat	Send message in active session	Text message, paid session	Message sent and appears real-time on both sides	Valid
Chat	End counseling session	Active session roomId	Session ended and session data removed from system	Valid
AI Chat	Send question to AI assistant	Mental health question text	Gemini AI response displayed in chat interface	Valid
MBTI Test	Complete all MBTI test questions	Answers to all questions	MBTI personality type calculated and result displayed	Valid

All 12 functional test scenarios produced valid outputs consistent with expectations, with no failures detected.

4.2 Discussion

The testing results confirm that the three-layer architecture operates as intended across all tested functionalities. On architecture and security, this study responds to weaknesses documented in prior research. Ulum (2022) reported that data management in online counseling applications remained manual with high risk of information loss — a problem the structured three-layer architecture adopted here directly addresses, with each layer maintaining clear responsibilities following the separation of concerns principle advocated by Abdurrahman (2025). While Abdurrahman (2025) demonstrated Spring Boot's effectiveness for building stable RESTful APIs, his implementation was confined to basic CRUD operations with a relational MySQL database. The present study extends that application by adapting Spring Boot to interact with MongoDB and integrating

Spring Security for session cookie management (JSESSIONID) and CSRF protection, with Google OAuth2 authentication added to further strengthen access control. These decisions also respond to the weaknesses identified by Nelyahardi *et al.* (2023), where reliance on third-party platforms such as Instagram and WhatsApp left user data security largely outside the system's control, and by Hanapi and Harjono (2025), whose cyber-counseling system lacked structured authentication despite 44% of students expressing a preference for website-based counseling.

The WebSocket implementation (SockJS and StompJS) with a room-based scheme stands as a key distinguishing feature of this study. Prayoga *et al.* (2025) found through systematic review that digital counseling is viable for urgent situations with limited time, though challenges remain for severe psychological cases. The synchronous bidirectional communication built into this platform enables direct client-psychiatrist interaction — a capability entirely absent from Akbar *et al.*'s (2024) system. The Gemini AI assistant, integrated through Spring AI, is equally novel among the reviewed prior studies. Its inclusion responds to a documented behavioral shift: young people are increasingly turning to AI as a first point of contact for sharing personal concerns and seeking preliminary psychological support (Norsely *et al.*, 2023). The assistant is designed to support mental health literacy by providing accessible, on-demand information — particularly relevant given that Azedarach and Ariana (2022) found that low mental health literacy among students leads to undiagnosed psychological conditions with potential for prolonged stress. A formal user study measuring the assistant's actual impact on literacy outcomes has not yet been conducted; initial interaction testing showed that the assistant processed and responded to a range of mental health queries, but that observation alone does not constitute evidence of literacy improvement. Structured user feedback in future work would be needed to validate this contribution. Andalasari and Safitri (2024) also identified a digital literacy gap among students from different academic backgrounds that limits access to web-based services — by combining educational features and professional counseling within a single responsive platform, this study offers a more complete response to that problem than any single-feature predecessor. One technical limitation warrants direct acknowledgment. Because system evaluation was conducted primarily in a controlled local environment, the testing did not capture WebSocket reconnection behavior under unstable network conditions, nor did it measure server response time under high concurrent user loads. Recognizing this constraint at the initial development stage is important for guiding scalability work before any public deployment.

5. Conclusion and Recommendations

This study designed and deployed the "Mental Health Care" website as a web-based counseling and psychological education platform, built on Spring Boot for the backend and Tailwind CSS for the frontend. The system rests on a three-layer architecture — frontend, backend, and MongoDB database — that enforces a clear separation between interface and business logic. Black-box testing across 12 functional scenarios confirmed that all features produced expected outputs with no failures detected. Interface responsiveness testing further confirmed that all primary pages rendered correctly across Desktop, Tablet, and Mobile devices, validating Tailwind CSS as an effective tool for adaptive interface design at this level of complexity. Taken together, the results indicate that the Spring Boot–Tailwind CSS combination can produce a structured, secure, and accessible digital counseling system — one that addresses both the technical shortcomings and the accessibility gaps documented in prior work.

