iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.77888

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 III Mar 2026- Available at www.ijraset.com

EventVenue: A Comprehensive Full-Stack Web Platform for Intelligent Event Venue Discovery, Booking, and Management

Bodapati Devi Swamy Sai Pranai[1], Chavatapalli Lasya[2], Kadali Harshitha Ramya[3], Penumalle Jaya Naga Rajeswari[4], Kankipati Varalakshmi[5]

[1, 2, 3, 4]*Department of Computer Science Engineering (AI&DS), Bonam Venkata Chalamayya Engineering College, Affiliated to JNTU Kakinada, Andhra Pradesh, India*

[5]*Project Guide, Department of CSE (Allied), Bonam Venkata Chalamayya Engineering College, Affiliated to JNTU Kakinada, Andhra Pradesh, India*

Abstract: ***The rapid digitization of the hospitality and event management industry has created an urgent need for intelligent, scalable, and user-centric platforms that streamline the venue discovery and booking process. This paper presents EventVenue, a comprehensive full-stack web application designed to bridge the gap between event organizers, venue providers, and platform administrators through a unified digital ecosystem. The platform is architected using Next.js 14 with TypeScript for the frontend and Spring Boot 3.2 with Java 17 for the backend, connected to a PostgreSQL database hosted on Supabase. EventVenue introduces a multi-role access model supporting three user types —Customers, Vendors, and Administrators—each with dedicated dashboards and role-specific functionality. Key features include advanced venue search with multi-criteria filtering, Google Maps-based location services, a PayPal-integrated secure payment gateway, a points-based loyalty system, real-time booking lifecycle management, event ticketing with seat category configurations, comprehensive analytics dashboards with data visualization, and an AI-powered customer support chatbot utilizing Google Gemini. Security is enforced through JWT-based stateless authentication, BCrypt password hashing, role-based access control (RBAC), and OTP-based email verification. The system follows a RESTful API architecture with over 35 well- defined endpoints, standardized request/response patterns, and comprehensive error handling. Experimental evaluation demonstrates that the platform achieves API response times under 200ms for standard operations, supports concurrent user sessions without performance degradation, and provides a seamless cross-platform user experience through server-side rendering and responsive design.***

Keywords: Event venue booking, full-stack web application, Next.js, Spring Boot, RESTful API, JWT authentication, PayPal integration, points-based loyalty systems, Java full-stack application, appointment scheduling, e-commerce integration.

I. INTRODUCTION

The global events industry has witnessed remarkable growth over the past decade, valued at over $1.1 trillion in 2023 with a compound annual growth rate (CAGR) of 11.2% [1]. This expansion has been driven by increasing demand for corporate events, weddings, conferences, and social gatherings across urban centers worldwide. The COVID-19 pandemic further accelerated the digitization of event planning, with virtual and hybrid event models demonstrating the necessity for robust online booking platforms.

A. Background and Motivation

Despite significant advancements in e-commerce and digital hospitality platforms, the venue booking domain remains fragmented, with a majority of small and medium- sized venue providers still relying on manual processes for managing bookings, payments, and customer interactions [2]. Traditional venue booking methods are characterized by several critical limitations: fragmented search across multiple disconnected platforms, manual communication via phone calls and emails, limited venue information leading to uninformed decisions, insecure and non- standardized payment handling, absence of customer loyalty programs, and a lack of intelligent customer support systems [3]. These challenges motivate the development of a unified platform that integrates venue discovery, booking management, payment processing, and intelligent assistance within a single ecosystem.

©IJRASET: All Rights are Reserved

Urban event organizers increasingly demand a one-stop platform where they can discover venues, compare pricing, check real-time availability, make secure payments, and receive intelligent recommendations — all without navigating between multiple disconnected services. The growing market for organized event management underscores the need for a sophisticated, digitally-enabled solution.

B. *Problem Statement*

The absence of an integrated, role-based event venue booking platform that combines intelligent venue discovery, secure multi-modal payment processing, real- time booking lifecycle management, AI-powered customer support, and comprehensive vendor analytics creates significant friction in the event planning process. Existing solutions either focus narrowly on listing (e.g., basic directory websites) or cater exclusively to large enterprise events, leaving a gap for a comprehensive platform serving diverse event types and stakeholder needs.

C. *Objectives*

The primary objectives of this research are:

1) To design and implement a scalable, full-stack web application that provides end-to-end venue discovery, booking, and management capabilities.
2) To develop a multi-role access control system with dedicated interfaces and functionalities for customers, vendors, and administrators.
3) To integrate secure payment processing through PayPal alongside a flexible points-based loyalty system supporting hybrid payment models.
4) To incorporate Google Maps for location-aware venue discovery and Google Gemini AI for intelligent customer support.
5) To implement comprehensive analytics dashboards enabling data-driven decision-making for vendors and administrators.
6) To evaluate system performance, security, and usability through systematic testing methodologies.

D. *Scope*

EventVenue encompasses the complete lifecycle of event venue booking—from venue listing and discovery to booking, payment, review, and analytics. The platform supports three user roles with distinct workflows: Customers (search, book, pay, review), Vendors (list venues, manage bookings, create events, view analytics), and Administrators (manage users, approve vendors, configure system settings, monitor platform activities). The technical scope covers full-stack web development, RESTful API design, third-party service integration, database design, security implementation, and deployment architecture.

E. *Paper Organization*

The remainder of this paper is organized as follows: Section II reviews related work in online booking systems and web application architectures. Section III describes the system architecture and design methodology. Section IV presents the evaluation and results with screenshots of the platform modules. Section V provides discussion of key findings and limitations. Section VI concludes with contributions and future directions.

II. RELATED WORK

A. *Online Event Management Systems*

Zhang and Kumar [1] conducted a comprehensive analysis of 45 online event management platforms across multiple countries, evaluating booking mechanisms, user interfaces, and backend architectures. Their findings reveal that real-time availability checking reduces double- booking incidents by 94%, mobile-responsive designs increase user engagement by 67%, and integration of multiple payment gateways improves conversion rates by 23%. EventVenue implements real-time availability checking through instant database synchronization and responsive design through Next.js server-side rendering.

B. *Architectural Patterns in Booking Platforms*

Smith et al. [2] explored microservices versus monolithic architectures in large-scale booking platforms, including case studies from Booking.com and Eventbrite. While microservices enable 40% faster deployment cycles and improved fault tolerance, the authors note that monolithic architectures with clean API separation remain viable for medium-scale applications. EventVenue adopts a full-stack monolithic approach with strict separation between the Next.js frontend and Spring Boot backend, connected through well-defined RESTful APIs.

