# EventX: PROJECT REPORT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Industry Project Title | | | EventX – Real-Time End-to-End Event Management System | | |
| Intern Name | | | Savinay K | | |
| Campus ID | | | 31437 | | |
| Registration No | | | 23BBCDAI181 | | |
| University | | | Yenepoya University, Bangalore | | |
| Start Date | End Date | Total Effort (hrs.) | | Project Environment | Tools Used |
| 20/12/2025 | 03/02/2026 | 65 Hrs. | | Django, Python, Bootstrap | VS Code, GitHub |

## TABLE OF CONTENT

* Acknowledgements
* Objective and Scope
* Problem Statement
* Existing Approaches
* Approach / Methodology – Tools and Technologies Used
* Workflow
* Assumptions
* Implementation – Data Collection, Processing Steps, Diagrams
* Solution Design
* Challenges & Opportunities
* Reflections on the Project
* Recommendations
* Outcome / Conclusion
* Enhancement Scope
* Link to Code and Executable File
* Research Questions and Responses
* References

## **Project Title**

**EventX – Real-Time End-to-End Event Management System**

**Internship Program:**

**TCS iON AIP 135: Real-Time End-to-End Event Management with Automated Build**

## PROJECT DETAILS

Project Title: EventX – Real-Time End-to-End Event Management System  
Internship Program: TCS iON AIP 135  
Intern Name: Savinay K  
Campus ID: 31437  
Registration Number: 23BBCDAI181  
University: Yenepoya University, Bangalore  
Project Domain: Full Stack Web Development (Real-Time Systems)  
Technologies Used: Django, Python, Bootstrap  
Database: SQLite3

## **ACKNOWLEDGEMENTS**

I would like to express my sincere gratitude to **TCS iON** for providing the Academic Internship Program (AIP 135), which offered an excellent platform to gain real-world industry exposure. This internship allowed me to apply theoretical concepts learned during my academic curriculum to a practical, real-time project environment.

I am deeply thankful to **Yenepoya University, Bangalore**, for providing continuous academic support throughout the internship period. I extend my heartfelt thanks to my faculty mentors and coordinators for their valuable guidance, encouragement, and constructive feedback. Their support played a crucial role in the successful completion of this project.

## **1. OBJECTIVE AND SCOPE**

### **1.1 Objective**

The primary objective of the EventX project is to design and develop a **real-time, automated, and admin-moderated event management system** that simplifies the complete lifecycle of event handling. The system aims to replace traditional manual event coordination processes with a digital solution that ensures efficiency, transparency, and security.

The project focuses on:

- Automating event submission and approval workflows

- Ensuring role-based access for users and administrators

- Providing real-time updates and status tracking

- Enhancing user experience through a modern UI

### **1.2 Scope**

The scope of EventX includes:

- User registration and authentication

- Event creation with image uploads

- Admin approval and rejection system

- Public display of approved events

- Secure backend processing and validation

- Responsive design for mobile, tablet, and desktop

- EventX can be used by educational institutions, corporate organizations, community groups, and event management agencies.

## **2. PROBLEM STATEMENT**

Traditional event management systems depend heavily on manual coordination through paperwork, spreadsheets, emails, and messaging applications. These methods often result in delayed approvals, lack of transparency, duplication of data, and inefficient communication.

Additionally, there is no centralized platform to track event status, manage approvals, or maintain historical records. This creates confusion among organizers and administrators. Hence, there is a strong need for a **centralized, real-time, and automated event management system** that ensures secure handling of event data and controlled publication.

## **3. EXISTING APPROACHES**

Existing approaches to event management include:

- Paper-based approval processes

- Spreadsheet-based tracking systems

- Email or messaging platforms for coordination

- Static websites without moderation

These approaches suffer from several limitations:

- Lack of automation

-No real-time updates

- Poor data security

- No role-based access control

- Limited scalability

-EventX overcomes these drawbacks by introducing a fully digital, admin-controlled workflow.

## **4. APPROACH / METHODOLOGY – TOOLS AND TECHNOLOGIES USED**

EventX is developed using a **Full Stack Web Development approach** based on Django’s **Model-View-Template (MVT)** architecture. This architecture ensures a clean separation between data logic, business logic, and user interface.

### **Technologies Used**

**Backend:**

Python 3.x, Django Framework

**Frontend:**

HTML5, CSS3, JavaScript (ES6)

**UI & Styling:**

Bootstrap 5.3, Glassmorphism UI, FontAwesome 6, Google Fonts (Outfit)

**Database:**

SQLite3 (Relational Database)

**Tools:**

Visual Studio Code, Git, GitHub, Django Admin Panel

## **5. WORKFLOW**

The workflow of EventX follows a structured sequence:

1. User registers and logs into the system

2. User submits a new event with complete details and images

3. The event status is set to *Pending* by default

4. Administrator reviews the event in the admin dashboard

5. Admin approves or rejects the event

6. Approved events are published publicly on the homepage

- This workflow ensures quality control and transparency.

## **6. ASSUMPTIONS**

- Users have access to the internet

- Users possess basic digital literacy

- Administrators actively manage event moderation

- The system is accessed through modern web browsers

## **7. IMPLEMENTATION – DATA COLLECTION, PROCESSING STEPS, DIAGRAMS**

### **7.1 Data Collection**

Data is collected through user-friendly web forms where users provide:

- Event name

- Event category

- Event date and location

- Event description

- Event banner image

### **7.2 Processing Steps**

- Server-side form validation using Django

- Secure image upload handling

- Data storage using Django ORM and SQLite

- Real-time updates to admin dashboard

- Status changes reflected instantly to users

## **8. SOLUTION DESIGN**

EventX follows a **modular and scalable design**:

- Separate backend and frontend layers

- Reusable templates and components

- Centralized database management

- Role-based access control

- This design improves maintainability and allows future enhancements with minimal changes.

## **9. CHALLENGES & OPPORTUNITIES**

### **Challenges**

- Secure handling of user-uploaded images

- Designing real-time admin updates

- Maintaining responsiveness across devices

### **Opportunities**

- Integration of analytics dashboards

- Email and SMS notification systems

- Event ticket booking and payment gateways

## **10. REFLECTIONS ON THE PROJECT**

This project significantly enhanced my understanding of:

- Django MVT architecture

- Full stack application development

- Secure authentication mechanisms

- Real-time workflow implementation

It also improved my problem-solving skills and confidence in developing production-ready systems.

## **11. RECOMMENDATIONS**

- Integrate email and SMS notifications

- Add advanced event search and filtering

- Implement payment gateway integration

- Provide admin analytics and reports

## **12. OUTCOME / CONCLUSION**

EventX successfully delivers a secure, automated, and user-friendly event management solution. The system meets all defined objectives and demonstrates the effective application of full stack web development concepts in solving real-world problems.

## **13. ENHANCEMENT SCOPE**

Future enhancements include:

- Mobile application development

- AI-based event recommendations

- Cloud-based database integration

- Advanced analytics dashboard

## **14. LINK TO CODE AND EXECUTABLE FILE**

GitHub Repository:

**<https://github.com/SavinayKnair/EventX.git>**

## **15. RESEARCH QUESTIONS AND RESPONSES**

**Question:** How can real-time approvals be handled securely?

**Answer:** By implementing role-based authentication and admin moderation using Django’s built-in security features.

## **16. REFERENCES**

- Django Official Documentation

- Python Official Documentation

- Bootstrap Documentation

- W3Schools

- Google Fonts

## 17. CONCLUSION

EventX successfully demonstrates a real-time, full-stack event management solution.   
The project fulfills all objectives and provides a strong foundation for future improvements.   
It showcases practical application of Django, Python, and modern UI technologies in solving   
real-world problems.

## 18. ScreenShots

