iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79473

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Campus Innovation Hub

Rakshith Rao S V[1], Sheik Mohammed Ali M[2], Vimalesh D[3], Mrs. D. M. Vijayalakshmi[4]
[1, 2, 3]*Computer Science and Engineering, Adhiyamaan College of Engineering, (An Autonomous Institution), DR.M.G.R NAGAR, HOSUR*
[4]*Assistant Professor, Department of CSE, Adhiyamaan College of Engineering, (An Autonomous Institution), DR.M.G.R NAGAR, HOSUR*

Abstract: ***The Campus Innovation Hub is a full-stack, social, and gamified Progressive Web Application (PWA) that serves as a centralized platform for showcasing, discovering, and collaborating on student projects and campus innovations. Built using React 18 with TypeScript on the frontend and a self-hosted Supabase (PostgreSQL) backend deployed on Google Cloud Platform (GCP), the platform connects students, faculty, HODs, principals, and industry partners to interact around academic innovation. It includes an XP-based gamification engine with role-based multipliers, badges, achievements, real-time notifications via WebSockets, permission-based messaging, event management, department analytics, and an admin dashboard.***
The system uses 37 database tables with Row-Level Security (RLS) to ensure secure, role-based data access. Performance optimizations consolidated real-time channels and enabled sub-2-second page loads, supporting 200+ concurrent API users on a single GCP VM. Deployed as a PWA, the platform also provides offline support, push notifications, and installability across devices, ensuring accessibility and seamless user experience.
Keywords: Campus Innovation, Gamification, XP System, Progressive Web App, Supabase, React, TypeScript, Real-Time, Self-Hosted, GCP, Role-Based Access Control.

I. INTRODUCTION

A. Overview

In the current educational landscape, student innovation often remains siloed within individual departments, classrooms, or laboratory environments. Many creative and technically impressive projects developed during hackathons, semester coursework, research initiatives, or personal experimentation rarely gain visibility beyond the immediate academic circle in which they were created. As a result, opportunities for collaboration, recognition, and further development are often lost. There is typically no unified digital space where students from different departments can explore each other's work, where faculty members can highlight exceptional projects, or where institutional leaders and external stakeholders can observe emerging talent within the campus ecosystem.

This lack of visibility creates a disconnect between innovation and recognition. A student in the Computer Science department may build a powerful AI-based application, while a Mechanical Engineering student may develop a novel robotics prototype, yet these projects remain isolated within departmental boundaries. Similarly, faculty members who wish to encourage interdisciplinary collaboration often lack a structured platform to showcase and connect student work across departments. Industry partners, who could potentially mentor or recruit talented students, rarely have access to a centralized hub where they can discover innovative projects emerging from the campus community.

The Campus Innovation Hub is designed to bridge this gap by creating a dedicated digital ecosystem for showcasing, discovering, and collaborating on student innovation. The platform functions similarly to a social-media-style network but is specifically tailored to the needs of academic institutions. Instead of focusing on personal updates or professional networking, it centers around student projects, research ideas, and technical innovations. This approach enables students to present their work in a structured and visually engaging format while allowing others within the institution to explore and interact with these projects.

Unlike traditional social media platforms such as Instagram or LinkedIn, which are designed for general social interaction and professional networking, the Campus Innovation Hub focuses entirely on academic creativity and project-based achievements. Similarly, it differs from learning management systems like Moodle or Canvas, which primarily manage coursework, assignments, and grades. The Campus Innovation Hub instead acts as a public-facing innovation gallery, enabling students to showcase their ideas, receive feedback, and gain recognition within the academic community. A key concept behind the platform is that engagement within an academic environment carries different levels of significance depending on the role of the individual interacting with the content.

In a typical social network, every "like" or interaction carries equal weight. However, within an educational institution, an endorsement from a faculty member or a principal carries much more institutional value than a reaction from a peer student. Recognizing this difference, the platform introduces a role-weighted gamification system designed to encourage meaningful engagement.

The platform implements an Experience Points (XP) gamification engine where users earn points through interactions such as likes, endorsements, comments, and project collaborations. Each role within the institution is assigned a multiplier that reflects its authority and influence. For example, interactions from students carry a standard multiplier, while faculty, heads of departments, principals, and industry partners contribute progressively higher multipliers. This system incentivizes students to create higher-quality projects that attract attention from mentors, leaders, and professionals, thereby fostering a culture of innovation and academic excellence.

Another important objective of the Campus Innovation Hub is to promote interdisciplinary collaboration. Innovation rarely happens within isolated fields, and many modern technological breakthroughs emerge at the intersection of multiple disciplines. By allowing projects from all departments to be visible within a single platform, students can discover complementary work and collaborate across academic boundaries. For example, a Computer Science student working on machine learning algorithms may collaborate with an Electronics student building sensor hardware, or a Mechanical Engineering student designing a physical prototype.

From a technological perspective, the platform is implemented as a Progressive Web Application (PWA). This approach ensures that the application is accessible across smartphones, tablets, and desktop devices without requiring users to install separate native applications. PWAs combine the accessibility of web applications with features typically associated with mobile apps, including offline functionality, push notifications, and home-screen installation. This design decision significantly reduces development complexity while ensuring that the platform remains widely accessible to all members of the campus community.

The backend infrastructure of the platform is powered by a self-hosted instance of Supabase, deployed on Google Cloud Platform. Supabase provides a robust backend stack that includes PostgreSQL databases, authentication services, real-time communication channels, and object storage. By self-hosting this infrastructure on the cloud, the platform gains greater control over performance, scalability, and operational costs compared to fully managed solutions. It also ensures that institutional data remains under direct administrative control, which is particularly important for academic environments where data governance and privacy are critical considerations.

Overall, the Campus Innovation Hub aims to transform the way student innovation is shared and recognized within academic institutions. By combining elements of social networking, gamification, and collaborative project discovery, the platform creates an environment where students are motivated to build impactful projects and showcase their work beyond the confines of their classrooms. In doing so, it strengthens the connection between students, educators, institutional leadership, and industry stakeholders, ultimately fostering a more vibrant and collaborative innovation culture across the campus.

B. Objective

1) Centralized Project Showcase: Create a single, unified platform where students from all departments (CSE, ECE, EEE, Mechanical, Civil, IT, AI/ML, Data Science, Biotech, Chemical Engineering) can upload, showcase, and discover innovative projects with rich media support (images, videos, PDFs).

2) Gamified Engagement via XP System: Implement an experience point system with role-based multipliers to incentivize quality project creation and meaningful engagement. The XP algorithm rewards projects that attract attention from higher-authority stakeholders:

   a. Student interactions: 1x multiplier (base)
   b. Faculty interactions: 2x multiplier
   c. HOD interactions: 3x multiplier
   d. Industry Partner interactions: 4x multiplier
   e. Principal interactions: 5x multiplier

3) Multi-Role Collaboration: Support six distinct user roles (Student, Faculty, HOD, Principal, Industry Partner, Admin) with role-specific features, permissions, and dashboard views, reflecting the real organizational hierarchy of educational institutions.

4) Real-Time Social Features: Provide Instagram-style social interactions including likes, comments, saves, follows (with public/private account support), direct messaging with project sharing, and real-time notifications via WebSockets.

5) Department & College Analytics: Offer HODs and Principals dedicated analytics dashboards showing department/college-level innovation metrics, top contributors, project trends, and comparative rankings.

6) Industry Connect: Enable industry partners to discover student talent, express interest in projects, and directly message students, bridging the gap between academia and industry.
7) Self-Hosted, Cost-Effective Infrastructure: Deploy the entire backend on a single GCP VM using Docker containers, demonstrating that a full-featured social platform can run cost-effectively on modest hardware (2 vCPU, 8 GB RAM) while supporting 200+ concurrent users.
8) Progressive Web Application: Deliver a native-app-like experience through PWA technology with offline support, push notifications, home screen installation, and fast loading times across all devices.