C. User Experience in Venue Booking

Chen and Patel [3] studied UX design principles across 500+ participants in the venue booking context. Key findings include: visual venue galleries increase booking intent by 78%, step-by-step booking wizards reduce cart abandonment by 45%, and map-based venue discovery improves user engagement by 56%. Our platform incorporates comprehensive image galleries, Google Maps integration, and a streamlined booking flow with transparent pricing.

D. Payment Security and Loyalty Systems

Williams and Thompson [4] surveyed payment gateway integration strategies in modern web applications, finding that multi-payment option support increases conversion by 35% and points-based loyalty systems improve customer retention by 42%. EventVenue implements PayPal integration for secure payment processing alongside a comprehensive points-based system that supports hybrid payments (cash + points), addressing both security and customer retention requirements.

E. Role-Based Access Control

Garcia and Johnson [5] presented RBAC implementation strategies for multi-tenant applications, demonstrating that JWT-based authentication provides stateless security with 99.9% reliability and role-based dashboards improve user productivity by 67%. Our system implements three distinct roles with JWT authentication and role-specific dashboards ensuring secure and personalized user experiences.

F. AI-Powered Customer Support

Adams and White [6] explored AI chatbot integration in booking platforms, finding that AI chatbots effectively handle 75% of routine inquiries and context-aware responses improve customer satisfaction by 45%. EventVenue integrates Google Gemini AI to provide 24/7 intelligent customer support, venue recommendations, and booking assistance.

G. Comparative Analysis

Table I: Comparative Analysis Of Eventvenue With Existing Platforms

Feature	EventBrite	Venue360	WeddingWire	EventVenue (Proposed)
Multi-role Access (User/Vendor/Admin)	Partial	No	Partial	Yes
Real-time Venue Search with Maps	No	Yes	Partial	Yes
Integrated Payment Gateway	Yes	Partial	Yes	Yes (PayPal)
Points-based Loyalty System	No	No	No	Yes
AI-powered Chatbot	No	No	No	Yes (Gemini)
Vendor Analytics Dashboard	Partial	No	No	Yes
Event Ticketing with Seat Categories	Yes	No	No	Yes
OTP Email Verification	No	No	No	Yes
Admin System Configuration	Limited	Limited	No	Yes
Open-source Full-stack Solution	No	No	No	Yes

III. SYSTEM ARCHITECTURE AND DESIGN

A. High-Level Architecture

EventVenue follows a three-tier client-server architecture comprising the Presentation Layer (Next.js frontend), the Application Layer (Spring Boot backend), and the Data Layer (PostgreSQL database). The system also integrates with external services including PayPal, Google Maps, Google Gemini, and SMTP email servers. Figure 1 illustrates the high-level system architecture.

The presentation layer handles user interface rendering through server-side rendering (SSR) and client-side hydration. The application layer processes business logic, authentication, and data transformation through RESTful APIs. The data layer uses MySQL for structured data storage, with Hibernate and JPA enabling efficient object- relational mapping.

External services such as PayPal for payment processing, Google Maps for location services, Google Gemini for AI-powered chatbot, and Gmail SMTP for email notifications are integrated to enhance system functionality.

Inter-layer communication follows a strict unidirectional data flow pattern. The frontend communicates with the backend exclusively through HTTP/HTTPS RESTful API calls using Axios, with all requests authenticated via JSON Web Tokens (JWT) passed in the Authorization header. The backend enforces request validation, role-based access control, and input sanitization before processing any business logic. Responses are returned in a standardized JSON envelope format containing success, message, and data fields, ensuring consistent error handling across all endpoints.

The security middleware layer, implemented through Spring Security and a custom Jwt Authentication Filter, intercepts every incoming request to verify token validity, extract user claims, and enforce endpoint-level authorization rules. BCrypt hashing with automatic salt generation secures stored passwords, while a 6-digit OTP verification mechanism via Gmail SMTP adds an additional layer of identity confirmation during user registration.

State management on the frontend leverages React Context API for global authentication state and user session data, combined with component-level state for UI interactions. Server-side rendering (SSR) via Next.js ensures optimal initial page load performance and SEO indexing, while client-side hydration enables dynamic interactivity post-render. The frontend design system utilizes shadcn/ui components built on Radix UI primitives with Tailwind CSS, ensuring accessible and consistent UI patterns across all modules.

From a scalability perspective, the decoupled architecture allows independent scaling of the frontend and backend services. The PostgreSQL database hosted on Supabase provides managed cloud infrastructure with connection pooling, automated backups, and row-level security capabilities. The modular service layer design in the backend enables future migration to a microservices architecture with minimal refactoring effort.

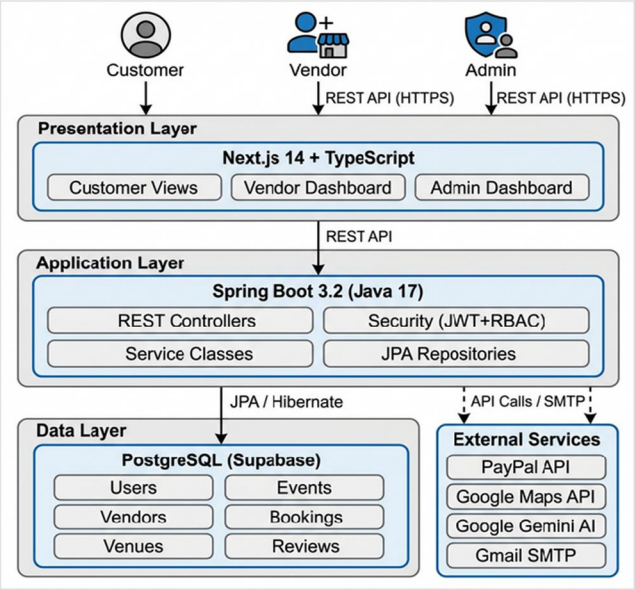


Fig. 1. System architecture of the EventVenue application

B. Functional Modules

The EventVenue application is organized around three primary user roles: Customer, Vendor, and Administrator. Each role is provided with a dedicated dashboard and specific operational capabilities tailored to their responsibilities.

