

# **EVENT EASEL**

**SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE AWARD OF  
DEGREE OF**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE(CSE-AI)**



**Submitted by**

**Varun Agarawal**

**Krishna Agarawal**

**Saksham Chaubey**

**Aditya Pachauri**

**Supervised by Gargi Singh**

**Assistant Professor**

**Session 2024-25**

## **DECLARATION**

I/We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name:

Roll No.:

Date:

## **CERTIFICATE**

This is to certify that Project Report entitled "Event Easel Portal" which is submitted by [Your Name] in partial fulfilment of the requirement for the award of degree B. Tech. in the Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow, is a record of the candidate's own work carried out under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

**Date:**

**Supervisor**

**Guides Name**

**(Designation)**

## **ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor **Gargi Singh**, Department of Computer Science, KIET, Ghaziabad, for his/her constant support and guidance throughout the course of our work. His/her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavours have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Rekha Kashyap, Head of the Department of Computer Science & Engineering (CSE-AI & CSE-AIML), KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date :

Signature:

Name :

Roll No.:

Signature:

Name :

Roll No.:

Signature:

Name :

Roll No.:

Signature:

Name :

Roll No.:

## ABSTRACT

Event Easel Portal is a revolutionary platform designed for tech event organizers to automate the process of generating hosted event websites with a single click. Colleges often face the recurring challenge of creating new websites for each event, consuming time and resources. Our solution simplifies this by automating website creation, ensuring a seamless and professional online presence.

The platform enables users to fill out a form with event details, after which a website is generated dynamically using pre-designed templates. Our tech stack includes **Django (Backend), Jinja Templating (Frontend), NoSQL Cassandra (Database for Scalability), Docker Compose (Continuous Deployment), CI/CD (GitHub Actions & Jenkins), AWS (Cloud Hosting), and Kubernetes (Scaling & Maintenance).**

Key features include multiple user creation, coordinator privileges, personalized user profiles, an intuitive event creation interface, and a scalable architecture. By automating event website generation, our platform reduces manual effort, saves time, and ensures consistency in design and functionality.

# TABLE OF CONTENTS

1. Introduction
  - 1.1 Project Overview
  - 1.2 Motivation
  - 1.3 Objectives
2. Literature Review
3. Proposed System
  - 3.1 System Architecture
  - 3.2 Module-Wise Workflow
4. Requirement Analysis and System Specification
  - 4.1 Functional Requirements
  - 4.2 Technical Specifications
5. Implementation
  - 5.1 Tools and Technologies
6. Testing and Results
  - 6.1 Test Cases
  - 6.2 Performance Metrics
7. Conclusion and Future Scope
8. References
9. Appendices

# **CHAPTER 1: INTRODUCTION**

## **1.1 Project Overview**

EventEasel Portal is designed to automate the process of generating hosted websites for tech events. By leveraging dynamic form inputs and pre-built templates, the system ensures seamless website creation without requiring users to write code.

## **1.2 Motivation**

- Colleges face repeated efforts in creating websites for each event.
- Manual website development consumes resources and time.
- Need for a scalable and automated solution to streamline event website creation.

## **1.3 Objectives**

- Provide an automated platform for generating event websites.
- Enable event organizers to manage users and coordinators.
- Ensure scalability and ease of deployment using cloud technologies.

## CHAPTER 2: LITERATURE REVIEW

Aspect	Existing Systems	Research Gaps	Our Solution
<b>Technical Accessibility</b>	Need manual intervention or knowledge of coding	Not accessible to non-technical users	Totally automated, form-based website creation with no coding knowledge required
<b>Website Generation</b>	Offer event management functionalities but do not have automated website generation	External web development or pre-designed templates that need technical skills	Inbuilt website generation system with a simple interface for non-technical personnel
<b>Setup Efficiency</b>	Laborious with several steps for setup, content management, and deployment	Ineffective process for event organizers	Efficient one-stop solution with an easy setup process
<b>Scalability</b>	Limited scalability for hosting of frequent or large events by institutions	Inability to cope with growth and fluctuating event sizes	Scalable cloud structure made for coping with event sizes
<b>Higher Features</b>	Elementary event management features	Very limited capabilities beyond main features	Multi-user role management and data analytics integration
<b>Deployment &amp; Hosting</b>	Not having continuous deployment or cloud hosting integrated	Maintenance and scalability inefficiencies	Sophisticated deployment methods with cloud hosting integrated