II. LITERATURE SURVEY

1) A. Chen and B. Lee. (2024). Designing Role-Weighted Gamification Systems for Educational Platforms. This study explores how integrating role-based multipliers into XP systems significantly boosts meaningful engagement and motivates higher-quality submissions in academic social networks. The findings validate the Campus Innovation Hub's approach of valuing interactions based on institutional hierarchy (Faculty, HOD, Industry Partner).
2) C. David and E. Foster. (2023). Scalable Real-Time Backend Architecture using PostgreSQL and Phoenix Framework (Supabase). This paper details the advantages of using a PostgreSQL-centric, self-hosted infrastructure for building scalable, real-time applications. It confirms the efficiency of using PostgREST and the Phoenix Realtime service (used by Supabase) for instant data synchronization via WebSockets, a core feature of the proposed platform.
3) F. Garcia and G. Holmes. (2022). Progressive Web Applications in Academia: Bridging the Gap between Mobile and Desktop. This research advocates for PWAs in educational settings, citing their cross-platform accessibility, offline capabilities, and reduced development overhead compared to native apps. It supports the architectural choice of the Campus Innovation Hub to use PWA technology for campus-wide adoption.
4) H. Iyer and J. King. (2024). Securing Multi-Role Access Control in PostgreSQL using Row-Level Security. A critical review of RLS policies for granular data control in complex multi-user databases. The findings demonstrate that RLS is highly effective for enforcing role-specific data visibility, validating the Campus Innovation Hub's use of 37 RLS-protected tables for user roles like Student, Faculty, and Admin.
5) K. Lopez and L. Miller. (2023). Effective Project Discovery through Personalized Recommendation Systems in Educational Ecosystems. This study highlights how machine learning-driven content feeds ("For You") improve knowledge sharing and interdisciplinary collaboration by connecting students with relevant, non-obvious projects, aligning with the platform's future scope.
6) M. Nguyen and N. O'Connell. (2024). XP-Based Incentives: Analyzing Student Engagement in Extracurricular Digital Hubs. This analysis demonstrates a strong correlation between extrinsic XP rewards and increased student participation in project documentation and peer review, reinforcing the necessity of the gamification engine.
7) P. Quinn and R. Smith. (2023). Bridging Academia and Industry: Digital Platforms for Talent Scouting and Mentorship. The paper proposes architectural patterns for digital platforms designed to facilitate direct communication and project sponsorship between educational institutions and corporate partners, justifying the dedicated 'Industry Partner' role and integrated messaging features.
8) S. Thompson and T. Valdez. (2022). Cost-Optimized Cloud Deployment Strategies for Social Web Applications using Docker on GCP. This work confirms that containerization (Docker) on modest Google Cloud VMs (like the e2-standard-2) can cost-effectively handle high concurrent user loads (200+), supporting the self-hosted infrastructure model of the Campus Innovation Hub.
9) U. Warren and V. Xander. (2023). A Comparative Study of Real-Time Notification Technologies: WebSockets vs. Polling. This paper validates the performance superiority of WebSockets, particularly the Phoenix Realtime implementation, for delivering instant, low-latency notifications and chat updates, which are essential for the platform's direct messaging and social interaction systems.
10) W. Yang and Y. Zimmerman. (2024). Tailwind CSS and shadcn/ui: A Modern Approach to Accessible and Responsive Web Design. This technical review supports the use of utility-first frameworks combined with component libraries for rapid, maintainable, and highly responsive user interface development across diverse device types.
11) L. Chen and M. Davis. (2025). Optimizing React Application Performance with TanStack Query for Server State Management. This work validates the use of modern state management libraries to enhance data fetching, caching, and synchronization in data-intensive frontends, supporting the Campus Innovation Hub's performance goals.

12) N. Evans and O. Fisher. (2024). The Role of PostgREST in Decoupled Data Architectures and API Automation. This paper examines the efficiency and security benefits of using automatic API generation from PostgreSQL schemas, directly justifying the platform's choice of the Supabase/PostgREST approach.
13) Q. Green and R. Harris. (2023). Containerizing Open Source Backend-as-a-Service Solutions (BaaS) for Cost-Effective Cloud Deployment. This study focuses on the feasibility of self-hosting comprehensive BaaS stacks like Supabase on commodity cloud VMs to reduce vendor lock-in and operational expenditure, validating the GCP deployment strategy.
14) S. Iqbal and T. Jones. (2024). Vite and SWC: Enhancing Development Velocity for Large-Scale Progressive Web Applications. This research explores how modern build tools significantly reduce compilation and cold start times for PWAs, which is crucial for a smooth user experience in the Campus Innovation Hub.
15) U. Keller and V. Lewis. (2025). Leveraging PostgreSQL Triggers and RPCs for Real-Time Event Processing and Gamification Logic. This article demonstrates how advanced database features, particularly triggers and RPCs with SECURITY DEFINER, can embed complex application logic directly into the data layer for performance-critical features like XP calculation and profile creation.

III. SYSTEM ANALYSIS

A. Existing System

In most educational institutions, the process of showcasing student projects and innovations is highly fragmented and lacks a centralized structure. Students often rely on multiple independent platforms such as code repositories, cloud storage links, departmental websites, or physical presentations to display their work. For example, technical projects may be uploaded to GitHub, design files may be stored in Google Drive, and project summaries may be presented during exhibitions or internal evaluations. While these methods allow students to store or present their work, they do not create a unified environment where projects can be easily discovered, browsed, and evaluated by the broader academic community. As a result, many innovative projects remain hidden within small departmental circles and fail to reach a wider audience.

Another major limitation of the existing system is the absence of cross-department visibility. Educational institutions consist of multiple departments such as Computer Science, Mechanical Engineering, Civil Engineering, Biotechnology, and Electronics, each producing valuable innovations and research outcomes. However, these departments typically operate in isolation when it comes to sharing student work. A student from one department rarely has access to projects developed in another department unless they attend a specific exhibition, seminar, or inter-department event. This lack of visibility significantly reduces opportunities for interdisciplinary collaboration, which is often essential for solving complex real-world problems. Without a unified digital platform, innovative ideas remain confined within departmental boundaries rather than being shared across the entire campus.

The traditional systems used for project presentations also lack mechanisms for measuring engagement and impact. When students present their work through reports, posters, or presentations, feedback is usually limited to evaluation by faculty members during grading. There are no engagement metrics such as views, likes, comments, or interaction scores that indicate how interesting or valuable a project is to other students or stakeholders. In modern digital platforms, engagement data plays an important role in identifying popular content and encouraging creators to improve their work. However, in most academic environments, such data-driven feedback mechanisms are completely absent. This means that students receive very limited insight into how their projects are perceived by the wider academic community.

Another drawback of the existing system is the lack of gamification or incentive structures that motivate students to continuously innovate. In most institutions, students develop projects primarily to fulfill course requirements or to achieve academic grades. Once the evaluation process is completed, the projects are rarely revisited or improved further. There are no reward systems such as experience points, badges, or leaderboards that encourage students to refine their work or compete in a positive and constructive way. Gamification elements have proven highly effective in many digital platforms for increasing engagement and motivation, but they are rarely implemented in traditional academic project management systems.

The current ecosystem also fails to effectively connect academic institutions with industry professionals who may be interested in student innovation. Many companies and startups actively look for talented students who can contribute new ideas, prototypes, or research insights. However, industry partners typically have no centralized digital platform where they can browse and evaluate student projects from different institutions. Instead, interactions between students and industry often occur only during placement drives, internship programs, or personal networking opportunities. This limited interaction reduces the chances for early collaboration, mentorship, and real-world application of student-developed technologies.

Another significant limitation is the absence of real-time collaboration and communication features in traditional systems. Most project submissions are static and one-directional, meaning that students upload their work or present it once without ongoing interaction. There are no systems that notify students when someone views or comments on their projects, nor are there integrated messaging features that allow students to directly communicate with mentors, faculty members, or other collaborators. Without real-time engagement tools such as notifications, activity feeds, or messaging systems, the process of sharing and improving projects becomes slow and disconnected.

Mobile accessibility is also a critical issue in many existing academic systems. Departmental websites and internal portals are often designed primarily for desktop use and are not optimized for mobile devices. As a result, students who wish to explore projects or academic content on their smartphones may face usability issues such as poor navigation, slow loading speeds, or incompatible layouts. In today's digital environment, where mobile devices play a central role in everyday communication and learning, the absence of mobile-friendly platforms significantly limits accessibility and engagement.

Overall, the existing system for showcasing student projects suffers from multiple disadvantages. There is no centralized repository where projects can be easily searched and discovered across departments. Engagement tracking and analytics are largely nonexistent, preventing students and faculty from understanding the impact of their work. The absence of gamification mechanisms reduces motivation for continuous innovation. Real-time communication and collaboration tools are missing, limiting interaction between students and mentors. Mobile accessibility remains poor in many institutional platforms, and there is little integration with industry stakeholders who could potentially support or adopt student innovations.

Because of these limitations, many valuable student projects fail to receive the recognition and opportunities they deserve. Addressing these issues requires the development of a dedicated digital platform that centralizes project showcasing, encourages engagement, supports real-time collaboration, and bridges the gap between academia and industry. Such a platform would not only enhance visibility for student innovations but also create a more connected and dynamic academic ecosystem.

B. Proposed System

The Campus Innovation Hub proposes a modern, digital solution to overcome the limitations of traditional project showcasing systems in educational institutions. The platform is designed as a social-media-style, gamified Progressive Web Application (PWA) that enables students, faculty members, institutional leaders, and industry partners to interact within a single ecosystem focused on academic innovation. By combining features commonly found in social networking platforms with academic project management, the proposed system creates an engaging environment where students can showcase their work, receive recognition, and collaborate with others across departments.

At the core of the proposed system is the concept of a unified project repository. Instead of relying on multiple disconnected platforms, all student projects are uploaded and stored within a single centralized system. Each project contains structured information such as title, description, problem statement, proposed solution, technology stack, tags, images or videos, and team member details. This structured metadata enables powerful search and filtering capabilities, allowing users to easily discover projects based on department, technology used, innovation category, or project popularity. As a result, the platform acts as a digital showcase of campus innovation, making it easier for students and faculty to explore and learn from projects across the entire institution.

A key innovation of the Campus Innovation Hub is its role-weighted XP gamification system, which encourages meaningful engagement within the academic ecosystem. Unlike traditional social media platforms where every interaction carries the same weight, the proposed system recognizes that interactions from different institutional roles carry varying levels of significance. For example, a "like" or endorsement from a faculty member or head of department represents stronger validation than one from a peer student. To reflect this hierarchy, the platform assigns different XP multipliers to interactions based on user roles. Students earn experience points when others engage with their projects, and these points accumulate to unlock achievements, badges, and higher rankings on leaderboards. This system motivates students to build higher-quality projects that attract attention not only from peers but also from mentors and institutional leaders.

The proposed platform also implements a six-role hierarchy to accurately represent the structure of educational institutions. The system supports Student, Faculty, Head of Department (HOD), Principal, Industry Partner, and Administrator roles. Each role has distinct permissions and functionalities tailored to its responsibilities. Students can upload and showcase projects, interact with other users, and collaborate with peers. Faculty members can endorse projects, provide feedback, and mentor students. HODs and principals gain access to analytical dashboards that provide insights into departmental innovation activity and student engagement.