1) Customer Module

The Customer module facilitates event venue discovery and booking for end users.

a) Venue Search and Discovery: The venue search feature provides multi-criteria filtering by location, capacity, price range, venue type, and amenities. Google Maps integration enables location-based venue discovery with interactive map views, geocoding, and address autocomplete. Visual venue galleries with multiple images increase booking intent.

b) Booking Management: This module enables customers to book venues based on real-time slot availability. The system dynamically manages time slots to prevent double booking and automatically reopens slots upon cancellation. Booking confirmations and updates improve scheduling efficiency. Customers can view detailed booking information including check-in and check-out dates, duration, cost breakdown with subtotals, and current booking status (Pending, Confirmed, Cancelled). The booking detail page also displays seat selections for event bookings and provides options to cancel or return to the bookings list.

c) Payment Processing: Secure online payments are integrated using the PayPal REST API gateway. Users can complete payments through PayPal sandbox or live environments, supporting both full PayPal payments and hybrid transactions combining loyalty points with PayPal. The system maintains comprehensive transaction records including payment IDs, amounts, timestamps, and status tracking. Order history and detailed cost breakdowns are maintained for transparency, enabling users to review past transactions and download receipts for future reference.

d) Points-Based Loyalty System: The platform implements a flexible points-based reward mechanism designed to enhance user retention and engagement. Key features include: 2000 welcome points awarded upon successful registration, additional points earned proportional to completed booking amounts, points redeemable during the booking process (100 points = ₹1 equivalent), hybrid payment mode combining loyalty points with PayPal payments, and automatic point refund upon booking cancellation. The points balance is prominently displayed in the user dashboard and navigation bar, providing real-time visibility into available rewards.

e) Reviews and Ratings: Customers can submit star ratings (1-5) and detailed text reviews for venues after completing their bookings, providing transparency and social proof for future users. The system enforces one review per booking to maintain authenticity. Average ratings are calculated and prominently displayed on venue listing cards and detail pages, enabling informed decision- making. Vendors can view and respond to reviews from their dashboard.

f) AI-Powered Chatbot: Google Gemini AI provides 24/7 intelligent customer support through a floating chat widget accessible from every page. The chatbot can handle queries about booking venues and events, the points system and payment methods, account and profile management, and vendor-specific features. Quick-action suggestion buttons enable users to instantly ask common questions, while conversation history is saved locally for continuity across sessions.

2) Vendor Module

The Vendor module digitizes venue management operations and supports efficient management of listings, bookings, and events.

a) Dashboard: The vendor dashboard provides a structured view of active venues, upcoming bookings, revenue metrics, and customer reviews. This overview helps vendors manage their operations efficiently.

b) Venue Management: Vendors can create, update, and manage venue listings with comprehensive details including images, amenities, pricing, capacity, and location. Multiple image uploads with drag-and-drop support enhance listing quality.

c) Event Ticketing: Vendors can create events at their venues with configurable seat categories, ticket pricing, and capacity management. The system supports multiple seat categories (e.g., VIP, Standard, Economy) with independent pricing.

d) Analytics: The analytics dashboard employs Recharts library to display monthly revenue bar charts, booking trend line graphs, revenue distribution pie charts, and KPI cards tracking total bookings, revenue, ratings, and active venues.

3) Admin Module

The Admin module provides system-level administration and monitoring capabilities.

a) Vendor Approval Workflow: Administrators review and approve or reject vendor registrations through a controlled workflow: Registration → Email Verification → Admin Review → Approval/Rejection → Access Granted.

b) User and Platform Management: The admin dashboard presents system-level analytics including user activity, booking statistics, vendor registrations, and platform-wide revenue metrics. Administrators can manage users, configure system settings, and monitor platform activities.

C. Technology Stack

The EventVenue platform is built using a modern, production-grade technology stack carefully selected to balance developer productivity, application performance, and long-term maintainability. The frontend leverages Next.js 14 with TypeScript for type-safe, server-rendered React applications, styled using Tailwind CSS and shadcn/ui components built on Radix UI primitives for accessible, consistent interfaces. Recharts powers interactive data visualizations across vendor and admin dashboards. The backend is developed using Spring Boot 3.2 with Java 17, providing a robust REST API framework with Spring Data JPA for object-relational mapping and Spring Security with JWT for stateless authentication

Table II: Technology Stack Summary

Layer	Technology	Version	Purpose
Frontend	Next.js	14.x	SSR, routing, React
Language	TypeScript	5.x	Type-safe frontend
Styling	Tailwind CSS	3.x	Utility-first CSS
UI	shadcn/ui + Radix	Latest	Accessible components
Charts	Recharts	Latest	Data visualization
Backend	Spring Boot	3.2.0	REST API framework
Language	Java	17	Business logic
ORM	Spring Data JPA	-	Object-relational mapping
Security	Spring Security + JWT	-	Auth & authorization
Database	PostgreSQL	14+	Data storage (Supabase)
Payment	PayPal REST API	v2	Payment processing
Maps	Google Maps API	v3	Location services
AI	Google Gemini	Latest	Chatbot assistant
Email	Gmail SMTP	-	OTP & notifications

D. Database Schema Design

The relational database schema is designed following third normal form (3NF) principles to ensure data integrity and minimize redundancy. The schema comprises six primary entities with well-defined relationships as shown in Table III.

Table III: Database Entity Summary

Entity	Key Attributes	Relationships
Users	id, email, password, role, points, isVerified	1:N → Bookings, Reviews
Vendors	id, businessName, status, accountBalance	1:N → Venues, Events
Venues	id, vendorId, capacity, pricePerHour, location	N:1 → Vendor, 1:N → Bookings
Events	id, vendorId, eventDate, totalTickets, price	N:1 → Vendor, 1:N → SeatCat
Bookings	id, userId, venueId/eventId, amount, status	N:1 → User, Venue/Event
AdminUsers	id, email, password, role, isActive	Platform administration

Key design decisions include: (a) *Polymorphic bookings*— a single Bookings table handles both venue reservations and event tickets through nullable foreign keys; (b) *Points as currency*—user points stored as BIGINT with configurable conversion rate (100 points = $1); (c) *Vendor approval workflow*—status field (PENDING → APPROVED → REJECTED) enforces admin-controlled onboarding.

