EVOMEET - EVENT MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

KEERTHANA S
BALAMURUGAN B

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

SAVEETHA ENGINEERING COLLEGE, THANDALAM

An Autonomous Institution Affiliated to

ANNA UNIVERSITY - CHENNAI 600 025

NOVEMBER 2025

ABSTRACT

EvoMeet – Evolving the Way Events Are Managed is a web-based Event Management System developed using **Spring Boot**, **React.js**, and **MySQL**. The main goal of this project is to simplify and organize the process of creating, verifying, and participating in various types of events through a structured and transparent workflow. EvoMeet provides an efficient platform where users, organizers, managers, and admins can collaborate seamlessly to ensure every event is authentic and well-managed.

The system introduces four main roles—**Admin, Manager, Organizer, and User**—each with defined responsibilities. Organizers can create events such as book fairs, concerts, dance shows, or drama performances. Managers act as an intermediate layer, reviewing and approving events before they are made visible to users. This ensures quality control and prevents duplicate or fake event listings. Users can browse approved events, register for them, and complete a simulated payment process to confirm participation. The Admin oversees the entire system, managing users, events, and approvals, ensuring smooth and secure operation.

EvoMeet's backend, built with **Spring Boot**, handles all business logic and API integration, while the **React.js** frontend offers an interactive and responsive user experience. **MySQL** serves as the database, storing user details, event information, and registrations securely. Additional tools like **Postman**, **GitHub**, and **Maven** are used for API testing, version control, and project build management respectively.

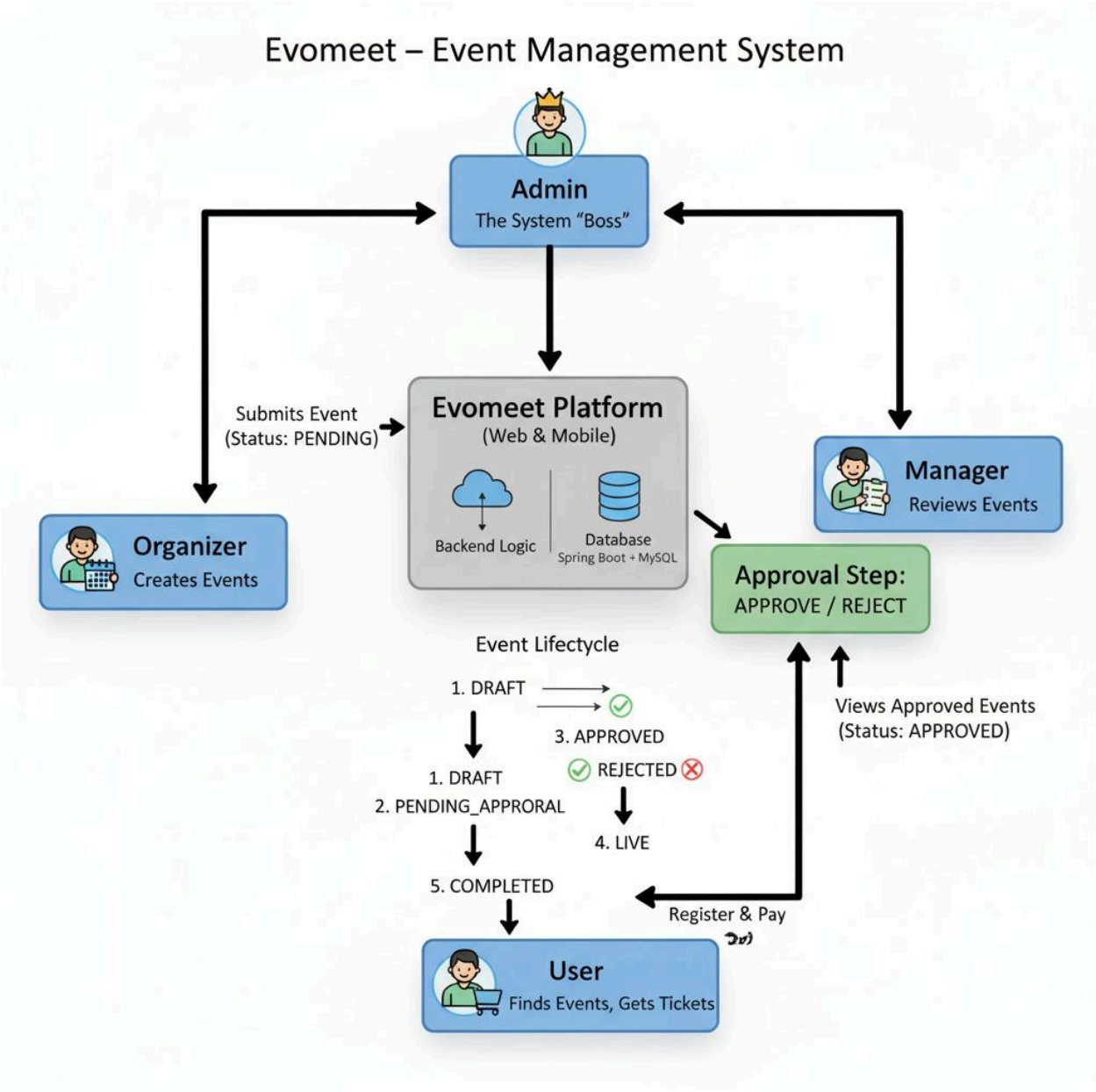
This project stands out for its **multi-level event approval workflow**, which ensures transparency and accountability at every stage. It provides an efficient system for managing events digitally while maintaining a professional and user-friendly interface. By combining modern web technologies with structured approval processes, **EvoMeet** successfully redefines how events are planned, reviewed, and experienced—making event management more organized, reliable, and accessible to everyone.

PROBLEM STATEMENT

Managing events manually often leads to issues such as poor coordination, lack of transparency, data inconsistency, and inefficient communication among organizers, participants, and administrators. Traditional event handling methods rely heavily on paperwork and human supervision, resulting in delays, duplicate entries, and limited scalability. There is also a lack of proper verification mechanisms to ensure the authenticity of events before being published to users.

To overcome these challenges, EvoMeet – Evolving the Way Events Are Managed aims to develop a centralized, web-based Event Management System that provides a structured workflow connecting Admins, Managers, Organizers, and Users. The system ensures smooth event creation, approval, and participation with real-time updates, secure data management, and a transparent approval process that enhances reliability, efficiency, and user experience.

ARCHITECTURE DIAGRAM



TECHNOLOGIES USED

- **Frontend:** React.js
    - Used to build a dynamic and user-friendly interface for smooth user interaction.
    - Developed using **Visual Studio Code (VS Code)** as the main IDE.
- **Backend:** Spring Boot (Java)
    - Handles business logic, RESTful API creation, and communication with the database.
    - Developed using **IntelliJ IDEA** for efficient coding and debugging.
- **Database:** MySQL
    - Used to store and manage application data such as user details, events, and registrations.
- **API Testing Tool:** Postman
    - Used for testing REST APIs and verifying request and response accuracy.
- **Version Control:** Git & GitHub
    - Used for source code management, collaboration, and maintaining version history.
- **Build Tool:** Maven
    - Manages dependencies and automates the project build process.

PROJECT PLAN

Phase 1: Project Planning and Requirement Analysis (Days 1–4)

Define project objectives, scope, and roles. Finalize tools, technologies, and prepare a basic workflow plan.

Phase 2: Backend Development (Days 5–12)

Develop the backend using Spring Boot in IntelliJ IDEA. Implement signup, login, and CRUD operations for events and categories.

Phase 3: Database Design and Integration (Days 13–16)

Design the MySQL database schema and integrate it with the backend. Establish entity relationships and test connectivity.

Phase 4: API Testing (Days 17–19)

Test all backend APIs using Postman. Validate responses, handle errors, and fix bugs for smooth performance.

Phase 5: Frontend Development (Days 20–25)

Build the frontend using React.js in VS Code. Design pages for login, event management, and dashboards connected to APIs.

Phase 6: Integration and Debugging (Days 26–28)

Integrate the frontend with the backend. Ensure proper data flow, fix integration issues, and optimize functionality.

Phase 7: Final Testing and Documentation (Days 29–30)

Conduct end-to-end testing, finalize the project, and prepare documentation for submission.