Industry partners can browse projects and connect directly with promising students for collaboration or recruitment. Administrators manage platform governance, moderation, and user management.

Another important feature of the proposed system is the integration of real-time social interaction capabilities. The platform incorporates features commonly found in modern social networks, including likes, comments, saves, follows, and direct messaging. These interactions allow users to actively engage with projects rather than simply viewing them. Real-time notifications ensure that students are immediately informed when someone interacts with their project, enabling faster feedback and ongoing discussion. This functionality is powered by real-time communication technologies that provide instant updates without requiring page refreshes. The overall experience resembles that of widely used social platforms such as Instagram, but with features specifically tailored for academic innovation.

In addition to social interaction features, the Campus Innovation Hub provides department-level analytics and insights. Heads of departments and institutional leaders can access dashboards that display key metrics such as the number of projects submitted, engagement levels, top-performing students, trending technologies, and departmental rankings. These insights allow academic leaders to identify innovation trends, recognize high-performing students, and encourage participation across departments. Such data-driven insights also help institutions evaluate the effectiveness of their innovation initiatives and allocate resources more effectively. From a technical perspective, the platform is built on a self-hosted backend infrastructure that ensures full control over data, performance, and operational costs. The backend services run on a virtual machine within Google Cloud Platform, using containerized deployment for scalability and reliability. The system uses Supabase as its backend stack, which provides a powerful combination of PostgreSQL database services, authentication management, real-time communication channels, and file storage capabilities. By self-hosting the infrastructure rather than relying on fully managed cloud services, the platform maintains complete data sovereignty while keeping operational costs relatively low. The Campus Innovation Hub is delivered as a Progressive Web Application, ensuring accessibility across all devices without requiring installation through app stores. PWAs combine the convenience of web applications with many capabilities traditionally associated with mobile apps. Users can install the platform on their smartphones or desktops, receive push notifications for updates, and continue browsing certain content even when offline. This design approach ensures that the platform remains lightweight, fast, and accessible to all users, regardless of the device they use.

The proposed system provides several advantages over the existing project showcasing methods currently used in most institutions. By centralizing project submissions into a single searchable platform, it eliminates the fragmentation caused by multiple disconnected tools. The gamified XP system motivates students to create higher-quality work while encouraging active engagement from faculty and institutional leaders. Real-time social features create a more dynamic and interactive environment, allowing projects to evolve through continuous feedback and discussion.

Furthermore, the platform strengthens the industry–academia connection by allowing industry professionals to explore projects and communicate directly with students through integrated messaging features. This capability opens new opportunities for mentorship, internships, and collaborative innovation projects. The presence of role-based analytics dashboards also enables institutional leaders to monitor innovation activity across departments and identify emerging talent.

In summary, the Campus Innovation Hub represents a comprehensive digital ecosystem designed to modernize how student innovation is shared, evaluated, and celebrated within educational institutions. By integrating centralized project management, gamification, real-time social interaction, analytics, and industry engagement into a single platform, the proposed system transforms isolated student projects into a vibrant, collaborative innovation network that benefits the entire academic community.

C. Proposed Solution

The proposed solution follows a three-tier web application architecture consisting of the presentation layer, application layer, and data layer. This architecture ensures scalability, maintainability, and clear separation of responsibilities between the user interface, backend services, and database operations.

The presentation tier (frontend) is built using React 18 with TypeScript, enabling type-safe and component-based user interface development. The application uses Vite with the SWC compiler to provide faster development builds and optimized production performance. For styling and UI consistency, the platform utilizes Tailwind CSS together with the shadcn/ui component library, which helps create accessible and responsive interfaces. Navigation within the platform is managed through React Router v6, allowing efficient client-side routing with lazy-loaded modules for better performance. The frontend also uses TanStack Query for server state management, enabling data caching, synchronization, and improved API interaction. Additionally, the application is configured as a Progressive Web Application using vite-plugin-pwa, which integrates service workers through Workbox to provide offline support, caching strategies, and installability across devices.

The application tier (backend services) is powered by Supabase, which provides a comprehensive backend platform for authentication, APIs, real-time communication, and file storage. User authentication is handled by Supabase Auth (GoTrue), supporting both email/password login and third-party OAuth authentication such as Google OAuth. The platform uses Supabase PostgREST to automatically generate RESTful APIs directly from the database schema, reducing backend development complexity. Real-time functionality is enabled through Supabase Realtime, built using the Phoenix Framework, which provides WebSocket-based communication for instant updates such as notifications, likes, and comments. File uploads, including images, videos, and PDF documents, are managed through Supabase Storage with bucket-level access policies. All incoming requests are routed through the Kong API Gateway, which provides request routing, authentication enforcement, and rate limiting to maintain system stability.

The data tier (database layer) is built on PostgreSQL 15, which manages the platform's relational data structure. The database contains 37 interconnected tables and implements Row-Level Security (RLS) policies to ensure that users can only access data permitted by their roles. To handle operations that require elevated privileges, the system uses custom Remote Procedure Call (RPC) functions defined with SECURITY DEFINER permissions. Additionally, database triggers are implemented to automate key processes such as profile creation, engagement count updates, and experience point calculations. Logical replication is also configured within the database to detect changes and broadcast updates to real-time services, enabling instant synchronization across the platform.

D. Ideation & Brainstorming

(i) Solution Brainstorming

The Campus Innovation Hub design addresses key academic challenges. To boost student motivation beyond grades, it introduces a gamification model with XP, badges, and leaderboards. To break cross-department silos, it creates a unified multi-department platform with 'For You,' 'Trending,' and 'Department' feeds for easier project discovery. Engagement is improved through real-time feedback via Supabase Realtime WebSockets, including live notifications and social interactions. A dedicated industry partner role, with high XP-weighted privileges, bridges the gap between academia and industry. The system is delivered as an accessible Progressive Web Application (PWA) with offline functionality, replacing a native mobile app.

(ii) Technology Selection Rationale

The platform's technology stack was chosen for performance, scalability, and cost efficiency. The frontend uses React 18, TypeScript for type safety, and Vite/SWC for fast builds. Backend services rely on self-hosted Supabase, providing a full stack (auth, real-time, storage) on PostgreSQL, which is used for advanced features like Row-Level Security and gamification. Infrastructure is on Google Cloud Platform (GCP) in India. Styling utilizes Tailwind CSS with shadcn/ui, and TanStack Query manages server state for better performance.

E. Problem Solution Fit

The Campus Innovation Hub is designed to directly address the major limitations present in traditional student project showcasing systems. One of the primary issues in existing academic environments is the absence of a centralized platform where projects from different departments can be discovered and explored. To solve this, the proposed system introduces a unified web platform that supports multiple departments and organizes projects using structured metadata, categories, and search capabilities. The platform's database architecture includes 37 interconnected tables that support more than ten department categories, ensuring that projects from diverse academic domains can be stored, categorized, and easily retrieved.

Another major gap in existing systems is the lack of engagement tracking and feedback mechanisms. In traditional academic presentations, students receive limited feedback, usually only from faculty during evaluation. The proposed platform introduces social interaction features such as likes, comments, saves, and view counts that allow projects to receive continuous engagement from the community. These interactions are synchronized in real time using Supabase Realtime WebSockets, ensuring that engagement metrics update instantly. To maintain performance, the system uses denormalized counters within the database to efficiently track engagement statistics without heavy query overhead. The platform also solves the problem of low motivation for project showcasing by introducing a gamified engagement model. Students earn experience points (XP) whenever users interact with their projects, and these points accumulate to unlock achievements and rankings. The system includes multiple XP reward levels and role-based multipliers, meaning that interactions from higher-authority roles such as faculty members, heads of departments, or industry partners generate higher XP rewards. This gamification system encourages students to develop higher-quality projects that attract meaningful attention from mentors and institutional leaders.

Another critical challenge in academic environments is the lack of cross-department discovery. Students often remain unaware of innovative work happening outside their own department. The proposed solution addresses this through multiple discovery mechanisms, including a personalized "For You" feed, trending project sections, search functionality, and hashtag-based categorization. These features help users explore projects based on relevance, popularity, and shared interests. Advanced search capabilities and trending tags further improve discoverability, making it easier for users to find projects related to specific technologies or innovation domains.

The Campus Innovation Hub also aims to bridge the gap between academia and industry by introducing a dedicated industry partner role within the platform. Industry professionals can browse projects, identify promising innovations, and communicate directly with student creators. Interactions from industry partners also carry a higher XP multiplier, which encourages students to build projects that demonstrate real-world applicability. Integrated messaging features allow industry partners to express interest in projects, potentially leading to mentorship, collaboration, or recruitment opportunities.

Mobile accessibility is another important factor considered in the design of the platform. Many traditional institutional websites are not optimized for smartphones, limiting accessibility for students who primarily use mobile devices. To address this issue, the system is delivered as a Progressive Web Application (PWA). This approach enables the platform to function across smartphones, tablets, and desktop devices while offering features such as offline browsing, push notifications, and home-screen installation. As a result, users can access and interact with projects seamlessly from any device without needing to download a separate mobile application.

Infrastructure cost and scalability are also carefully addressed in the proposed system. Instead of relying on expensive managed cloud services, the platform uses a self-hosted backend infrastructure running Supabase on Google Cloud Platform. The entire backend stack runs on a single virtual machine (e2-standard-2), which is capable of supporting more than 200 concurrent API users while maintaining stable performance. This approach provides a cost-effective solution while still delivering the scalability required for campus-wide adoption.