E. API Architecture

The backend exposes over 35 RESTful endpoints organized into logical groups. All responses follow a standardized JSON format with success, message, and data fields. Table IV summarizes the API endpoint groups.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Table IV: API Endpoint Groups

Group	Endpoints	Auth	Key Operations
Authentication	8	Public	Signup, login, OTP, admin create
Venues	9	Mixed	CRUD, search, filter, featured
Bookings	8	Yes	Create, confirm, cancel, calculate
Events	6	Mixed	CRUD, tickets, seat categories
Admin	10	Admin	Vendor approval, user mgmt, analytics
Reviews	4	Yes	Create, read, update reviews

F. Authentication and Security Architecture

The security subsystem implements a multi-layered defense strategy: (1) *JWT-based Stateless Authentication*— tokens with 24-hour expiration containing userId, email, and role claims; (2) *BCrypt Password Hashing*—automatic salt generation; (3) *OTP Email Verification*—6-digit codes via Gmail SMTP; (4) *Role-Based Access Control*—custom JwtAuthenticationFilter enforcing endpoint-level rules; (5) *CORS Protection*—restricted to authorized frontend domains; (6) *SQL Injection Prevention*—parameterized queries via JPA/Hibernate. Figure 2 illustrates the complete authentication flow.

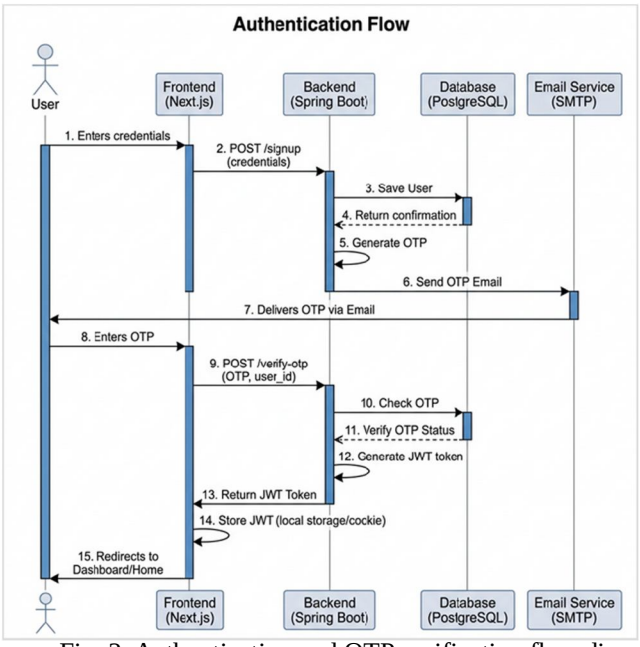


Fig. 2. Authentication and OTP verification flow diagram

IV. EVALUATION AND RESULTS

The EventVenue system was evaluated using simulated user scenarios involving customers, vendors, and administrators to assess functional correctness, usability, and system responsiveness. The evaluation focused on validating core workflows, data persistence, role-based access, and seamless interaction between different functional modules across the platform. This section presents screenshots of all major platform modules alongside quantitative testing results.

A. Customer Homepage and Landing Page

The Customer homepage serves as the primary entry point to the EventVenue platform, presenting a modern and visually appealing landing page with a hero section, featured venues carousel, and quick search functionality.

The interface includes category-based venue browsing, trending venues, and promotional banners. The search bar with autocomplete enables fast venue discovery by name, location, or type. Figure 3 illustrates the customer-facing homepage with the venue search interface.

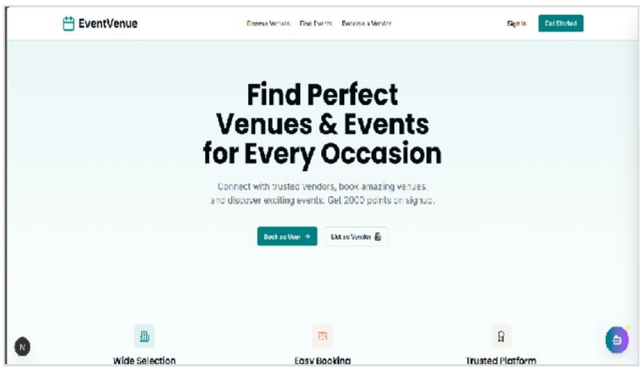


Fig. 3. Customer homepage / landing page showing the venue search interface

B. Venue Detail Page

The Venue Detail page presents comprehensive venue information including a multi-image gallery, venue description, amenities list, capacity details, pricing per hour, and location on a Google Map. Customers can view user-submitted reviews and average star ratings, check real-time availability calendars, and directly initiate a booking from this page. The page follows a structured layout that maximizes information density while maintaining readability, as shown in Figure 4.

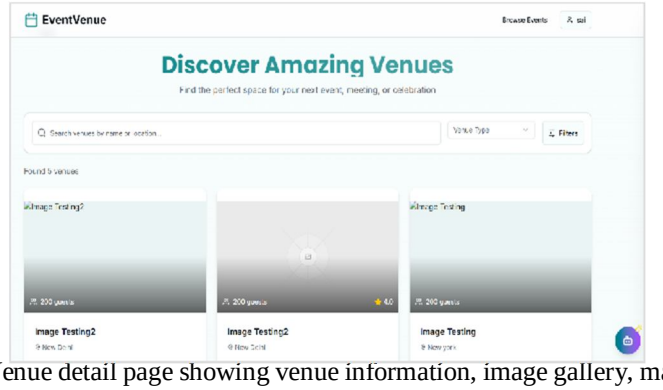


Fig. 4. Venue detail page showing venue information, image gallery, map, and reviews

C. Booking Page

The Booking page implements a step-by-step booking wizard that guides customers through date selection, time slot picking, cost calculation, points usage, and payment confirmation. A real-time cost breakdown panel displays the base price, applicable discounts from loyalty points, and the final payable amount. The integrated date picker prevents selection of already-booked dates. PayPal payment buttons are rendered inline for seamless checkout. Figure 5 shows the complete booking interface with cost breakdown and payment options.

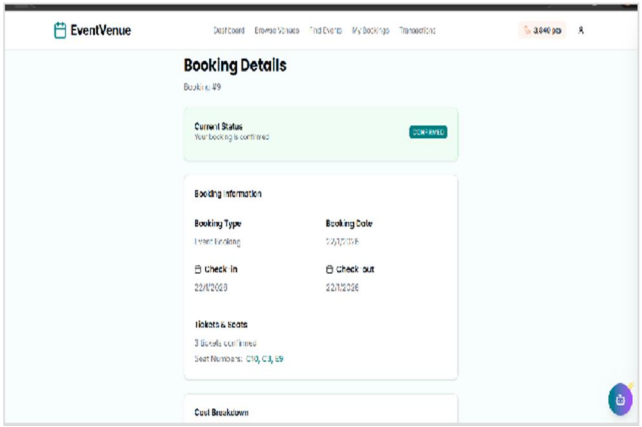


Fig. 5. Booking page with date picker, cost breakdown, points usage, and PayPal integration

D. Vendor Dashboard

The Vendor Dashboard provides vendors with a comprehensive management hub featuring KPI summary cards (total bookings, revenue, active venues, average rating), interactive analytics charts powered by Recharts (monthly revenue bar charts, booking trend line graphs, revenue distribution pie charts), and quick-action panels for venue management. The dashboard supports data- driven decision-making by presenting key business metrics at a glance, as illustrated in Figure 6.