This study offers several contributions to health informatics and web application development. First, the platform brings together professional counseling, psychological education, and self-assessment tools within a single system — a combination not found in any of the reviewed prior studies. Second, real-time counseling through WebSocket (SockJS and StompJS) with a room-based architecture directly addresses the absence of synchronous communication in previous e-counseling platforms. Third, the Gemini AI assistant integrated through Spring AI introduces an intelligent support feature for mental health literacy that earlier research had not explored. Fourth, the adoption of Spring Security with session cookie management, CSRF protection, and Google OAuth2 authentication provides a security framework appropriate for handling sensitive psychological data — a dimension that several prior systems either overlooked or treated superficially.

Several limitations should be stated plainly. The payment system currently operates as a simulation and has not been connected to a payment gateway for real-world transaction processing. The AI assistant provides general mental health information without personalization based on individual interaction history, which constrains its clinical relevance. The system was developed and tested exclusively as a web application, potentially limiting reach for users who primarily access services through mobile devices. System evaluation was confined to technical Black-box testing conducted internally by the development team — the absence of User Acceptance Testing (UAT) involving actual psychiatrists and target users means that clinical usability and real-world user experience have not been empirically validated. Quantitative performance benchmarking, including WebSocket response time, message latency, and server endurance under concurrent user loads, was

also not conducted. These are not minor oversights; they represent the boundary conditions of what this study can and cannot claim.

Based on these limitations, several directions for future development warrant attention. The payment mechanism should be connected to a real payment gateway — Midtrans, for instance — to support authentic transaction processing in a production environment. The AI assistant can be strengthened by incorporating conversation history-based personalization, enabling more contextual responses tailored to individual users over time. Development of a native mobile application for Android or iOS should be prioritized to extend the platform's reach beyond users with consistent desktop or browser access. Rigorous performance benchmarking should be carried out to quantify WebSocket message latency and evaluate server scalability under high concurrent user loads — data that would be indispensable before any public deployment. Equally important, future work should incorporate structured UAT involving real patients and licensed mental health professionals to empirically validate the platform's clinical usability and subjective user experience. Addressing these gaps would move the platform from a technically validated prototype toward a genuinely deployable mental health service.

References

Abdurrahman, D. (2025). Cake sales website based on Spring Boot and MySQL for vocational students. *Invotec*, *21*(1), 21–39. https://doi.org/10.17509/invotec.v21i1.82146

Akbar, Z., & Zakiah, E. (2024). Pengembangan program e-konseling berbasis web untuk meningkatkan kualitas bimbingan akademik dan kesehatan mental mahasiswa. *Jurnal Pendidikan Indonesia*, *5*(11), 1590–1598. https://doi.org/10.59141/japendi.v5i11.6287

Andalasari, N., & Safitri, O. (2024). Perbandingan literasi kesehatan mental pada mahasiswa keperawatan dan non-keperawatan. *Caring: Jurnal Keperawatan Al-Ikhlas*, *1*(1), 58–66. https://doi.org/10.70800/jckk.v1i1.125

Azedarach, M. R., & Ariana, A. D. (2022). Hubungan literasi kesehatan mental dengan intensi mencari bantuan pada mahasiswa. *Buletin Riset Psikologi dan Kesehatan Mental (BRPKM)*, *2*(1), 640–651. https://e-journal.unair.ac.id/BRPKM/article/view/36578

Azhariyah, S., & Mukhlis, M. (2024). Framework CSS: Tailwind CSS untuk front-end website store PT. XYZ. *Jurnal Informatika*, *3*(1), 12–20. https://doi.org/10.57094/ji.v3i1.1601

Dewi, F. V., Adianta, A., & Parwati, M. (2022). Hubungan antara literasi kesehatan mental depresi dan stigma diri dengan sikap mencari bantuan masalah kesehatan mental pada mahasiswa keperawatan. *Jurnal Riset Kesehatan Nasional*, *6*(2), 125–133. https://doi.org/10.37294/jrkn.v6i2.438

Hafidz, H., Al Huda, F., & Kharisma, A. P. (2024). Pengembangan aplikasi edukasi kesehatan mental berbasis progressive web app. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *8*(3), 112–120.

Hanapi, K., & Harjono, H. S. (2025). Development of a web-based cyber counseling model for guidance and counseling services at SMA Negeri 1 Bayung Lencir. *Indonesian Journal of Educational Development (IJED)*, *5*(4), 498–507. https://doi.org/10.59672/ijed.v5i4.4500

Hartono, M. A., & Mailoa, E. (2024). Pembuatan website REST API ensiklopedia tumbuhan dengan kombinasi framework Spring Boot dan MyBatis Generator. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, *5*(2), 1508–1520. https://doi.org/10.35870/jimik.v5i2.712

Kesuma, K. A. B. W., Wijaya, I. N. Y. A., & Putra, I. G. J. E. (2024). Implementasi Next.js, TypeScript, dan Tailwind CSS untuk pengembangan aplikasi frontend sistem inventory perusahaan APAR (Studi kasus: CV Indoka Surya Jaya). *Jikom (Jurnal Informatika dan Komputer)*, *14*(2), 95–108. https://doi.org/10.55794/jikom.v14i2.195

Maharani, P. (2025). Pengembangan website PT. Rantangin Digital Indonesia menggunakan framework Next.js dan Tailwind CSS. *Repeater: Publikasi Teknik Informatika dan Jaringan*, *3*(1), 129–137. https://doi.org/10.62951/repeater.v3i1.355

Nelyahardi, N., Wahyuni, H., Sekonda, F. A., & Fahrianti, F. (2023). Digital based e-counseling application on Instagram, WhatsApp, and website to support services counseling at Universitas Jambi. *Jurnal Konseling dan Pendidikan*, *11*(4), 276–287. https://doi.org/10.29210/1101000

Norsely, F., Arviani, H., & Achmad, Z. A. (2023). Pengalaman interaksi pengguna remaja curhat dengan ChatGPT. *Komunikologi: Jurnal Pengembangan Ilmu Komunikasi dan Sosial*, *7*(2), 120–135. https://doi.org/10.30829/komunikologi.v7i2.16653

Pamungkas, G. D., Parwati, Y., & Putranto, B. D. (2025). Pengembangan aplikasi pendaftaran siswa baru berbasis web dengan React.js dan Tailwind CSS. *Jurnal Algoritma*, *22*(1), 37–48. https://doi.org/10.33364/algoritma/v.22-1.2135

Prasetya, A. D., Sugiyo, S., & Japar, M. (2019). Web-based cyber counseling to improve students' counseling interests. *Jurnal Bimbingan Konseling*, *8*(4), 144–150.

Prayoga, A., Habsy, B. A., & Purwoko, B. (2025). Keberhasilan konseling berbasis digital: Kajian systematic literature review. *G-COUNS: Jurnal Bimbingan dan Konseling*, *9*(2), 1410–1419. https://doi.org/10.31316/g-couns.v9i2.7226

Saba, S. S. (2024). Membangun profesionalisme dalam era teknologi: Transformasi layanan bimbingan konseling online. *JBK Jurnal Bimbingan Konseling*, *2*(1), 17–24.

Santoso, M. F. (2025). Perbandingan efektivitas Bootstrap dan Tailwind CSS dalam pengembangan UI web responsif. *Jurnal Teknologi dan Sistem Informasi Bisnis*, *7*(4), 489–497. https://doi.org/10.47233/jteksis.v7i4.2260

Ulum, M. (2022). Rancang bangun aplikasi konseling online berbasis web (Studi kasus: Merly's Consulting). *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, *10*(4), 530–541. https://jurnal.untan.ac.id/index.php/52216

Yulianto, R., Mardiana, M., Pradipta, R. A., & Nama, G. F. (2024). Performance comparison analysis of Spring Boot and Laravel frameworks using API web service. *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, *12*(2). https://doi.org/10.23960/jitet.v12i2.4141.