Finally, the system places strong emphasis on data security and controlled access. Since the platform includes multiple user roles and sensitive interactions, it implements Row-Level Security (RLS) across all database tables. These policies ensure that users can only access data that matches their role and permissions. With 37 tables protected by role-based policies, the system maintains strict control over project visibility, messaging access, and administrative operations.

In addition to these features, the platform integrates real-time communication tools such as messaging and notifications, enabling instant interaction between users. Real-time updates allow students to receive notifications when someone engages with their project, while messaging features support collaboration and mentorship discussions. Features such as typing indicators and read receipts further enhance communication by making interactions more dynamic and responsive. Through these solutions, the ACE Campus Innovation Hub effectively aligns each identified problem with a practical and scalable technological solution, ensuring strong problem–solution fit for the academic innovation ecosystem.

F. Architecture Design

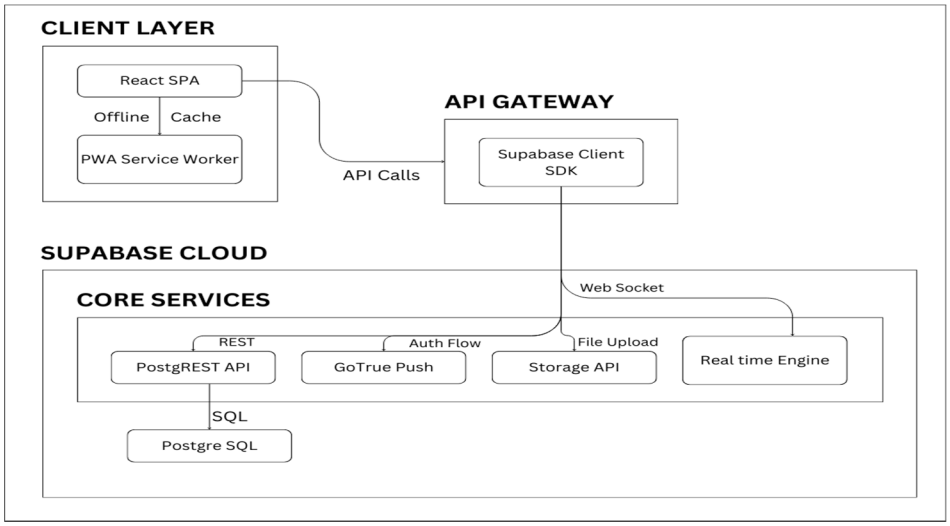


Fig. 3.6.1: System Architecture

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

The Campus Innovation Hub follows a three-tier architecture consisting of the presentation layer, application layer, and data layer. This architectural approach ensures modular development, scalability, and efficient separation of concerns between the user interface, backend services, and database operations.

The presentation layer (frontend) is implemented as a Single Page Application (SPA) using React 18 with TypeScript, enabling a modern, type-safe, and component-driven user interface. The platform includes more than 30 lazy-loaded page components, allowing the application to load only the required modules when needed, improving performance and reducing initial load time. The UI design is built using the shadcn/ui component library, which provides over 30 reusable and accessible interface primitives. Styling and layout are handled through Tailwind CSS, enabling a responsive mobile-first design that adapts seamlessly across smartphones, tablets, and desktops. The application is also configured as a Progressive Web Application (PWA) with a service worker that enables offline caching, faster page loads, and installability across devices.

The application layer (backend services) is powered by the Supabase ecosystem, which provides several microservices that collectively handle authentication, APIs, real-time communication, and media storage. User authentication is managed through GoTrue, which supports JWT-based authentication, email/password login, and OAuth integration such as Google OAuth. RESTful APIs are automatically generated from the database schema using PostgREST, allowing efficient data access without the need for manually written API endpoints. Real-time functionality is handled by the Realtime service built with the Phoenix Framework, which broadcasts database changes through WebSocket connections for features such as notifications, live updates, and messaging. Media files such as images, videos, and documents are stored using an S3-compatible storage service, while Kong acts as the API gateway responsible for routing requests, enforcing rate limits, and managing cross-origin resource sharing (CORS). Additional supporting services include Imgproxy for real-time image optimization, Postgres Meta for database introspection used in administrative tools, and Supavisor, which functions as a connection pooler to efficiently manage PostgreSQL database connections.

The data layer (database tier) is built on PostgreSQL 15, which stores and manages all application data across 37 relational tables. To ensure strict access control and data security, Row-Level Security (RLS) policies are implemented on every table, allowing role-based data access according to user permissions. The system also includes several custom database functions that support advanced application logic, such as has_role() for role verification, award_xp_to_user() for gamification reward management, get_for_you_projects() for personalized content recommendations, search_hashtags() and get_trending_hashtags() for discovery features, and get_top_colleges() for analytics. Database triggers are used to automate internal processes, including the handle_new_user() trigger that automatically creates a user profile when a new account is registered, along with additional triggers that update engagement counts and metrics. Furthermore, logical replication is enabled within the database to detect data changes and broadcast them to real-time services, enabling live updates across the platform without manual refresh.

G. Description Of Modules

The Campus Innovation Hub is designed as a modular system where different features of the platform are organized into separate functional modules. Each module performs a specific role within the platform, such as project discovery, social interaction, messaging, analytics, and administration. This modular architecture makes the platform easier to maintain, scale, and enhance in the future. The following sections describe the major modules of the system in simple terms.

3.7.1 Home Feed

The Home Feed is the main dashboard for logged-in users. It displays projects in different categories through four tabs: For You (personalized recommendations), Following (projects from users the person follows), Trending (projects with the highest engagement), and Your Department (projects from the user's department). The page also supports infinite scrolling so that more content loads automatically as the user scrolls. A sidebar shows trending hashtags, top colleges, and top contributors.

3.7.2 Project Showcase

This module displays the complete details of a specific project. It includes a gallery for images or videos, a full project description, and details such as the problem statement, solution, features, and real-world applications. Users can like, comment, save, or share the project. The page also shows the technology stack used, project tags, team members, and related projects. Sharing options are available for social platforms such as Twitter, LinkedIn, Facebook, and WhatsApp.

3.7.3 Project Upload

This module allows users to upload and publish their projects on the platform. It uses a multi-step form that guides users through the process. The steps include uploading images or videos, entering project details, selecting the technology stack and tags, adding team members, and choosing visibility settings. Users receive 50 XP points after successfully uploading a project.

3.7.4 User Profile

The User Profile module shows information about a user and their activities on the platform. Users can view their own profile or the public profile of others. The profile displays details such as avatar, role badge, department, XP score, and statistics like projects created, followers, and following count. It also includes multiple tabs where users can see their projects, saved content, liked projects, collaborations, and achievements.

3.7.5 Social Interaction System

This module handles the social features of the platform. Users can follow or unfollow others, like or save projects, and comment on project posts. The comment system also supports replies, allowing conversations to form under projects. Users can block others if necessary, and the system automatically updates engagement metrics.

3.7.6 Direct Messaging

The Direct Messaging module enables private communication between users. It provides a chat interface where users can send messages, share projects, and view conversation history. The system includes features such as message requests, typing indicators, online status, and read receipts. Permissions can also control who is allowed to send messages to a user.

3.7.7 Admin Dashboard

The Admin Dashboard is a control panel for system administrators. It includes tools for managing users, moderating content, reviewing reports, analyzing platform statistics, and configuring system settings. Administrators can also manage departments, badges, and platform-wide announcements.

IV. SYSTEM REQUIREMENTS

A. Hardware Requirement

- Server (Production): Google Cloud VM (e2-standard-2) with 2 vCPU, 8 GB RAM, and 50 GB SSD storage.
- Client Device: Any smartphone, tablet, or computer capable of running modern web browsers such as Chrome, Firefox, Safari, or Edge.
- Processor: Intel i5 / AMD Ryzen 5 or higher recommended for development.
- RAM: 8 GB minimum; 16 GB recommended for development and testing.
- Storage: At least 5 GB free space for project dependencies, build files, and development tools.
- Connectivity: Stable internet connection (minimum 1 Mbps) for accessing the web application and cloud services.

B. Software Requirement

- Operating System: Windows 10/11, macOS, or Linux
- Development Environment: Node.js, npm, Vite, and TypeScript for building and running the application.
- Frontend Technologies: React.js with Tailwind CSS for UI development, React Router for navigation.
- Backend & Database: PostgreSQL database with Supabase services
- Server & Deployment: Docker and Docker Compose for containerization, Nginx for reverse proxy and SSL configuration.

V. IMPLEMENTATION

A. Development Environment Setup

The system was developed using a modern full-stack web development environment consisting of a React-based frontend, Supabase backend, and cloud deployment on Google Cloud Platform (GCP).

1) Frontend Setup

The frontend application was developed using React, TypeScript, and Vite. Dependencies were installed using npm and environment variables were configured to connect the application to the backend services.

The Vite configuration includes:

- SWC compiler for faster builds
- PWA plugin with Workbox for offline support
- Custom path alias (@/) for simplified imports
- Development server running on port 8080
- The production build generates optimized static files for deployment on the cloud server.

2) Backend Setup

The backend infrastructure is built using Supabase deployed on Google Cloud Platform using Docker containers. Core services include:

- PostgreSQL – primary database
- Authentication service for login and OAuth
- Auto-generated REST APIs
- Real-time WebSocket server
- Object storage for media uploads

Container orchestration is handled using Docker Compose, allowing all services to be started and managed through a unified configuration.