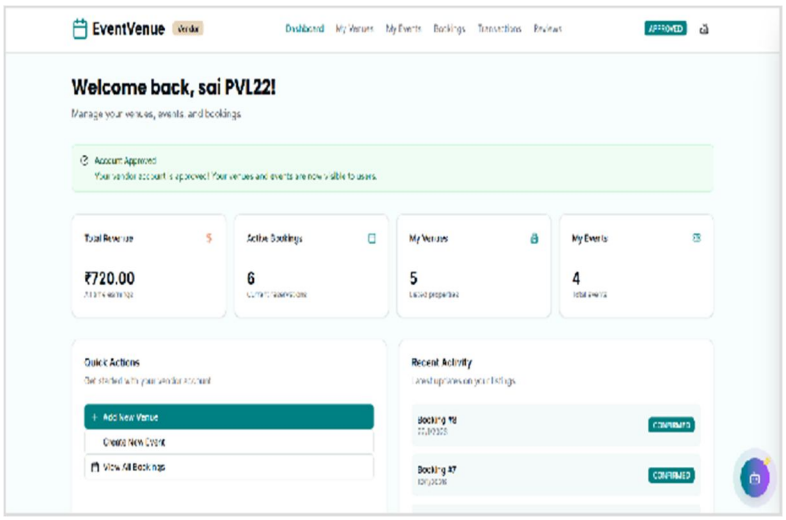


Fig. 6. Vendor dashboard showing analytics charts, KPI cards, and venue management

E. Admin Dashboard

The Admin Dashboard offers platform-level administrative controls with sections for user management, vendor approval workflows, booking oversight, and system analytics. KPI cards display total registered users, active vendors, total bookings, and platform-wide revenue. The vendor approval panel allows administrators to review pending vendor registrations and approve or reject them. Platform analytics visualize user growth, booking volumes, and revenue trends over time. Figure 7 presents the admin dashboard interface.

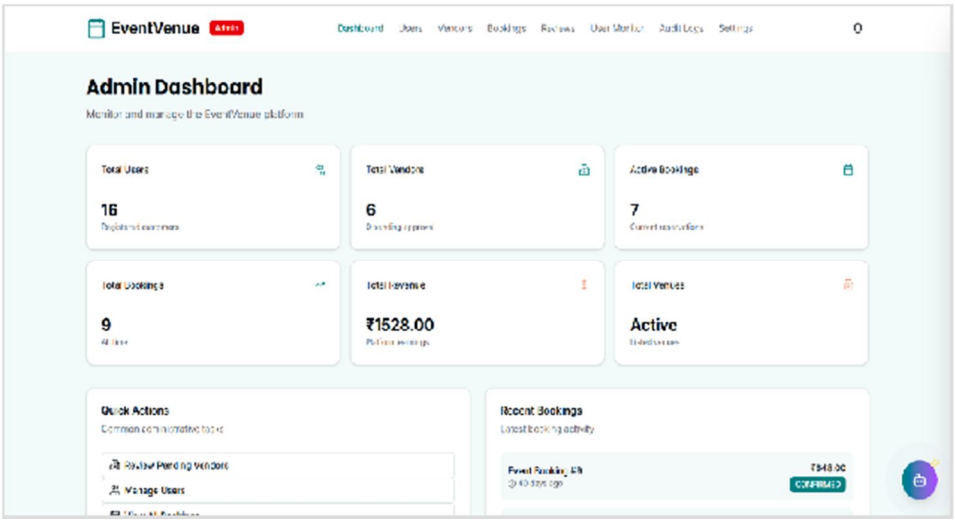


Fig. 7. Admin dashboard with user/vendor management, platform analytics, and vendor approval

F. Login and Signup with OTP Verification

The authentication module implements a secure multi- step registration and login flow. New users register with email and password, followed by a 6-digit OTP verification sent via Gmail SMTP. The login page features role-based routing—customers, vendors, and administrators are directed to their respective dashboards upon successful authentication. Vendors undergo an additional admin approval step before gaining full platform access. Figure 8 depicts the login/signup interface with the OTP verification modal.

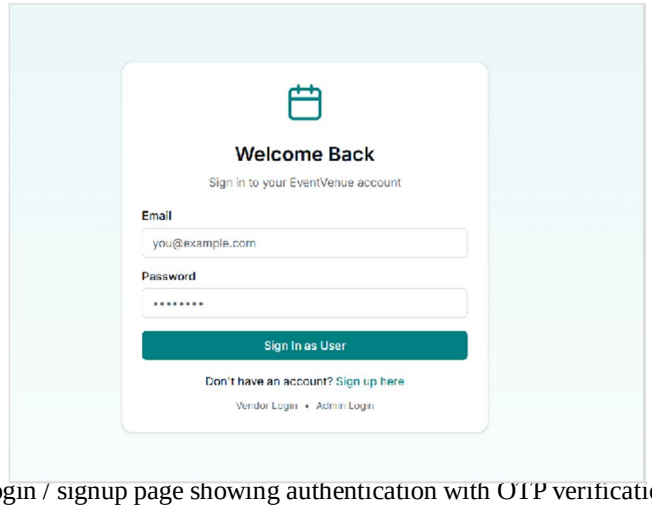


Fig. 8. Login / signup page showing authentication with OTP verification flow

G. Google Maps Integration

The platform integrates Google Maps JavaScript API v3 for location-aware venue discovery. Venues are displayed as interactive markers on an embedded map, with info windows showing venue name, rating, and pricing. Geocoding converts venue addresses to coordinates for precise map placement, while the Places Autocomplete API assists customers in entering location-based search queries. Figure 9 shows the Google Maps integration with venue markers.