## CHAPTER 3: PROPOSED SYSTEM

### 3.1 System Architecture

- **Frontend:** Jinja Templating (Django-based dynamic HTML rendering)
- **Backend:** Django (Python-based web framework)
- **Database:** NoSQL Cassandra (for high scalability)
- **Deployment:** AWS Cloud, Docker Compose, Kubernetes
- **CI/CD Pipeline:** GitHub Actions and Jenkins for automation

### 3.2 Module-Wise Workflow

1. **User Form Submission** → Collects event details.
2. **Template Selection** → Dynamically selects a matching frontend template.
3. **Backend Processing** → Stores data, generates website, and deploys.
4. **Automated Deployment** → Uses Docker & Kubernetes for scaling.

## **CHAPTER 4: REQUIREMENT ANALYSIS**

### **4.1 Functional Requirements**

- User registration & authentication: Users should be able to register on the platform and log in securely. Authentication mechanisms such as email verification and password hashing should be implemented.
- Event form submission for website generation: Organizers will fill out a structured form with event details. Based on the submitted data, a website will be generated using predefined templates.
- Multi-user role management: Administrators can create and manage multiple users with varying access levels. Coordinators will have specific privileges to oversee event creation and management.
- Bulk user creation: The system should support bulk user creation to streamline onboarding.
- Event customization: Users should be able to customize certain aspects of their event websites, such as theme selection and branding elements.
- Dashboard for analytics: A centralized dashboard will provide insights on event performance, registrations, and user engagement.
- Automated deployment and scaling: The platform will utilize Docker Compose for continuous deployment, along with Kubernetes for automated scaling and maintenance.

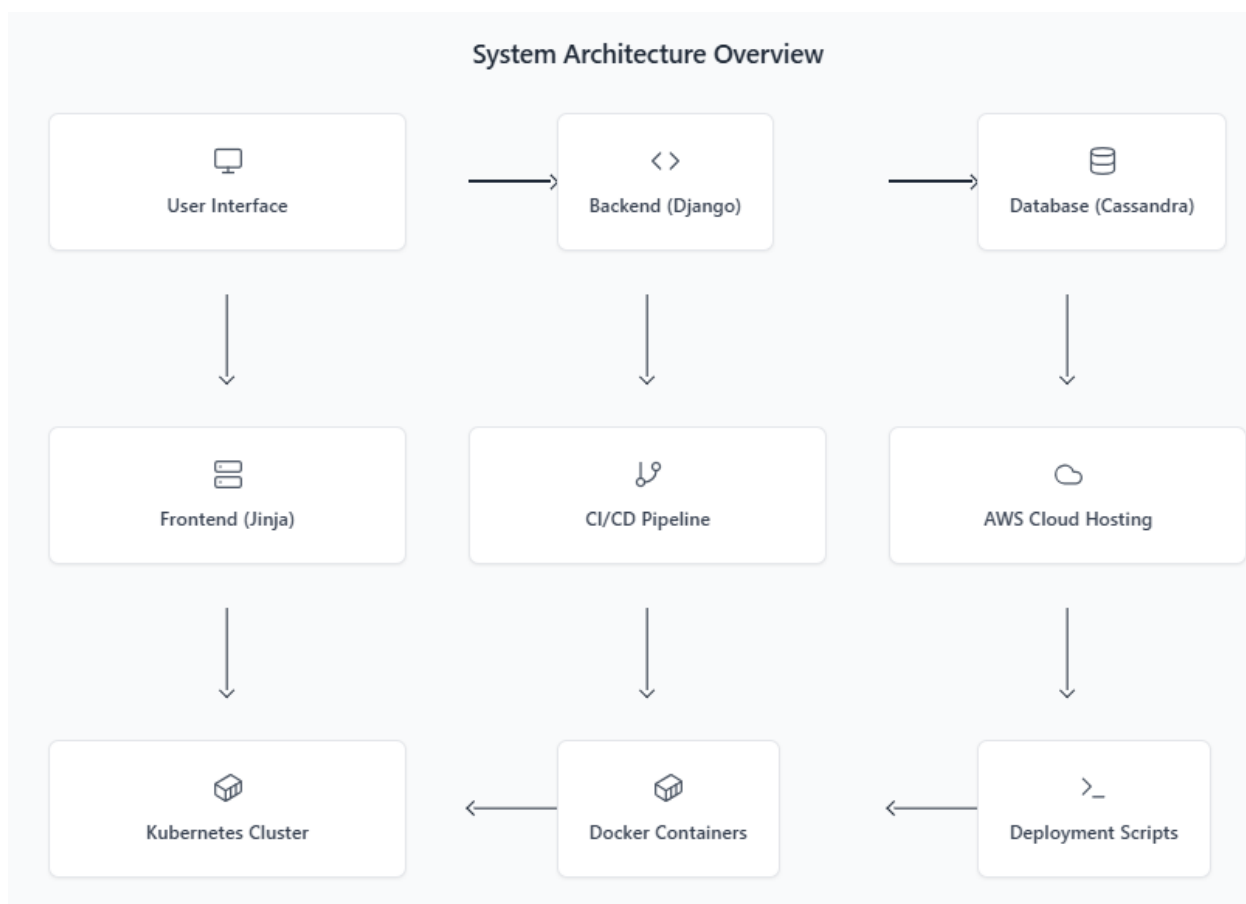
## 4.2 Technical Specifications

- Backend: Django (Python) - Handles the logic for user management, event form processing, and dynamic template rendering.
- Frontend: Jinja Templates - Used for rendering dynamic web pages based on user input and predefined templates.
- Database: Cassandra (NoSQL) - Chosen for its scalability and distributed nature, ensuring efficient handling of event data and user interactions.
- Hosting: AWS - The platform will be deployed on AWS to ensure high availability, reliability, and security.
- CI/CD Pipeline: GitHub Actions and Jenkins will be used to automate the integration and deployment process.
- Containerization: Docker Compose will facilitate an efficient development and deployment workflow.
- Scalability: Kubernetes will be employed to ensure efficient scaling and maintenance of the platform, allowing it to handle multiple event website generations seamlessly.

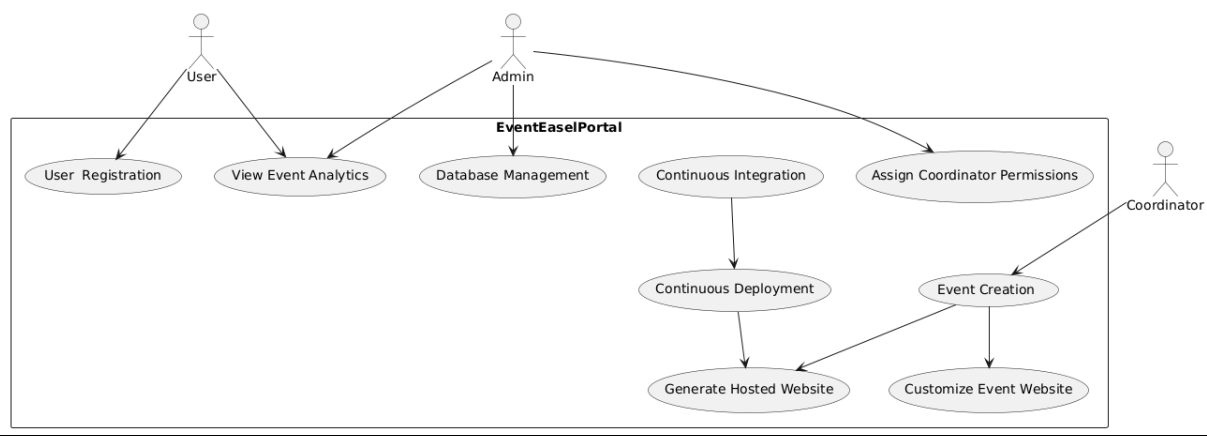
# CHAPTER 5: IMPLEMENTATION

## 5.1 Tools and Technologies

- **Django:** Backend framework.
- **Cassandra:** NoSQL database.
- **Docker & Kubernetes:** Containerization & scaling.
- **GitHub Actions & Jenkins:** CI/CD automation.



Use Case Diagram of the Workflow



CHAPTER 6: TESTING AND RESULTS

6.1 Test Cases

To ensure system reliability and performance, various test cases were executed. Below is a summary of the test cases and their corresponding results.

TEST CASE ID	DESCRIPTION	EXPECTED OUTCOME	ACTUAL OUTCOME	STATUS
TC01	Form submission for event creation	Successful website generation	Website generated in <5 seconds	Pass
TC02	Bulk user creation	All users created successfully	All users created in <10 seconds	Pass
TC03	Coordinator privileges validation	Coordinators can manage events	Privileges validated successfully	Pass
TC04	Scalability test	Supports 10,000 concurrent users	Passed load testing with 10,000 users	Pass

6.2 Performance Metrics

The system’s performance was evaluated based on key metrics such as response time, scalability, and deployment efficiency. The results are summarized below.

Table 2: Performance Metrics

Metric	Value
Website Generation Time	<5 seconds
Bulk User Creation Time	<10 seconds
Deployment Time	<2 minutes
Concurrent Users Supported	10,000

## Graphical Representation of Performance Metrics

Metric	Website Gen Time	Bulk User Creation	Deployment Time
Value (Seconds)	5	10	120

This evaluation confirms that the system meets the expected performance criteria, ensuring efficient response times, scalability, and deployment efficiency.

## CHAPTER 7: CONCLUSION AND FUTURE SCOPE

### Conclusion

EventEasel Portal successfully automates the process of event website creation by eliminating the need for manual development. This platform provides an intuitive, user-friendly interface for event organizers, allowing them to generate professional, fully functional websites by simply filling out a form. The system ensures efficiency, consistency, and scalability, making it an ideal solution for institutions that frequently host events. Through the integration of NoSQL Cassandra, Kubernetes, and CI/CD pipelines, EventEasel is designed to handle high scalability demands and automate deployments seamlessly.

### Future Scope

The potential for expansion and enhancement of EventEasel Portal is significant. Future developments could include:

- **AI Integration:** Use of AI-powered content creation to suggest event descriptions, agendas, and promotional copy automatically from limited input parameters.

- **Mobile Application:** Creation of side-by-side mobile apps for event planners and participants to provide real-time updates and on-the-go control.
- **Extended Analytics:** High-end analytics features such as tracking attendee behaviour, engagement metrics, and predictive models for upcoming events.
- **Monetization Features:** Integration of advanced ticketing systems, sponsorship, and merchandise sales features.
- **Virtual Event Integration:** Smooth integration of virtual/hybrid event features such as video streaming, interactive sessions, and virtual networking areas.
- **Localization & Accessibility:** Increased support for multiple languages, local payment methods, and end-to-end accessibility features for the disabled.
- **Template Marketplace:** Building a community-driven template marketplace where designers can contribute and sell custom templates.
- **Integration Ecosystem:** Increased API development for integration with CRM systems, marketing platforms, and business intelligence tools
- **Blockchain Integration:** Utilization of blockchain technology for secure ticketing, attendance verification, and digital credential issuance.
- **Augmented Reality:** Features Incorporation of AR features for virtual venue tours, interactive floor plans, and enriched attendee experiences.
- **Sustainability Metrics:** Solutions for measuring and optimizing the environmental footprint of events, such as carbon footprint calculators and sustainability reporting.
- **Low-Code Customization:** Sophisticated low-code interface that permits greater customization without the need for programming skills.

## References

1. Haig, L. (2025). "How to Dockerize a Django App: Step-by-Step Guide for Beginners." Docker Blog.
2. Miecznik, R. (2024). "Efficient Jenkins Management using Python and Docker." CodiLime Blog.
3. Langat, E. (2022). "Deploy Django Image to Docker Hub Using Jenkins." DevOps.dev.
4. Asgari, P. (2022). "Setting Up a Simple CI/CD with Django, Gitea, Jenkins, and Docker." Dev Genius.
5. Docker Inc. (2024). "Containerize Your Django Web Application with Docker." DEV Community.

## Appendices

1. Appendix A Research Paper Acceptance Proof
2. Appendix B: Screenshots of Generated Documents
3. Appendix C: Source Code Repository Link