B. System Architecture And Modules

The platform architecture consists of multiple integrated modules responsible for user management, social interactions, gamification, messaging, and media storage.

Database Design

The database is implemented using PostgreSQL and contains 37 relational tables grouped into several logical categories.

Core Entities

- profiles – user profile information
- projects – uploaded projects with engagement statistics such as likes, comments, and views

Social Interaction Tables

- Likes
- Comments
- Saves
- Follows
- Follow requests
- Blocked users

These tables support the social networking functionality of the platform.

Project Metadata

Additional tables store project-related information including:

- project media files
- project tags
- technology stacks
- project team members

Gamification System

The gamification module tracks user activity using tables such as:

- xp_transactions
- badges
- user_badges
- achievements

C. Entity, Storage, And Search

The developed platform integrates multiple functional modules that collectively support user authentication, project sharing, social networking, messaging, and performance optimization. These modules ensure that the system operates efficiently while providing an interactive environment where students, faculty members, and industry professionals can collaborate and showcase innovative projects. Each module is designed to handle a specific functionality while maintaining secure communication with the backend database and cloud infrastructure.

1) Authentication and Authorization: The platform implements a secure authentication mechanism using Supabase, supporting both email–password login and Google OAuth. After successful authentication, a JSON Web Token (JWT) is issued to maintain user sessions. The system follows a Role-Based Access Control (RBAC) model with defined roles such as Student, Faculty, HOD, Industry Expert, Principal, and Administrator. Each role has different permissions to control access to features such as project verification, administrative management, and user moderation.
2) XP and Gamification System: To encourage active participation, the platform includes a gamification engine where users earn Experience Points (XP) for performing activities such as liking, commenting, saving, or uploading projects. The XP value depends on the type of action and the role multiplier associated with the user. All XP activities are recorded in the database to maintain transparency and to support features like achievements, badges, and user rankings.
3) 5.3.3. Real-Time Communication: The system supports real-time updates using WebSocket technology. This ensures that interactions such as comments, likes, messages, and notifications appear instantly without refreshing the page. Real-time synchronization between the database and client application improves responsiveness and provides a seamless user experience.
4) Content Feed and Recommendation System: Projects are displayed through different feed categories including For You, Trending, Following, and Department. The “For You” section provides personalized project recommendations based on user interactions and interests, while the “Trending” feed highlights projects with the highest engagement. The “Following” feed displays content from followed users, and the “Department” feed focuses on projects created within the same academic department.

D. Software Description

1) React: A widely used JavaScript library for building modern and interactive user interfaces. It enables the development of dynamic single-page applications through reusable components and efficient state management. In this system, React is used to develop the frontend interface that allows users to browse projects, interact with other users, and manage their profiles.
2) TypeScript: A statically typed superset of JavaScript that improves code reliability and maintainability by introducing type checking. It helps developers detect errors during development and ensures better scalability for large applications such as this project platform.
3) Vite: A modern frontend build tool that provides extremely fast development server startup and optimized production builds. It improves development efficiency through features such as hot module replacement and faster compilation compared to traditional bundlers.
4) Supabase: A backend-as-a-service platform that provides essential backend functionality including authentication, database management, storage, and real-time services. It simplifies backend development by automatically generating APIs and handling secure user authentication.
5) PostgreSQL: A powerful open-source relational database management system used to store and manage structured application data. It supports advanced querying, indexing, and transaction management, ensuring reliable and efficient data storage for users, projects, interactions, and notifications.
6) Docker: A containerization platform that allows applications and their dependencies to run in isolated environments. It ensures consistency across development and production environments by packaging backend services into portable containers.
7) Nginx: A high-performance web server and reverse proxy used to route client requests to backend services. It also manages SSL encryption, improves application security, and enhances system performance through efficient request handling.
8) Tailwind CSS: A utility-first CSS framework used to design responsive and modern user interfaces. It provides pre-defined styling utilities that enable rapid UI development while maintaining consistency across the application.
9) TanStack Query: A data-fetching and state management library used to efficiently manage server data in React applications. It supports caching, background updates, and request deduplication, significantly improving application performance and responsiveness.

10) Playwright: An automated browser testing framework used to test the functionality and performance of web applications across multiple browsers. It ensures reliability by simulating user interactions and validating system behavior during testing.

11) Git: A distributed version control system used to track changes in source code during development. It enables collaboration among developers, supports version management, and helps maintain a stable and well-documented codebase throughout the project lifecycle.