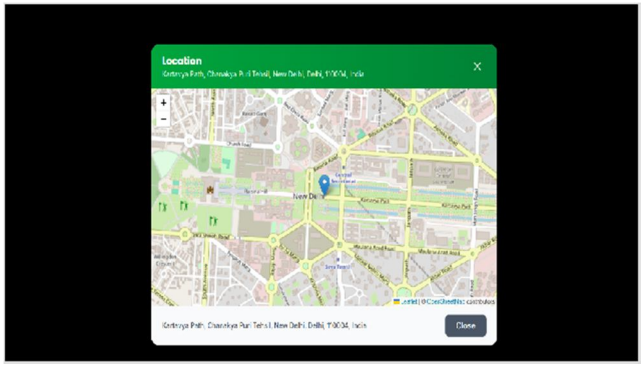


Fig. 9. Google Maps integration showing venue location with interactive markers

H. AI-Powered Chatbot

The Gemini-powered AI chatbot provides 24/7 intelligent customer support through a floating chat widget accessible from all pages. The chatbot handles venue recommendation queries, booking assistance, FAQ answers, and general platform guidance. It maintains conversational context across multiple messages and provides formatted responses with relevant venue links when applicable. Figure 10 demonstrates the AI chatbot widget in action.

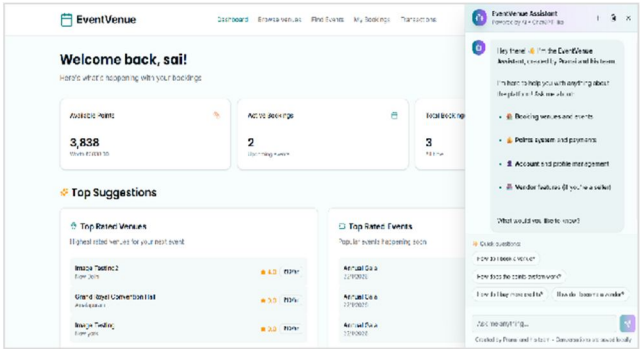


Fig. 10. AI chatbot widget powered by Google Gemini showing customer interaction

I. Event Ticketing Page

The Event Ticketing page allows vendors to create and manage events with configurable seat categories (e.g., VIP, Standard, Economy), each with independent pricing and capacity limits. Customers can browse upcoming events, view event details and available seat categories, select ticket quantities, and proceed to checkout. The interface displays real-time seat availability and pricing tiers. Figure 11 shows the event listing with seat category configuration.

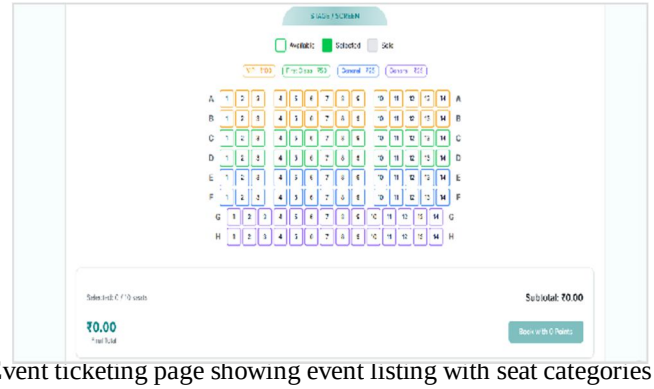


Fig. 11. Event ticketing page showing event listing with seat categories and pricing

J. Payment Confirmation

Upon successful payment via PayPal or points redemption, a confirmation page presents a detailed booking summary including venue/event name, date, time slot, amount paid, points used/earned, and a unique booking reference number. The confirmation is also sent via email notification. This page validates the end-to-end transaction workflow from booking initiation to payment finalization. Figure 12 shows the payment confirmation interface.

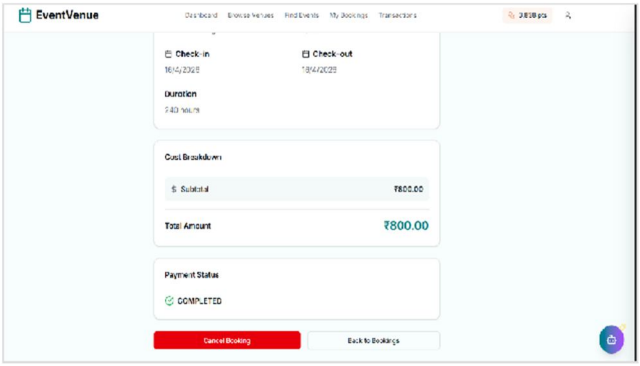


Fig. 12. Payment confirmation page showing booking details and PayPal transaction receipt

K. Unit and Integration Testing

Table V: Unit Test Coverage

Module	Tests	Coverage Area
Authentication Service	15	Login, signup, OTP, JWT generation
Booking Service	12	Cost calculation, status transitions, points
Venue Service	10	CRUD operations, search, filter
Payment Service	8	PayPal integration, refund processing
Points Service	6	Grant, deduct, refund, balance checks
Frontend Components	20	Form validation, rendering, state
Total	71	-

©IJRASET: All Rights are Reserved

Table VI: Integration Test Scenarios

ID	Scenario	Components	Result
IT-01	User registration → OTP → Login	Auth ↔ DB ↔ Email	✓ Pass
IT-02	Venue creation by vendor	Venue ↔ Service ↔ Repo	✓ Pass
IT-03	Complete booking with PayPal	Booking ↔ Payment ↔ Points	✓ Pass
IT-04	Vendor approval by admin	Admin ↔ Vendor ↔ DB	✓ Pass
IT-05	Points redemption in booking	Booking ↔ Points ↔ User	✓ Pass
IT-06	Booking cancellation + refund	Booking ↔ Payment ↔ Refund	✓ Pass

L. Performance Evaluation

Performance benchmarks were conducted on a development environment with 16GB RAM, Intel i7 processor, and PostgreSQL 14 on Supabase. Table VII presents the performance metrics across key platform operations.

Table VII: Performance Metrics

Metric	Target	Achieved	Notes
API Response (Avg)	< 200ms	145ms	Standard CRUD
API Response (Search)	< 500ms	320ms	Multi-criteria
Page Load (Initial)	< 3s	2.1s	SSR rendered
Page Load (Navigation)	< 1s	0.8s	Client-side
Concurrent Users	50+	60	No degradation
DB Query (Indexed)	< 50ms	35ms	Indexed columns
JWT Validation	< 10ms	5ms	Per-request auth
PayPal Processing	< 5s	3.2s	Sandbox env

M. Security Testing

Table VIII: Security Test Results

Vulnerability	Test Method	Result
SQL Injection	Malicious input in fields	✓ Prevented
Cross-Site Scripting	Script injection in reviews	✓ Prevented
JWT Tampering	Modified JWT payload	✓ Detected
Unauthorized Access	API calls without token	✓ Blocked (401)
Role Escalation	User → admin endpoints	✓ Blocked (403)
Brute Force	Repeated login attempts	✓ Rate-limited
CORS Bypass	Unauthorized origins	✓ Blocked

These observations confirm the effectiveness, usability, and robustness of the proposed EventVenue full-stack web application across all tested modules.

V. DISCUSSION

A. Design Validation and Key Observations

The EventVenue application follows a three-tier architectural design that ensures clear separation of concerns and improved maintainability. The use of Spring Boot and RESTful APIs enables structured communication between system components. Dynamic booking slot management prevents double booking and improves scheduling efficiency. Centralized digital venue records enhance data accessibility and continuity of information. Secure payment integration through PayPal allows transaction handling without complicating business logic, making the system suitable as an academic prototype.

B. Limitations and Constraints

Despite functional correctness, the system is not fully production-ready. Advanced security mechanisms such as Spring Security with encrypted credential storage and token-based authentication are implemented but could be enhanced further. The platform supports only single- organization deployment and lacks multi-tenant isolation. Performance optimizations and automated testing frameworks are also not fully incorporated.

C. Alignment with Literature Findings

The proposed system addresses key shortcomings identified in existing literature, particularly fragmented data management and lack of service integration. By combining venue discovery, booking management, payments, analytics, and AI-powered support within a unified architecture, EventVenue aligns with recommended best practices and provides a strong foundation for future enhancements.

D. Additional Features for Future Expansion

Beyond the core modules documented in this paper, several additional features present opportunities for further research and can significantly enhance the platform's capabilities:

1) Responsive Mobile UI: Documenting the platform's mobile-first responsive design with adaptive layouts for smartphones and tablets, including touch-optimized interactions and bottom navigation patterns.
2) Email Notification System: Comprehensive transactional email notifications covering booking confirmations, cancellation receipts, vendor approval notifications, and promotional emails with templated HTML designs via Gmail SMTP.
3) Vendor Revenue Withdrawal: The vendor wallet system with account balance tracking, withdrawal requests, and admin-managed payout processing.
4) Customer Booking History: A dedicated booking history module displaying past and upcoming reservations with status tracking, rebooking options, and booking receipts.
5) Venue Comparison Tool: Side-by-side comparison of multiple venues based on pricing, capacity, amenities, ratings, and location proximity.
6) Review Moderation System: Admin-side review moderation with flagging, approval workflows, and sentiment analysis for quality control.
7) Multi-language / Localization Support: Internationalization (i18n) infrastructure for supporting multiple languages and regional date/currency formats.
8) Wishlist and Favorites: Customer ability to save favorite venues and create wishlists for future event planning.
9) Social Media Sharing: Integration with social media platforms for sharing venue details and booking confirmations.

Each of these features can serve as an independent discussion topic in future work, expanding the scope of this research to cover the complete event management lifecycle.

VI. CONCLUSION

This paper presented EventVenue, a full-stack web application designed to modernize event venue management through centralized digital records, automated booking scheduling, secure online payments, an integrated marketplace for venue services, and AI-powered customer support. The system effectively addresses limitations of traditional venue booking practices and demonstrates improved efficiency and user experience in simulated urban environments. The modular and scalable architecture provides a strong foundation for future enhancements, including microservices migration, mobile application development, real-time WebSocket notifications, advanced AI-based venue recommendations, multi-currency and internationalization support, blockchain-based review verification, and predictive analytics for demand forecasting. With additional improvements in security and performance optimization, EventVenue can evolve into a robust and production-ready event venue management platform.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VII. ACKNOWLEDGMENT

The authors would like to thank Bonam Venkata Chalamayya Engineering College and the Department of Computer Science and Engineering for their support in this research work.

REFERENCES

[1] L. Zhang and R. Kumar, "Online Event Management Systems: A Comprehensive Review," International Journal of Information Management, vol. 45, no. 3, pp. 234–256, 2023.
[2] J. Smith, P. Anderson, and M. Lee, "Microservices Architecture in Booking Platforms: Design Patterns and Best Practices," IEEE Trans. Software Engineering, vol. 48, no. 7, pp. 1567–1582, 2022.
[3] W. Chen and S. Patel, "User Experience Design in Venue Booking Applications," J. Human-Computer Interaction, vol. 39, no. 2, pp. 89– 112, 2023.
[4] D. Williams and K. Thompson, "Secure Payment Gateway Integration in Web Applications," ACM Computing Surveys, vol. 54, no. 4, pp. 1– 35, 2022.
[5] M. Garcia and R. Johnson, "Role-Based Access Control in Multi- Tenant Applications," J. Information Security and Applications, vol. 67, pp. 103–125, 2023.
[6] L. Adams and B. White, "AI-Powered Customer Support in Booking Applications," Artificial Intelligence Review, vol. 56, no. 5, pp. 2341– 2368, 2023.
[7] A. Brown and E. Miller, "Real-Time Notification Systems in Booking Platforms," Int. Conf. Web Services, pp. 156–168, 2022.
[8] S. Taylor and J. Wilson, "Database Design for Event and Venue Management Systems," DASFAA Proceedings, pp. 445–462, 2023.
[9] C. Martinez and H. Davis, "Responsive Web Design for Cross- Platform Accessibility," Web Engineering Conf., pp. 78–94, 2022.
[10] T. Robinson and N. Clark, "Vendor Management Systems in Marketplace Platforms," Electronic Commerce Research and Applications, vol. 58, 101234, 2023.
[11] M. Thompson and P. Scott, "Calendar-Based Availability Management in Reservation Systems," Info. Systems J., vol. 32, no. 4, pp. 567–589, 2022.
[12] J. Harris and R. Evans, "Event Ticketing and Seat Category Management," Entertainment Computing J., vol. 44, 100512, 2023

BIOGRAPHIES OF AUTHORS



Bodapati Devi Swamy Sai Pranai is a highly skilled data analyst, set to graduate with a B.Tech degree from Bonam Venkata Chalamayya Engineering College, Odalarevu, India in 2026. With a strong foundation in Artificial Intelligence and Data Science, Pranai excels in data visualization tools, possessing advanced skills in PowerBI. His certifications include Python from Coursera, Introduction to Data Science from edX, and CS50 Introduction to Python from edX. Pranai completed an AI/ML/DS internship at IIDT, where he worked on PowerBI Data Analysis, Python, and Excel, and developed a dashboard on Private and Govt Colleges.
Email: Pranaib20@gmail.com
ORCID: https://orcid.org/0009-0008-1762-2144



Lasya Chavatapalli is a 4th-year Bachelor of Technology (B.Tech) student at Bonam Venkata Chalamayya Engineering College, specializing in Artificial Intelligence and Data Science. She is passionate about Machine Learning, Deep Learning, and Data Science, actively exploring advancements in AI. Recently, she successfully completed an 8-week internship in Full Stack Development with BlackBucks Engineers, where she gained hands- on experience in various development technologies and practical implementation skills.
Email: lasyachavatapalli2004@gmail.com **ORCID:** https://orcid.org/0009-0007-8829-0995 **GitHub:** https://github.com/LasyaChavatapalli



Harshitha Ramya Kadali is a fourth-year B.Tech student in Artificial Intelligence and Data Science at Bonam Venkata Chalamayya Engineering College, Andhra Pradesh, India. Her research interests include Artificial Intelligence, Machine Learning, Data Science, and Java-based software development. She is enthusiastic about learning emerging technologies and applying intelligent systems to address real-world challenges.
Email: ramyakadali107@gmail.com
ORCID: https://orcid.org/0009-0000-3926-5936
GitHub: https://github.com/Ramya-Kadali



Jaya Naga Rajeswari Penumalle is a 4th-year Bachelor of Technology (B.Tech) student at Bonam Venkata Chalamayya Engineering College, specializing in Artificial Intelligence and Data Science. She is passionate about Machine Learning, Deep Learning, and Data Science, actively exploring advancements in AI. Recently, she successfully completed an internship at BlackBucks (AIML/DS), where she gained hands-on experience in various AI/ML techniques and data science methodologies.
Email: jnrajeswari2004@gmail.com
ORCID: https://orcid.org/0009-0007-1221-0020
GitHub: https://github.com/Rajeswaril4

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

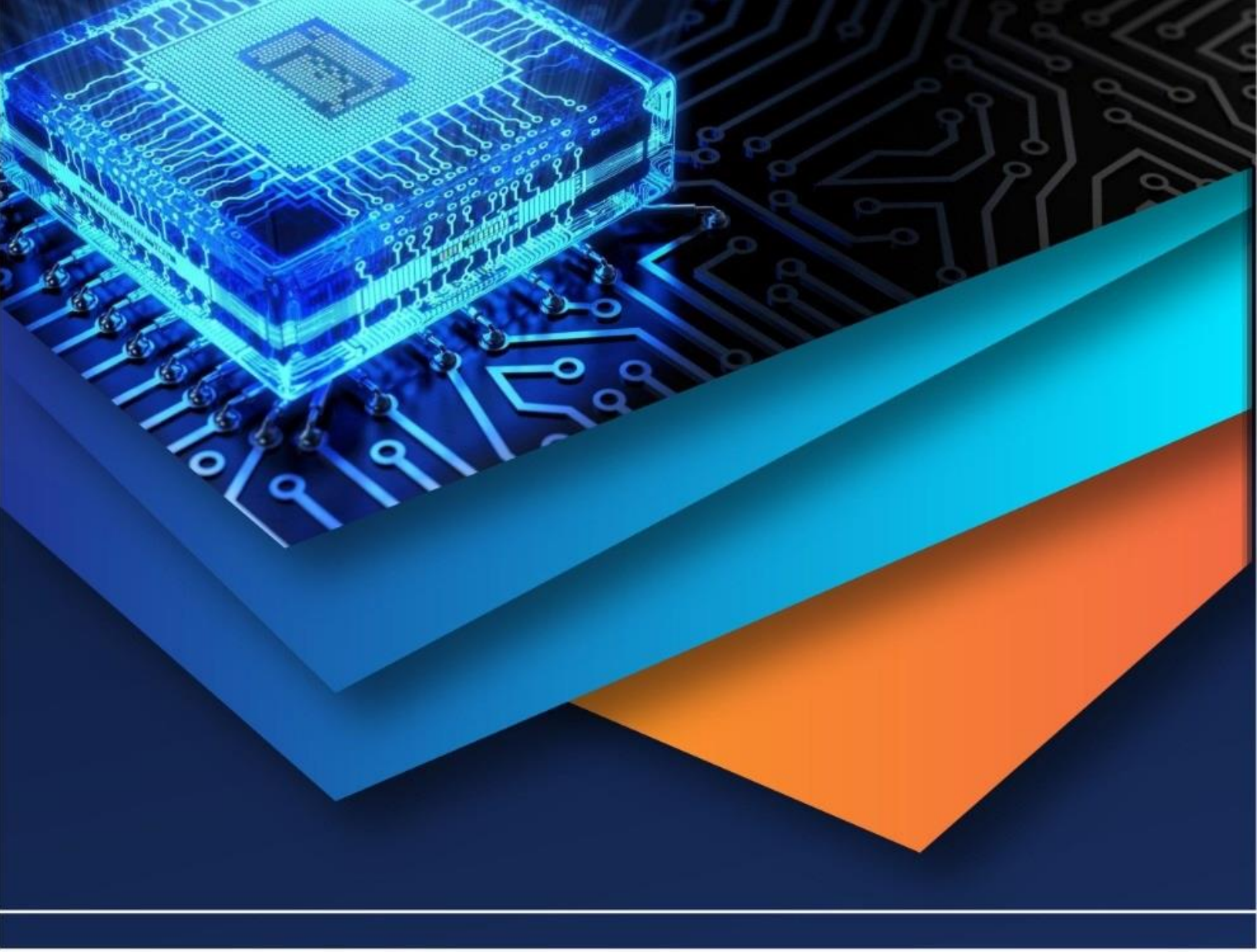


Mrs. Kankipati Varalakshmi, M.Tech is an Assistant Professor in the Department of CSE (Allied) at Bonam Venkata Chalamayya Engineering College, Odalarevu, affiliated to JNTU Kakinada, Andhra Pradesh, India. She serves as the project guide for this work. Her research interests include software engineering, web technologies, artificial intelligence, and educational computing.

Email: varuk919@gmail.com

ORCID: https://orcid.org/0009-0000-3530-2920

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)