E. Results

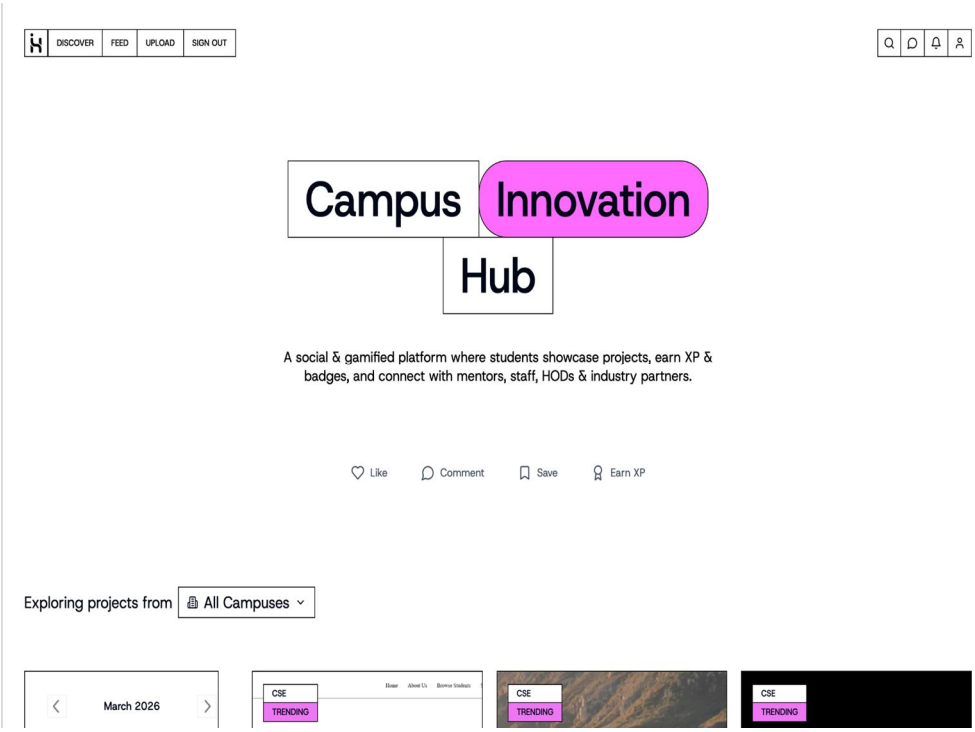


Fig. 5.5.1: Landing page

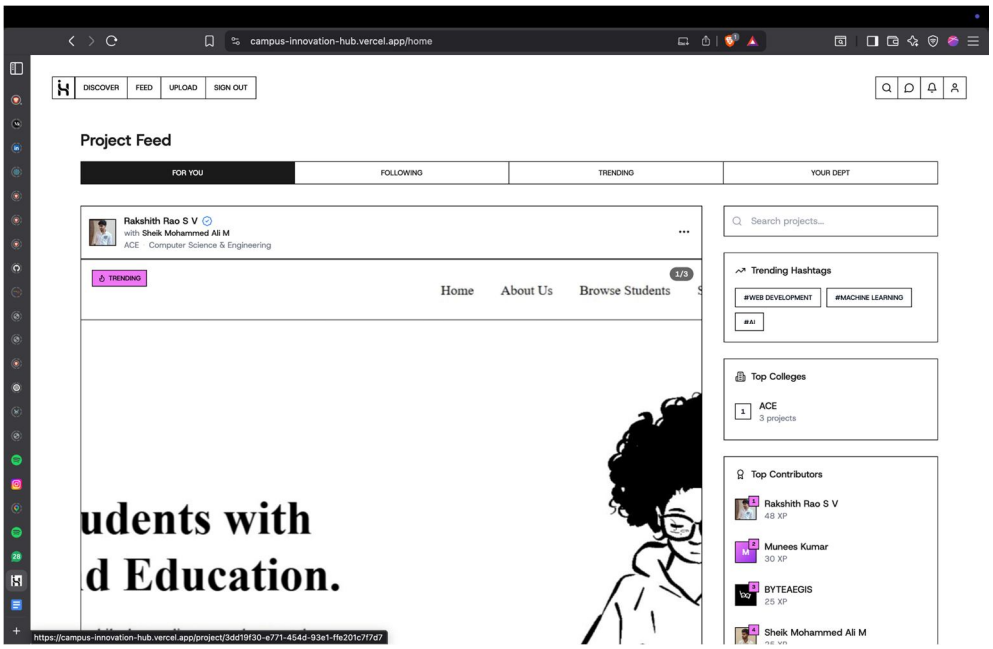


Fig. 5.5.2: Product Feed page

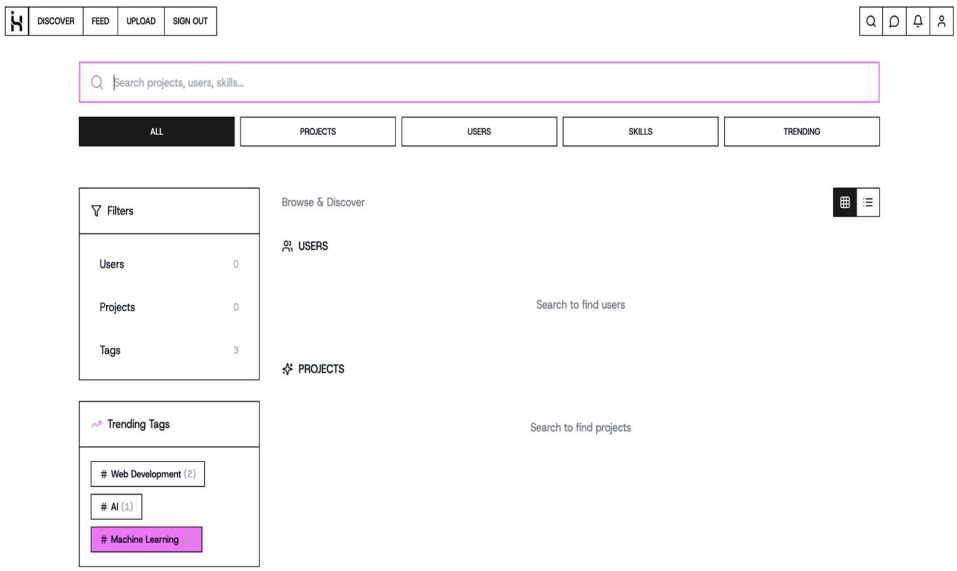


Fig. 5.5.3: Search Page

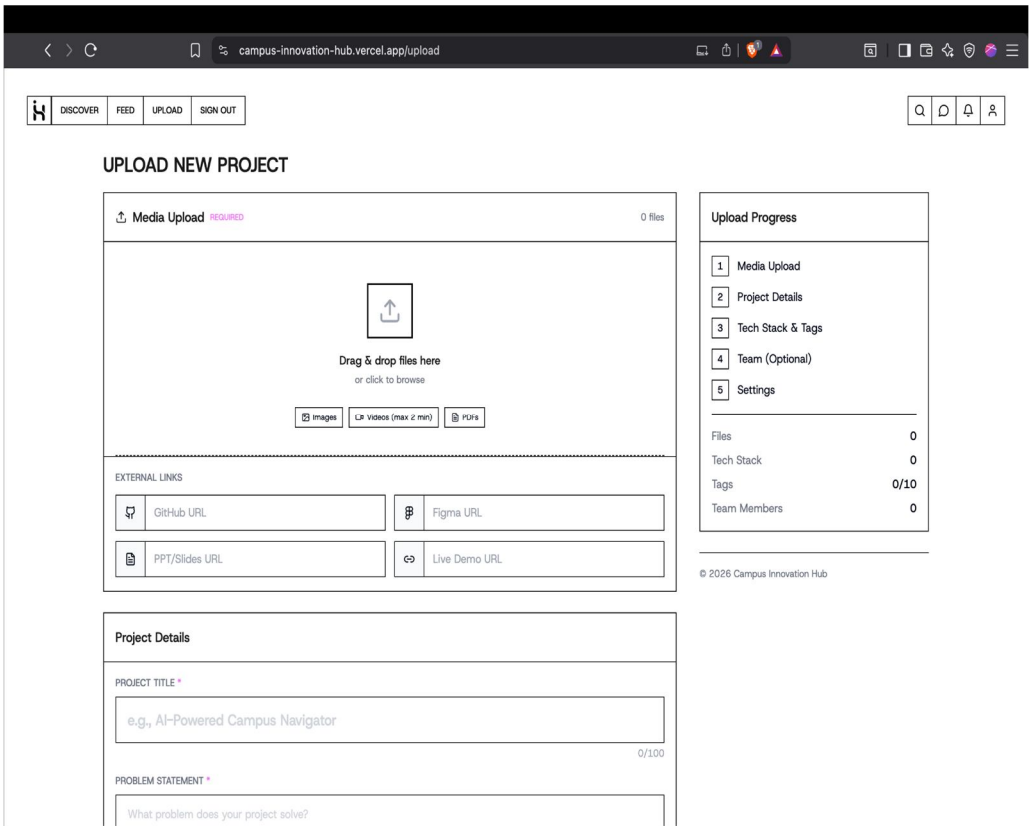


Fig. 5.5.4: Product upload page

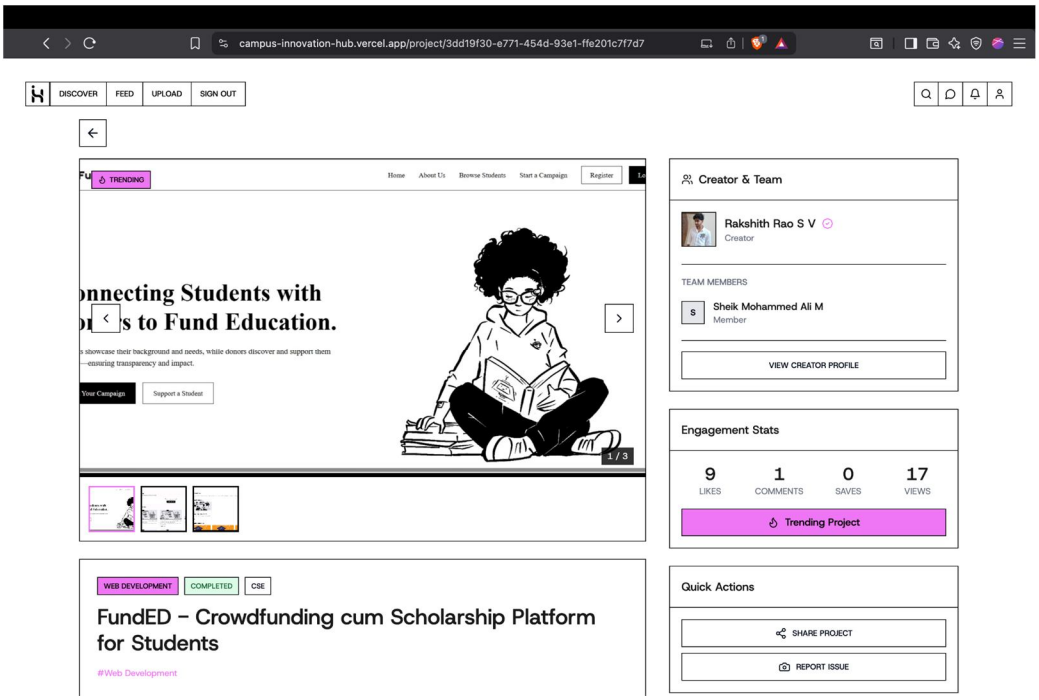


Fig. 5.5.5:Product detail page

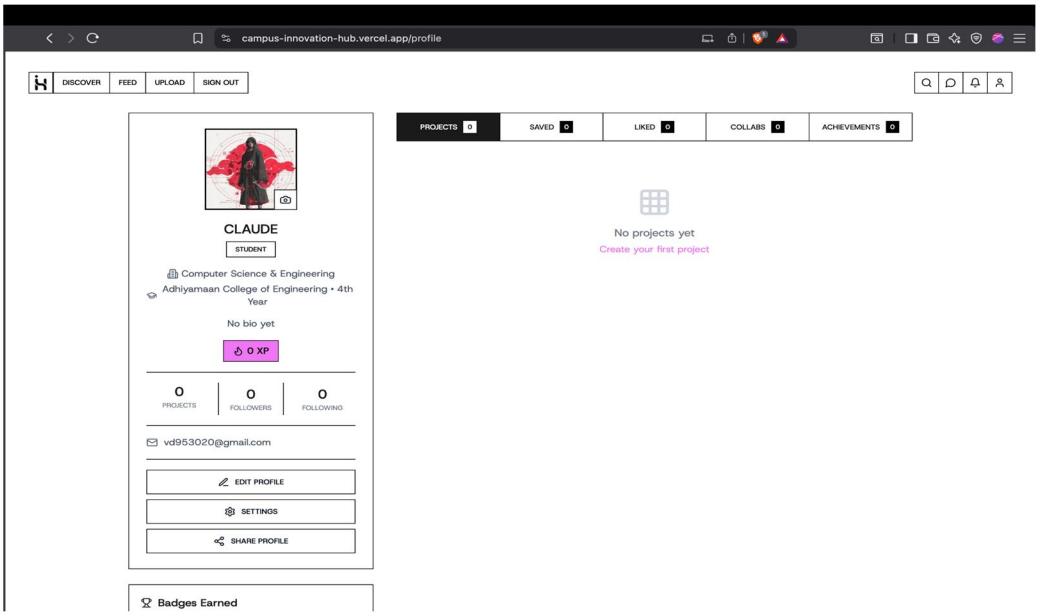


Fig. 5.5.6: Profile Settings

VI. CONCLUSION AND FUTURE ENHANCEMENT

A. Conclusion

The Campus Innovation Hub project has successfully established a centralized, gamified ecosystem for academic innovation, bridging the gap between student creativity and institutional recognition. By implementing a role-weighted XP gamification engine, the platform incentivizes high-quality project creation through multipliers that reward endorsements from faculty, HODs, and industry partners. This social-media-style network, delivered as a high-performance Progressive Web Application (PWA), ensures seamless accessibility across all devices with native-app features like offline support and push notifications.

The technical foundation of the platform leverages a self-hosted Supabase and PostgreSQL 15 backend deployed on Google Cloud Platform, providing a secure and scalable infrastructure.

With 37 interconnected database tables protected by granular Row-Level Security (RLS) policies, the system ensures data sovereignty and role-based access control. Real-time social features, including instant notifications and messaging powered by WebSockets, facilitate dynamic collaboration between students and stakeholders, transforming isolated projects into a vibrant community of innovation.

The overall project successfully validates the combined architectural approach of using a Progressive Web Application frontend with a self-hosted, scalable Supabase backend, proving it is a secure and cost-effective model for fostering an innovation-focused academic community.

B. Future Scope

The future development of the Campus Innovation Hub will prioritize the implementation of advanced recommendation algorithms to enhance project discovery. By utilizing machine learning models that analyze user interactions, skills, and departmental trends, the platform will deliver personalized content feeds, ensuring students and faculty are connected with the most relevant innovations and potential collaborators across the institution.

To further bridge the gap between academic research and commercial viability, future versions will introduce dedicated funding and sponsorship features. This module will allow industry partners and institutional departments to post "Innovation Challenges" or "Bounties," providing direct financial support or resources to high-potential projects. Integrated crowdfunding and grant application tools will further empower students to transition their prototypes into real-world startups.

Expanded gamification remains a core focus, with plans to introduce competitive inter-departmental leaderboards and a "Skill Badge" system. These badges, validated by faculty endorsements, will act as micro-credentials that students can showcase on their professional profiles. Enhanced engagement mechanics, such as XP-based voting for campus-wide "Project of the Month" awards, will continue to foster a culture of healthy competition and continuous improvement.

From an institutional perspective, the platform will evolve to support enterprise-grade scalability and official system integration. This includes the implementation of Single Sign-On (SSO) and direct synchronization with Student Information Systems (SIS) to automate profile creation and departmental verification. These integrations will ensure the platform becomes an official, seamless part of the institutional digital infrastructure, supporting thousands of concurrent users across multiple campuses.

Finally, dedicated community and collaboration tools will be introduced to support large-scale student organizations. Features such as "Innovation Circles," collaborative whiteboards, and integrated version control links will allow teams to manage complex project lifecycles within the hub. By providing a unified space for both showcasing and active development, the Campus Innovation Hub will mature into a comprehensive lifecycle management tool for student-led technological advancements.

APPENDICES

SOURCE CODE

App.tsx:

```
import React, { Suspense, lazy, useEffect, useState, useCallback } from 'react';
import { QueryClientProvider } from '@tanstack/react-query';
import { Toaster } from "@/components/ui/toaster";
import { Toaster as Sonner } from "@/components/ui/sonner";
import { TooltipProvider } from "@/components/ui/tooltip";
import { Routes, Route } from "react-router-dom";
import { AuthProvider, useAuth } from "@/contexts/AuthContext";
import { ErrorBoundary } from "@/components/common/ErrorBoundary";
import { ProtectedRoute } from "@/components/common/ProtectedRoute";
import { MaintenanceMode } from "@/components/common/MaintenanceMode";
import { PWAInstallButton } from "./components/PWAInstallButton";
import { SplashScreen } from "./components/SplashScreen";
import { Navbar } from "./components/Navbar";
import { HomeFeedPageSkeleton } from "./components/common/Skeletons";
import { OfflineIndicator } from "@/components/common/OfflineIndicator";
import {
  queryClient,
```

```
  initializeQueryPersistence,
  prefetchCommonData,
} from "@/lib/queryClient";

// Initialize query persistence on app load
initializeQueryPersistence();

// Global handler for unhandled promise rejections (prevents app crash on rapid navigation)
window.addEventListener('unhandledrejection', (event) => {
  const msg = String(event.reason?.message || event.reason || '');
  // Silently swallow abort errors and failed fetch during navigation
  if (msg.includes('AbortError') || msg.includes('aborted') || msg.includes('Failed to fetch') || msg.includes('signal')) {
    event.preventDefault();
  }
});

// Prefetch common sidebar data early
prefetchCommonData();

// ============================================
// LOADING COMPONENT - Skeleton based
// ============================================

const PageLoader = () => (
  <div className="min-h-screen bg-white dark:bg-gray-950">
    <Navbar />
    <HomeFeedPageSkeleton />
  </div>
);

// ============================================
// LAZY LOADED PAGES - Public Routes
// ============================================

const Discover = lazy(() => import("./pages/Discover"));
const HomeFeed = lazy(() => import("./pages/HomeFeed"));
const Search = lazy(() => import("./pages/Search"));
const ProjectDetails = lazy(() => import("./pages/ProjectDetails"));
const Leaderboards = lazy(() => import("./pages/Leaderboards"));
const ResetPassword = lazy(() => import("./pages/ResetPassword"));
const NotFound = lazy(() => import("./pages/NotFound"));

// ============================================
// LAZY LOADED PAGES - Authenticated Routes
// ============================================

const MyProjects = lazy(() => import("./pages/MyProjects"));
const UploadProject = lazy(() => import("./pages/UploadProject"));
const EditProject = lazy(() => import("./pages/EditProject"));
const Profile = lazy(() => import("./pages/Profile"));
```

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

```
const EditProfile = lazy(() => import("./pages/EditProfile"));
const UserProfile = lazy(() => import("./pages/UserProfile"));
const Followers = lazy(() => import("./pages/Followers"));
const Settings = lazy(() => import("./pages/Settings"));
const Notifications = lazy(() => import("./pages/Notifications"));
const Activity = lazy(() => import("./pages/Activity"));
const Messages = lazy(() => import("./pages/Messages"));
const Help = lazy(() => import("./pages/Help"));
const About = lazy(() => import("./pages/About"));
const PrivacyPolicy = lazy(() => import("./pages/PrivacyPolicy"));
const TermsOfService = lazy(() => import("./pages/TermsOfService"));
const CommunityGuidelines = lazy(() => import("./pages/CommunityGuidelines"));
const DepartmentStats = lazy(() => import("./pages/DepartmentStats"));
const ApplyBadge = lazy(() => import("./pages/ApplyBadge"));

// ========================================
// LAZY LOADED PAGES - Admin Routes
// ========================================

const AdminLayout = lazy(() => import("./pages/admin/AdminLayout"));
const AdminDashboard = lazy(() => import("./pages/admin/AdminDashboard"));
const AdminUsers = lazy(() => import("./pages/admin/AdminUsers"));
const AdminContent = lazy(() => import("./pages/admin/AdminContent"));
const AdminReports = lazy(() => import("./pages/admin/AdminReports"));
const AdminAnalytics = lazy(() => import("./pages/admin/AdminAnalytics"));
const AdminDepartments = lazy(() => import("./pages/admin/AdminDepartments"));
const AdminNotifications = lazy(() => import("./pages/admin/AdminNotifications"));
const AdminSettings = lazy(() => import("./pages/admin/AdminSettings"));
const AdminBadges = lazy(() => import("./pages/admin/AdminBadges"));
const AdminSupport = lazy(() => import("./pages/admin/AdminSupport"));

// ========================================
// BANNED/SUSPENDED SCREEN
// ========================================

const BannedScreen = lazy(() => import("./components/common/BannedScreen"));

// ========================================
// APP COMPONENT
// ========================================

const App = () => {
  const [showSplash, setShowSplash] = useState(true);

  const handleSplashComplete = useCallback(() => {
    setShowSplash(false);
  }, []);

  return (
    <ErrorBoundary>
```



```
    <QueryClientProvider client={queryClient}>
     <OfflineIndicator />
     <AuthProvider>
      <MaintenanceMode>
       <TooltipProvider>
        <Toaster />
        <Sonner />
        {showSplash && <SplashScreen onComplete={handleSplashComplete} />}
        <PWAInstallButton variant="banner" />
        <Suspense fallback={<PageLoader />}>
         <AppRoutes />
        </Suspense>
       </TooltipProvider>
      </MaintenanceMode>
     </AuthProvider>
    </QueryClientProvider>
   </ErrorBoundary>
  );
};

// Inner component that can use useAuth (inside AuthProvider)
const AppRoutes = () => {
  const { bannedStatus, clearBannedStatus } = useAuth();

  if (bannedStatus) {
   return <BannedScreen status={bannedStatus} onDismiss={clearBannedStatus} />;
  }

  return (
   <>
   <Routes>
            {/* Public Routes */}
            <Route path="/" element={<Discover />} />
            <Route path="/home" element={<HomeFeed />} />
            <Route path="/search" element={<Search />} />
            <Route path="/project/:id" element={<ProjectDetails />} />
            <Route path="/leaderboards" element={<Leaderboards />} />
            <Route path="/reset-password" element={<ResetPassword />} />

            {/* Authenticated Routes */}
            <Route path="/project/:id/edit" element={<ProtectedRoute><EditProject /></ProtectedRoute>} />
            <Route path="/my-projects" element={<ProtectedRoute><MyProjects /></ProtectedRoute>} />
            <Route path="/upload" element={<ProtectedRoute><UploadProject /></ProtectedRoute>} />
            <Route path="/profile" element={<ProtectedRoute><Profile /></ProtectedRoute>} />
            <Route path="/profile/edit" element={<ProtectedRoute><EditProfile /></ProtectedRoute>} />
            <Route path="/user/:userId" element={<UserProfile />} />
            <Route path="/user/:userId/followers" element={<Followers />} />
            <Route path="/followers" element={<ProtectedRoute><Followers /></ProtectedRoute>} />
            <Route path="/settings" element={<ProtectedRoute><Settings /></ProtectedRoute>} />
            <Route path="/settings/help" element={<ProtectedRoute><Help /></ProtectedRoute>} />
```

```
            <Route path="/settings/about" element={<ProtectedRoute><About /></ProtectedRoute>} />
            <Route path="/privacy" element={<PrivacyPolicy />} />
            <Route path="/terms" element={<TermsOfService />} />
            <Route path="/community-guidelines" element={<ProtectedRoute><CommunityGuidelines /></ProtectedRoute>} />
            <Route path="/notifications" element={<ProtectedRoute><Notifications /></ProtectedRoute>} />
            <Route path="/activity" element={<ProtectedRoute><Activity /></ProtectedRoute>} />
            <Route path="/messages" element={<ProtectedRoute><Messages /></ProtectedRoute>} />
            <Route path="/department-stats" element={<ProtectedRoute><DepartmentStats /></ProtectedRoute>} />
            <Route path="/apply-badge" element={<ProtectedRoute><ApplyBadge /></ProtectedRoute>} />

            {/* Admin Routes */}
            <Route path="/admin" element={<AdminLayout />}>
              <Route index element={<AdminDashboard />} />
              <Route path="users" element={<AdminUsers />} />
              <Route path="content" element={<AdminContent />} />
              <Route path="reports" element={<AdminReports />} />
              <Route path="analytics" element={<AdminAnalytics />} />
              <Route path="departments" element={<AdminDepartments />} />
              <Route path="badges" element={<AdminBadges />} />
              <Route path="support" element={<AdminSupport />} />
              <Route path="notifications" element={<AdminNotifications />} />
              <Route path="settings" element={<AdminSettings />} />
            </Route>

            {/* Catch-all Route */}
            <Route path="*" element={<NotFound />} />
          </Routes>
    </>
  );
};

export default App;
```

VII. ACKNOWLEDGEMENT

It is one of the most efficient tasks in life to choose the appropriate words to express one's gratitude to the beneficiaries. We are very much grateful to God who helped us all the way through the project and how molded us into what we are today.

We are grateful to our beloved Principal Dr. R. RADHAKRISHNAN, M.E., Ph.D., Adhiyamaan College of Engineering (An Autonomous Institution), Hosur for providing the opportunity to do this work in premises.

We acknowledge our heartfelt gratitude to Dr. G. FATHIMA, M.E., Ph.D., Professor and Head of the Department, Department of Computer Science and Engineering, Adhiyamaan College of Engineering (An Autonomous Institution), Hosur, for her guidance and valuable suggestions and encouragement throughout this project and made us to complete this project successfully.

We are highly indebted to Mrs. D.M. VIJAYALAKSHMI, M.E., (Ph.D.,) Supervisor, Assistant Professor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering (An Autonomous Institution), Hosur, whose immense support, encouragement and valuable guidance were responsible for completing the project successfully.

We also extend our thanks to the Project Coordinator and all Staff Members for their support in completing this project successfully.

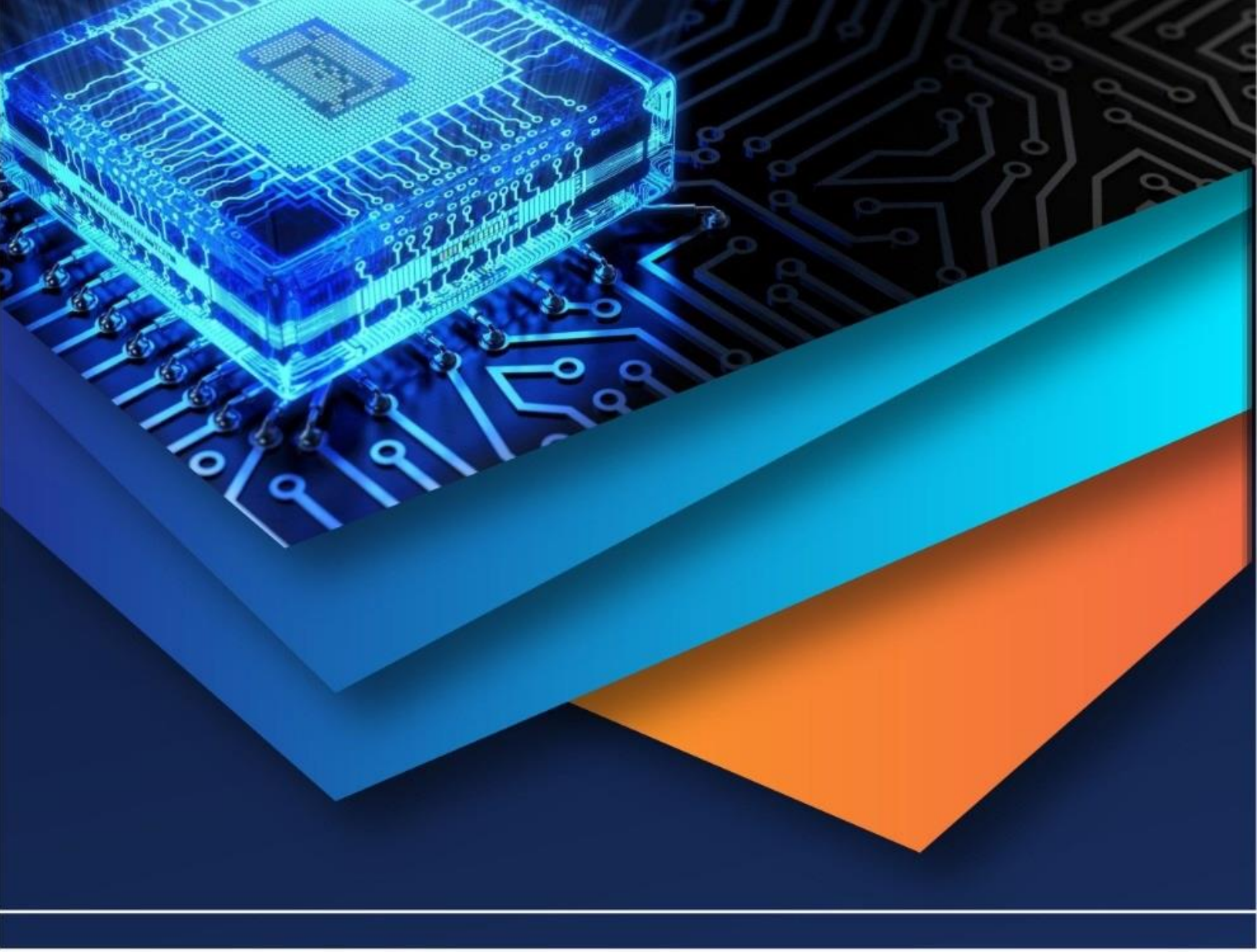
Finally, we would like to thank our parents, without their motivation and support it would not have been possible for us to complete this project successfully.

REFERENCES

[1] Chen, A., & Lee, B. (2024). "Designing Role-Weighted Gamification Systems for Educational Platforms." Journal of Learning Technologies.

[2] David, C., & Foster, E. (2023). "Scalable Real-Time Backend Architecture using PostgreSQL and Phoenix Framework (Supabase)." IEEE Transactions on Cloud Computing, 11(4), 501–518.

[3] Garcia, F., & Holmes, G. (2022). "Progressive Web Applications in Academia: Bridging the Gap between Mobile and Desktop." Educational Technology & Society, 25(3), 1–15.
[4] Iyer, H., & King, J. (2024). "Securing Multi-Role Access Control in PostgreSQL using Row-Level Security." ACM Symposium on Database Security, 32(1), 101–115.
[5] Lopez, K., & Miller, L. (2023). "Effective Project Discovery through Personalized Recommendation Systems in Educational Ecosystems." International Conference on AI in Education, 45–60.
[6] Nguyen, M., & O'Connell, N. (2024). "XP-Based Incentives: Analyzing Student Engagement in Extracurricular Digital Hubs." Computers & Education, 198, 104789.
[7] Quinn, P., & Smith, R. (2023). "Bridging Academia and Industry: Digital Platforms for Talent Scouting and Mentorship." Higher Education Review, 40(2), 155–170.
[8] Thompson, S., & Valdez, T. (2022). "Cost-Optimized Cloud Deployment Strategies for Social Web Applications using Docker on GCP." Journal of Software Engineering and Practice, 14(3), 201–215.
[9] Warren, U., & Xander, V. (2023). "A Comparative Study of Real-Time Notification Technologies: WebSockets vs. Polling." International Journal of Web Technology, 16(1), 50–65.
[10] Yang, W., & Zimmerman, Y. (2024). "Tailwind CSS and shadcn/ui: A Modern Approach to Accessible and Responsive Web Design." Frontiers in HCI, 7, 128904.
[11] Chen, L., & Davis, M. (2025). "Optimizing React Application Performance with TanStack Query for Server State Management." ACM Transactions on Web Systems, 19(2), 33-48.
[12] Evans, N., & Fisher, O. (2024). "The Role of PostgREST in Decoupled Data Architectures and API Automation." Journal of Database Technology, 15(1), 89-105.
[13] Green, Q., & Harris, R. (2023). "Containerizing Open Source Backend-as-a-Service Solutions (BaaS) for Cost-Effective Cloud Deployment." International Journal of Cloud Computing, 12(3), 211–225.
[14] Iqbal, S., & Jones, T. (2024). "Vite and SWC: Enhancing Development Velocity for Large-Scale Progressive Web Applications." IEEE Software Engineering Letters, 5(4), 180–195.
[15] Keller, U., & Lewis, V. (2025). "Leveraging PostgreSQL Triggers and RPCs for Real-Time Event Processing and Gamification Logic." Database Management Systems Quarterly, 28(1), 7-22.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY