VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI-590 018, KARNATAKA

MINI PROJECT REPORT

ON

"DEPARTMENTAL EVENT MANAGEMENT SYSTEM"

Submitted in the partial fulfilment of requirements for the award of Degree

B.E. in Computer Science and Engineering

PROJECT ASSOCIATES

Kavyashree H G	**4BD21CS059**
K Subramanyam	**4BD21CS053**
Komal P J	**4BD21CS062**
Meghana B	**4BD22CS074**

PROJECT GUIDE

Prof. ANU C S
Assistant Professor
Dept. of CS&E

Bapuji Institute of Engineering and Technology
Department of Computer Science and Engineering
Davanagere-577004

July 2024

Bapuji Institute of Engineering and Technology
Davangere – 577004

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **Kavyashree H G, K Subramanyam, Komal P J, Meghana B** bearing USN **4BD21CS059, 4BD21CS053, 4BD21CS062 and 4BD22CS074** respectively of Computer Science and Engineering department have satisfactorily submitted the Mini Project report entitled **"Departmental Event Management System"** in the partial fulfilment of the requirements for the award of Degree of Bachelor of Engineering (B.E.) in Computer Science and Engineering, under the VTU during the academic year 2023-24**.**

Project Guide
Prof. Anu C S
Assistant Professor
Dept. of CS&E

Mini-Project Co-ordinator
Prof. Shankar Sarji P
Assistant Professor
Dept. of CS&E

Dr. Nirmala C R Ph.D.
Head of Department
CS & E

Dr. H B Aravind Ph.D.
Principal

Date: 19-07-2024

Place: Davangere

ACKNOWLEDGEMENT

Salutations to our beloved and highly esteemed institute, **"BAPUJI INSTITUTE OF ENGINEERING AND TECHNOLOGY"** for having well-qualified staff and labs furnished with the necessary equipment.

We express our sincere thanks to our guide **Prof. Anu C S,** Assistant Professor, Department of Computer Science and Engineering, B.I.E.T., Davangere, for giving us the constant encouragement, support and valuable guidance throughout the course of the mini project without whose stable guidance this project would have not been achieved.

We are grateful to mini–Project Coordinators **Dr. Arun Kumar G H,** Associate Professor, Department of Computer Science and Engineering, B.I.E.T., **Dr. Naveen Kumar K R,** Associate Professor, Department of Computer Science and Engineering, B.I.E.T., and **Mr. Shankar Sarji P**, Assistant Professor, Department of Computer Science and Engineering, B.I.E.T., Davangere, for their constant support and feedback through-out this mini-project.

We express whole hearted gratitude to **Dr. Nirmala C R** who is our respectable H.O.D of Computer science and Engineering Department. We wish to acknowledge her help who made our task easy by providing with her valuable help and encouragement

We also express our wholehearted gratitude to our respected Principal, **Dr. H B Aravind** for his moral support and encouragement.

We would like to extend our gratitude to all Teaching and Non-Teaching staff of the **Department of Computer Science and Engineering** for the help and support rendered to us. We have benefited a lot from the feedback, and suggestions given by them.

We would like to extend our gratitude to all our family members and friends especially for their advice and moral support.

Kavyashree H G (4BD21CS059)
K Subramanyam (4BD21CS053)
Komal P J (4BD21CS062)
Meghana B (4BD22CS074)

Bapuji Educational Association (Regd.)

Bapuji Institute of Engineering and Technology, Davangere-577004

Vision and Mission of the Computer Science and Engineering Department

Vision

"To be a centre-of-excellence by imbibing state-of-the-art technology in the field of Computer Science and Engineering, thereby enabling students to excel professionally and be ethical."

Mission

1.	Adapting best teaching and learning techniques that cultivates Questioning and Reasoning culture among the students.
2.	Creating collaborative learning environment that ignites the critical thinking in students and leading to the innovation.
3.	Establishing Industry Institute relationship to bridge skill gap and make them industry ready and relevant.
4.	Mentoring students to be socially responsible by inculcating ethical and moral values.

Program Educational Objectives (PEOs):

PEO1	To apply skills acquired in the discipline of computer science and engineering for solving Societal and industrial problems with apt technology intervention.
PEO2	To continue their carrier ion industry /academia or pursue higher studies and research.
PEO3	To become successful entrepreneurs, innovators to design and develop software products and services that meets societal, technical and business challenges.
PEO4	To work in the diversified environment by acquiring leadership qualities with effective communication skills accompanied by professional and ethical values.

Program Specific Outcomes (PSOs):

PSO1	Analyse and develop solutions for problems that are complex in nature but applying the knowledge acquired from the core subjects of this program.
PSO2	To develop secure, scalable, resilient and distributed applications for industry and societal Requirements.
PSO3	To learn and apply the concepts and contract of emerging technologies like artificial intelligence, machine learning, deep learning, big-data analytics, IOT, cloud computing etc for any real time problems.

ABSTRACT

The Departmental Event Management System is a comprehensive web-based application designed to streamline and enhance the management of college events. Aimed at providing an efficient platform for students and event organizers, it facilitates seamless event registration, scheduling, management and feedback collection. The system employs Django for backend development, ensuring robust data handling and secure user interactions, while the frontend utilizes HTML, CSS, and JavaScript to deliver an intuitive user experience. By integrating modern web technologies, the system provides a scalable solution that can adapt to various event management needs, thereby fostering a more organized and interactive campus environment.

CONTENTS

LIST OF FIGURES

CONTENT PAGE NO

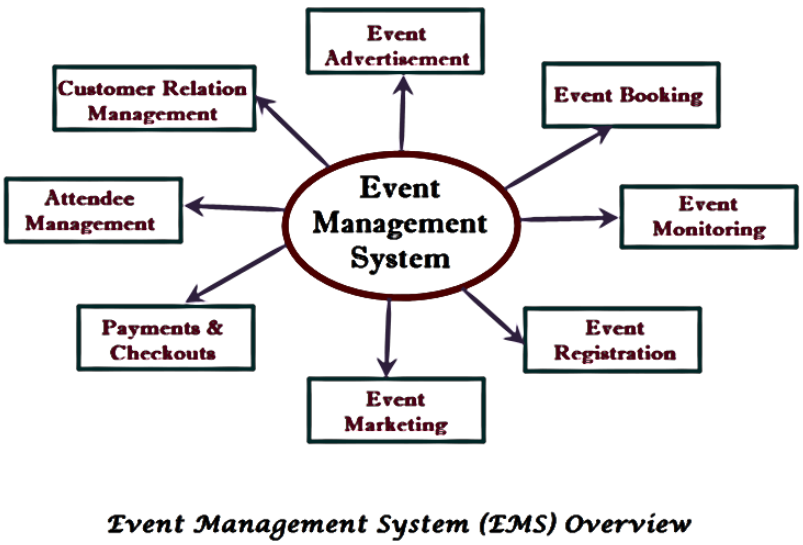
CHAPTER 1: INTRODUCTION

In the dynamic environment of higher education, organizing and managing departmental events such as workshops, events, seminars, and conferences is a significant challenge. The need for an efficient system to handle event registrations, scheduling, and feedback collection has become increasingly essential. Our Departmental Event Management System aims to address these challenges by providing a comprehensive, web-based application tailored for college departments.

Our website leverages modern web technologies, including Django for the backend and HTML, CSS, and JavaScript for the frontend, to deliver a robust and user-friendly platform. The system is designed to streamline the entire event management process, from planning and scheduling to participant registration and feedback collection. By integrating these functionalities into a single platform, it simplifies the workload for event organizers and enhances the experience for participants.

1.1 OVERVIEW OF THE PROJECT

The primary goal of this website is to create a seamless and efficient event management experience for both organizers and participants. By automating many of the manual processes associated with event management, the system aims to reduce administrative burden and improve overall event coordination. The subsequent sections of this report provide a detailed overview of the system's design and implementation, offering insights into the development process and the challenges encountered along the way. Through this project, we aim to contribute a valuable tool for managing departmental events, ultimately fostering a more organized and interactive academic environment.



Event Management System (EMS) Overview

Key Features:

- User Authentication and Roles: Secure login for students, organizers, and administrators.
- Event Management: Easy creation, editing, and deletion of events by organizers.
- Registration System: Online event registration for students with instant confirmation.
- Scheduling: Clear and organized event schedule view for all users.
- Feedback Collection: Post-event feedback system to gather attendee insights.
- Notifications: Automated email reminders and updates for registered participants.

System Architecture:

- Frontend: User interfaces developed with HTML, CSS, and JavaScript.
- Backend: Data management handled by Django framework.
- Database: SQLite database to store user information, event details, registrations, and feedback.

Development Tools:

- IDE: Visual Studio Code
- Version Control: Git

Project Phases:

1. Planning and Requirement Gathering: Identifying the needs of students and organisers, defining project scope, and outlining system requirements.
2. Design: Creating wireframes, mock-ups, and system architecture diagrams to visualize the system's structure and functionality.
3. Development: Implementing the frontend and backend components, integrating them to build a cohesive system.
4. Testing: Conducting thorough testing to ensure the system functions as expected, identifying and fixing bugs.
5. Deployment: Deploying the system on a web server, making it accessible to users.
6. Maintenance: Providing ongoing support and updates to ensure the system remains functional and secure.

1.2 PROBLEM STATEMENT

Managing departmental events in academic institutions is often inefficient due to manual processes and fragmented tools. This leads to challenges in event organization, communication, registration, feedback collection, and data management. The need for a centralized, integrated solution is evident to

streamline these processes, enhance communication, simplify registration, and improve data management. Our Departmental Event Management System addresses these issues by providing a comprehensive platform for efficient event coordination.

1.3 OBJECTIVES

The main objective of this project is:

- To develop a centralized platform to efficiently organize, schedule, and manage departmental events such as workshops, seminars, and conferences.
- To provide an intuitive and user-friendly interface for students, organizers, and administrators to easily interact with the system.
- To implement an online registration system for students, ensuring a smooth and error-free sign-up process with instant confirmation.
- To enable automated notifications and reminders to keep participants and organizers informed about event updates and schedules.
- To integrate a feedback system to gather and analyse participant insights, helping organizers evaluate event success and areas for improvement.

CHAPTER 2: METHODOLOGY

2.1 METHODS AND TECHNIQUES

The development of the Departmental Event Management System will follow a structured approach, employing the Agile methodology to ensure iterative progress and continuous improvement. The project will be divided into several phases: Planning and Requirement Gathering, Design, Development, Testing, Deployment, and Maintenance. They can be listed as:

1. Planning and Requirement Gathering

- Conducted meetings with key stakeholders, including students, event organizers, and administrators, to gather requirements and understand their needs.
- Documented functional and non-functional requirements, creating a comprehensive list of features and specifications.
- Developed a detailed project plan, including timelines, milestones, and resource allocation.

2. Design

- **System Architecture:** A simple system architecture showing admin, user and database is shown below:

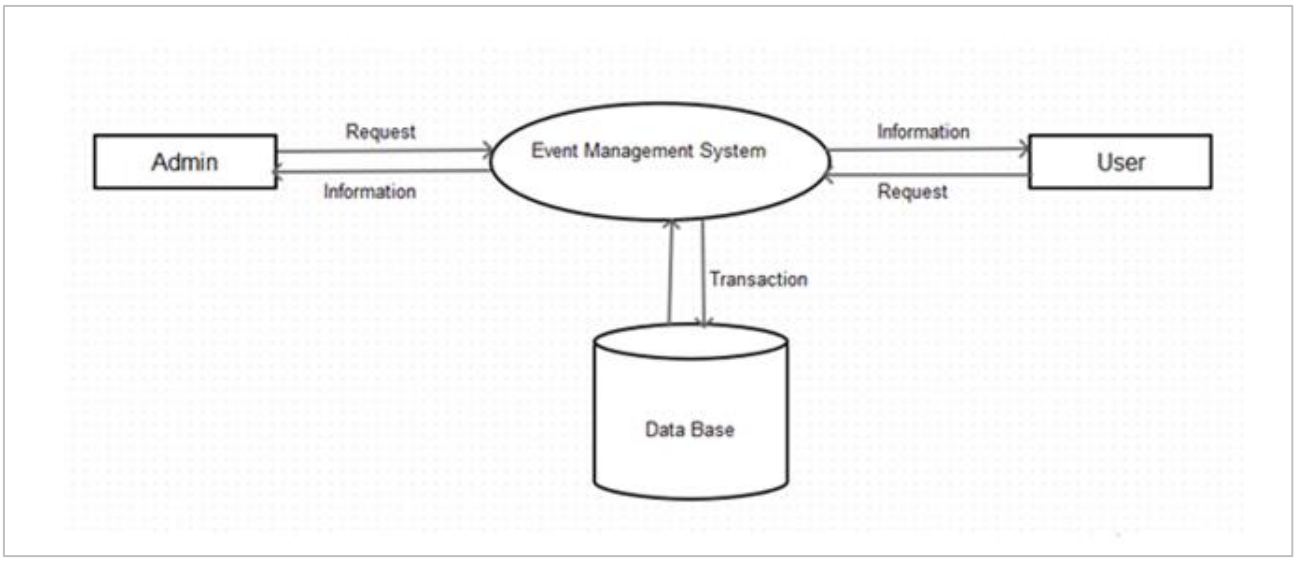


Fig 2.1.1 Simple System Architecture

- **Use Case Diagram:** Use case diagram shows us how entities in the database are related to each other.

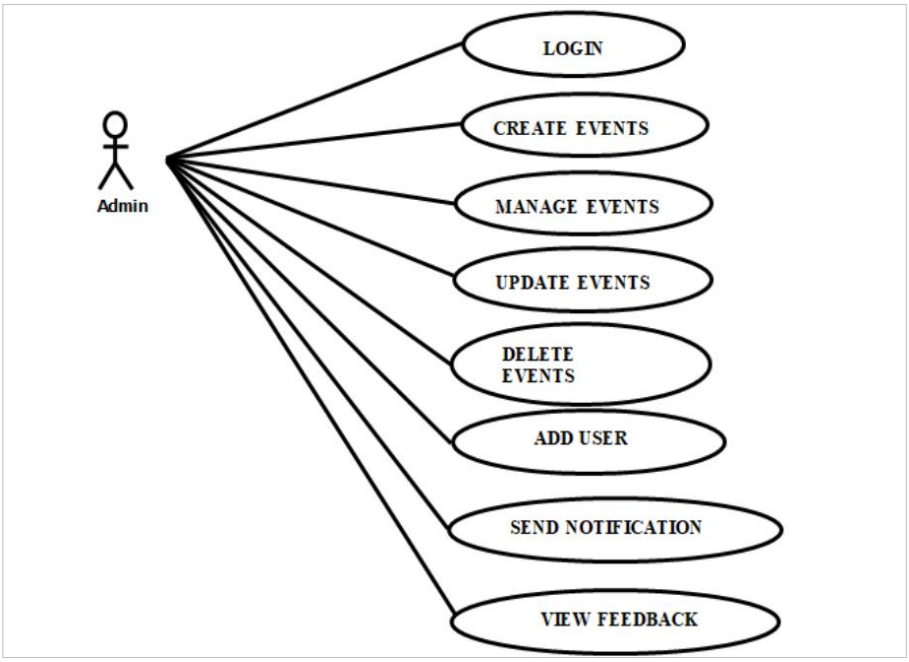


Fig.2.1.2 Use case diagram of Admin

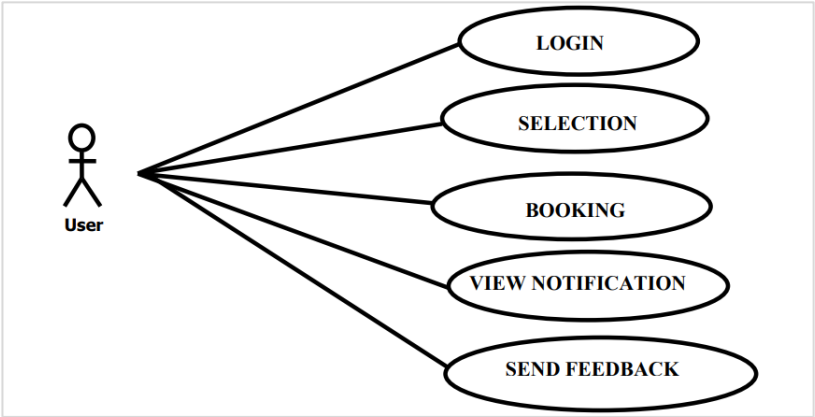


Fig.2.1.3 Use case diagram of User

- **Technical Specifications:** Document the technical specifications, including the technologies, frameworks, and tools to be used.
- **Database Design:** An Entity-Relationship (ER) diagram to outline the database schema, including tables for users, events, registrations, and feedback. An related ER diagram is shown below:

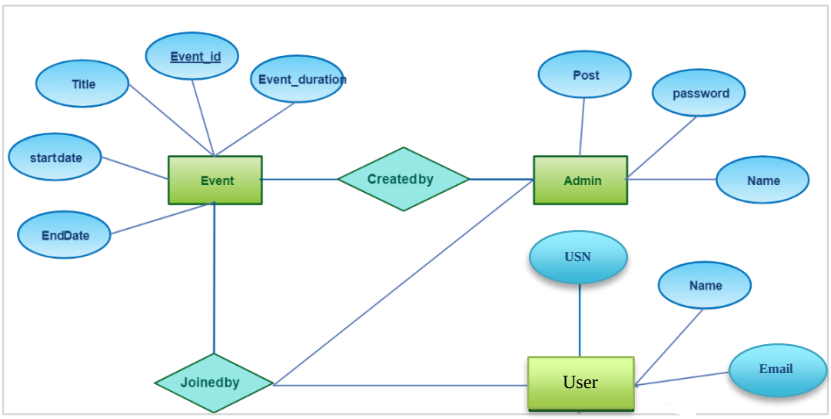


Fig.2.1.4 ER Diagram for Event Management System

4. Development:

Set Up Development Environment:

- Install necessary tools: Python, Django, SQLite, and a code editor like VS Code.
- Set up version control using Git and a repository on GitHub.

Frontend Development:

- HTML/CSS: Create the structure and styling for the web pages.
- JavaScript: Add interactivity to the web pages.

Backend Development:

- Django Setup: Create a new Django project and set up the SQLite database.
- User Authentication: Use Django's built-in authentication system for user login, registration, and roles (admin, event organizer, participant).
- Event Management: Create models for events, including fields like title, description, date, time, venue, and speaker. Develop views and templates for event creation, updating, and deletion.
- Registration Management: Develop functionality for users to register for events and track registrations. Send confirmation emails using Django's email functionality.
- Schedule Management: Display event schedules using Django templates and JavaScript for dynamic content.
- Feedback Collection: Create models and views for feedback collection and display aggregated feedback.

Database Development:

- Models: Define Django models for users, events, registrations, and feedback.
- Migrations: Use Django's migration system to create database tables.

5. Testing:

Unit Testing:

- Write unit tests for individual components and models.

Integration Testing:

- Test the interaction between different components (e.g., user registration and event registration).

System Testing:

- Perform end-to-end testing of the entire system.

User Acceptance Testing:

- Have end-users test the system to ensure it meets their requirements.

6. Deployment:

- Deploy the Django application to the production server.
- Configure the production database and any necessary environment variables.

7. Maintenance:

Bug Fixes:

- Address issues or bugs reported by users.

Feature Enhancements:

- Add new features or improve existing ones based on user feedback.

Performance Optimization:

- Optimize the system for better performance and scalability.

2.2 TOOLS AND TECHNOLOGIES

Here's a list of tools and technologies you can use to develop the Departmental Event Management System:

Frontend

- **HTML:** For structuring the web pages.
- **CSS:** For styling the web pages.
- **JavaScript:** For adding interactivity to the web pages.
- **Bootstrap/Tailwind CSS:** Optional frameworks used for responsive design and pre-built UI components.

Backend

- **Django:** The primary web framework for developing the backend of the application.

Database

- **SQLite:** A lightweight database used by Django by default for development and can be used for production for small-scale applications.

Development Tools

- **Python:** The programming language for writing the Django backend.
- **VS Code/PyCharm:** Code editors/IDEs with support for Python and web development.
- **Git:** Version control system to track changes in the codebase.
- **GitHub/GitLab/Bitbucket:** Repository hosting services for code collaboration and version control.

2.3 PROJECT TIMELINE

The project timeline for building a departmental event management system::

Week 1: Planning and Initial Setup

Project Planning

- Define project scope and requirements.
- Identify key features: event registration, scheduling, feedback collection.
- Design database schema (events, users, feedback).

Environment Setup

- Set up development environment (install Python, Django, SQLite).
- Initialize Django project and app.
- Configure settings and create initial database migrations.

Basic Models and Admin Interface

- Define models for events, users, and feedback in models.py.
- Register models with the Django admin site.
- Create initial superuser and test the admin interface.

Week 2: Event Registration and Scheduling

Event Registration

- Create views and templates for event registration.
- Implement forms for event registration using Django forms.
- Validate and save registration data to the database.

Event Scheduling

- Create views and templates for scheduling events.
- Implement logic to check for scheduling conflicts.
- Add event scheduling functionality to the admin interface.

Frontend Development

- Design basic HTML/CSS templates for event registration and scheduling pages.
- Ensure responsiveness and basic user experience design.

Week 3: Feedback Collection and User Authentication

Feedback Collection

- Create models for feedback.
- Develop views and templates for feedback submission.
- Implement feedback collection forms and validation.

User Authentication

- Implement user registration and login functionality using Django's built-in authentication system.
- Create user-specific views for managing their event registrations and feedback.

Integrate JavaScript for Dynamic Features

- Add JavaScript for form validation and dynamic content updates.
- Using AJAX to update schedules and feedback without reloading pages.

Week 4: Testing, Deployment, and Documentation

Testing

- Write unit tests for models, views, and forms.
- Perform integration testing to ensure all components work together.
- Conduct user acceptance testing to get feedback from potential users.

Deployment Preparation

- Prepare project for deployment (static files, settings adjustments).
- Choose a deployment platform (Heroku, AWS, etc.) and set up environment.
- Deploy the application and test in a live environment.

Documentation and Final Review

- Write documentation for installation, usage, and maintenance.
- Prepare a final project report and presentation.
- Conduct a final review and address any remaining issues.

CHAPTER 3: SYSTEM DESIGN AND IMPLEMENTATION

3.1 SYSTEM ARCHITECTURE

Created the overall system architecture, defining the interaction between the frontend, backend, and database components as shown below:

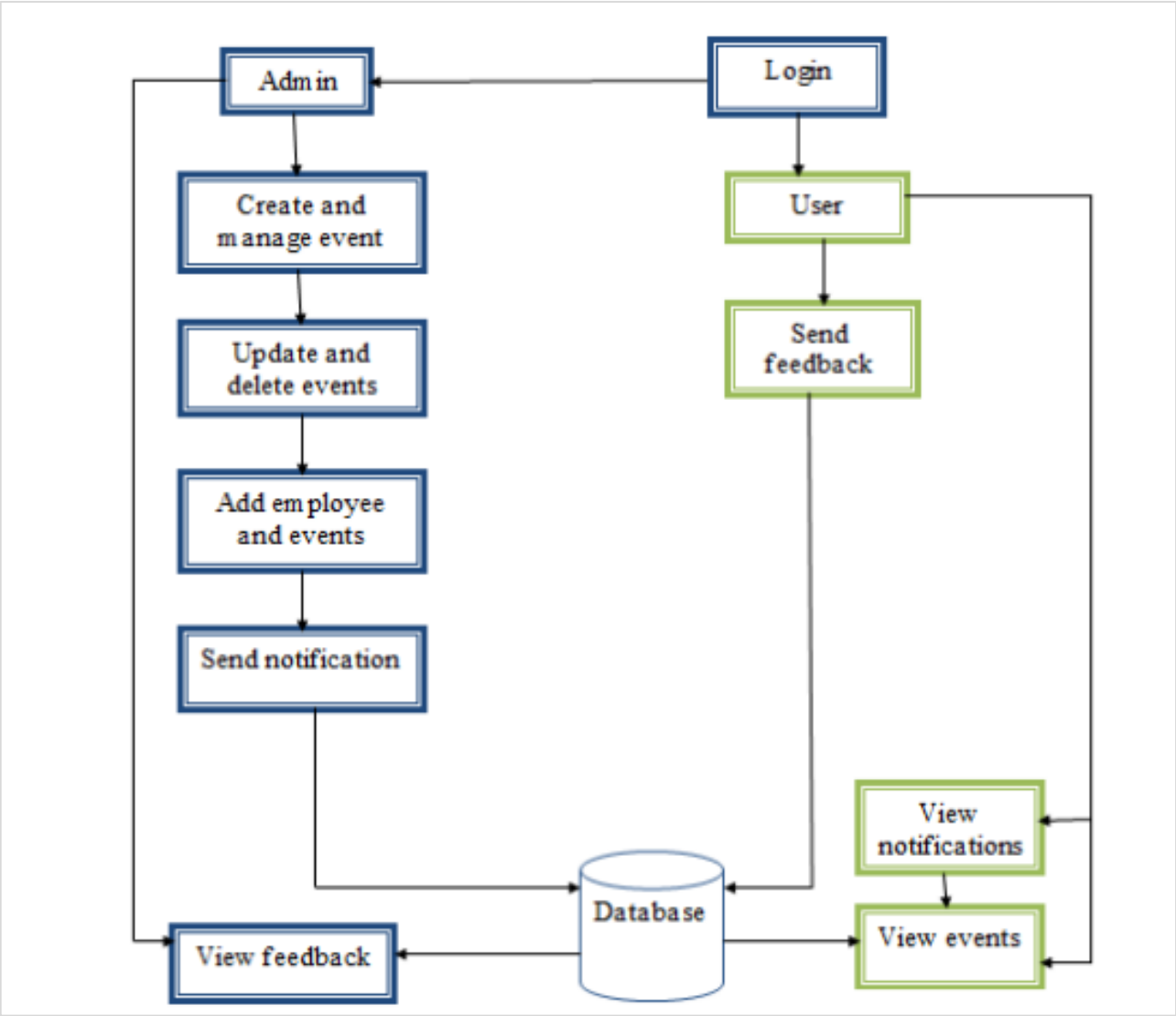


Fig 3.1: System Architecture

3.2 COMPONENT DESIGN

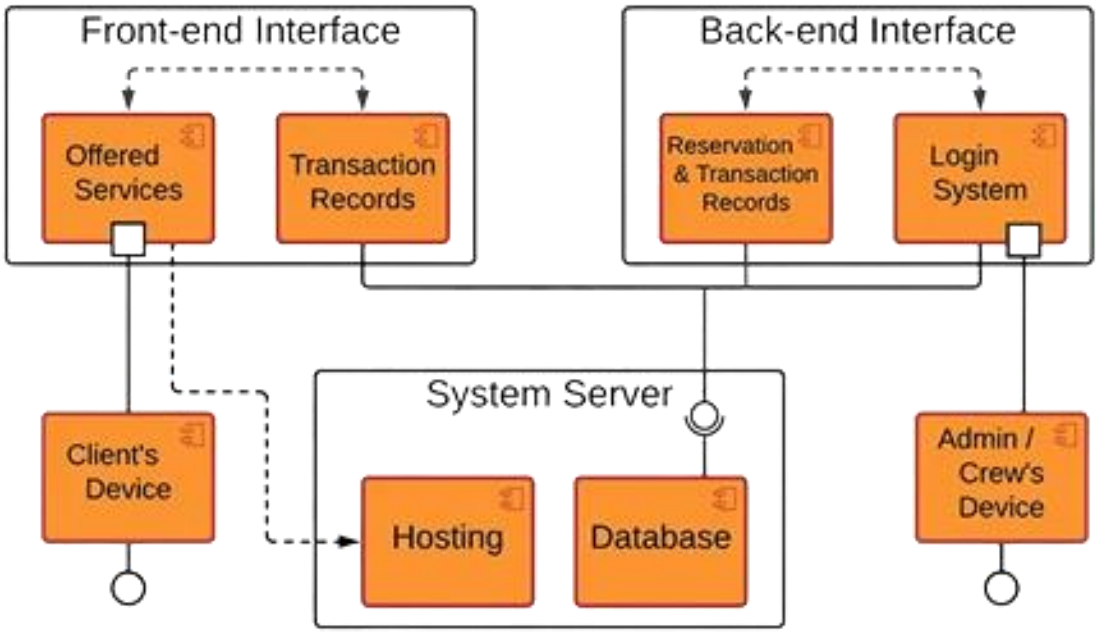


Fig 3.2: Component Design

3.3 IMPLEMENTATION DETAILS

The implementation details about departmental event management system:

Step 1: Setting Up the Development Environment

1. **Install Python**: Ensure Python is installed on your system.
2. **Install Django**: Use pip to install Django.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SQL CONSOLE
PS C:\Users\giris\OneDrive\Desktop\AA(Mini-Proj)\Event> pip install django
```

3. **Create a Django Project**: Initialize a new Django project.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SQL CONSOLE
PS C:\Users\giris\OneDrive\Desktop\AA(Mini-Proj)\Event> django-admin startproject event_management
```

4. **Create a Django App**: Create a new app within the project.

```
cd event_management
python manage.py startapp events
```

Step 2: Designing models and creating views & templates

Kavya_Project > app1 > models.py > ...

```python
# Create your models here.
from django.db import models

class Registration(models.Model):
    name = models.CharField(max_length=100)
    email = models.EmailField()
    event_type = models.CharField(max_length=50)
    registered_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f'{self.name} - {self.event_type}'
```

Kavya_Project > app1 > views.py > ...

```python
def registers(request):
        form = UserRegistrationForm()
    return render(request, 'register.html', {'form': form})

def user_login(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user is not None:
            login(request, user)
            return redirect('index')
        else:
            return render(request, 'login1.html', {'error': 'Invalid credentials'})
    return render(request, 'login1.html')
```

Step 3: Configuring URLs

Kavya_Project > app1 > urls.py > ...

```python
from django.contrib import admin
from django.urls import path

from app1.views import Appdev, cquest, index, register, home

urlpatterns = [
    path('admin/', admin.site.urls),
    path('register/',register),
    path('index/',index),
    path('home/',home),
    path('App/',Appdev),
    path('cquest/',cquest),
]
```

Kavya_Project > project > urls.py > ...

```python
from django.contrib import admin
from django.urls import  path
from app1.views import Appdev, appdev_reg, cquest, index, register, home, registers, user_login

urlpatterns = [
    path('admin/', admin.site.urls),
    path('register/',register),
    path('index/',index,name='index'),
    path('home/',home),
    path('App/',Appdev,name='App'),
    path('cquest/',cquest),
    path('appdev_reg/',appdev_reg,name='appdev_reg'),
    path('registers/',registers,name='register'),
    path('login/',user_login,name='login'),
]

```

Step 5: Running Migrations and development server

Migrate and start the Django development server and visit http://127.0.0.1:8000/events/ to see the application in action.

```
python manage.py makemigrations
python manage.py migrate
```

```
python manage.py runserver
```

CHAPTER 4: RESULTS AND DISCUSSION

4.1 PRESENTATION OF RESULTS

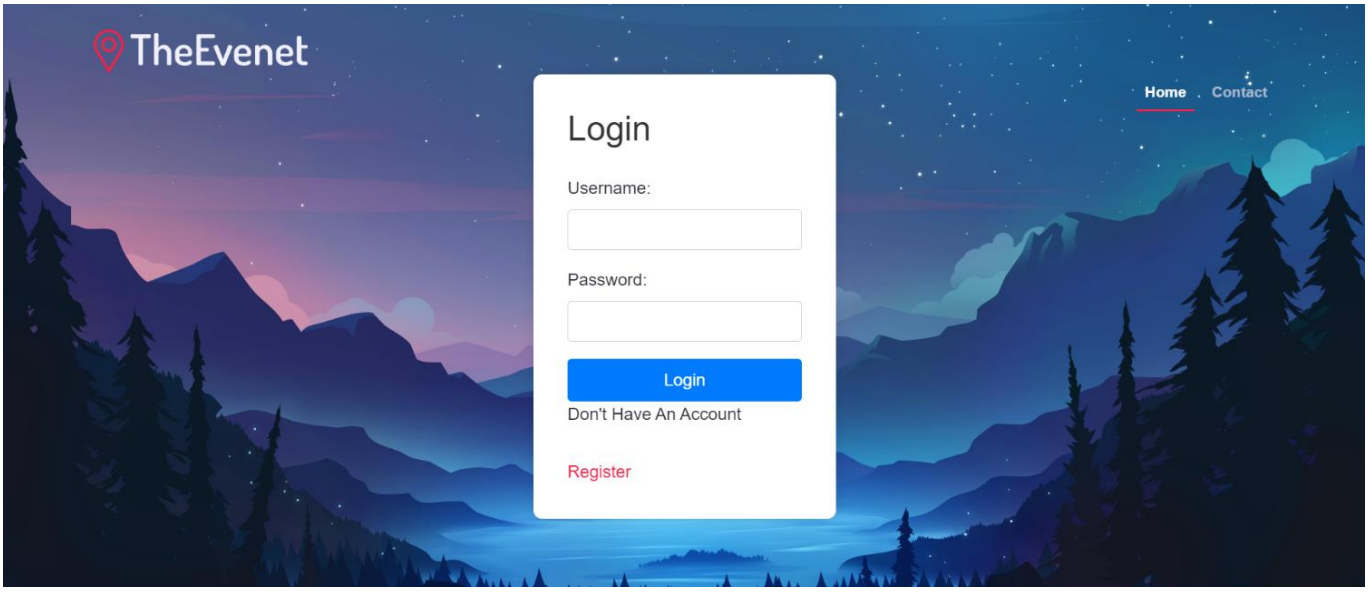


Fig. 4.1.1 Login Page

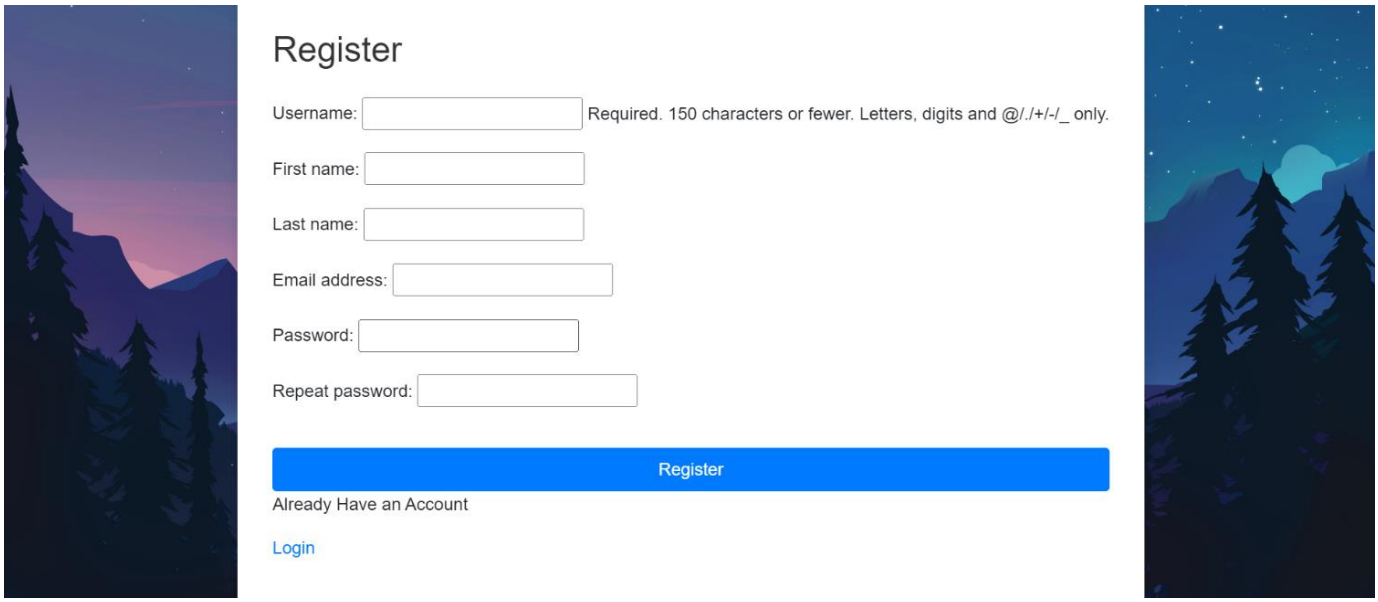


Fig.4.1.2 Register Page

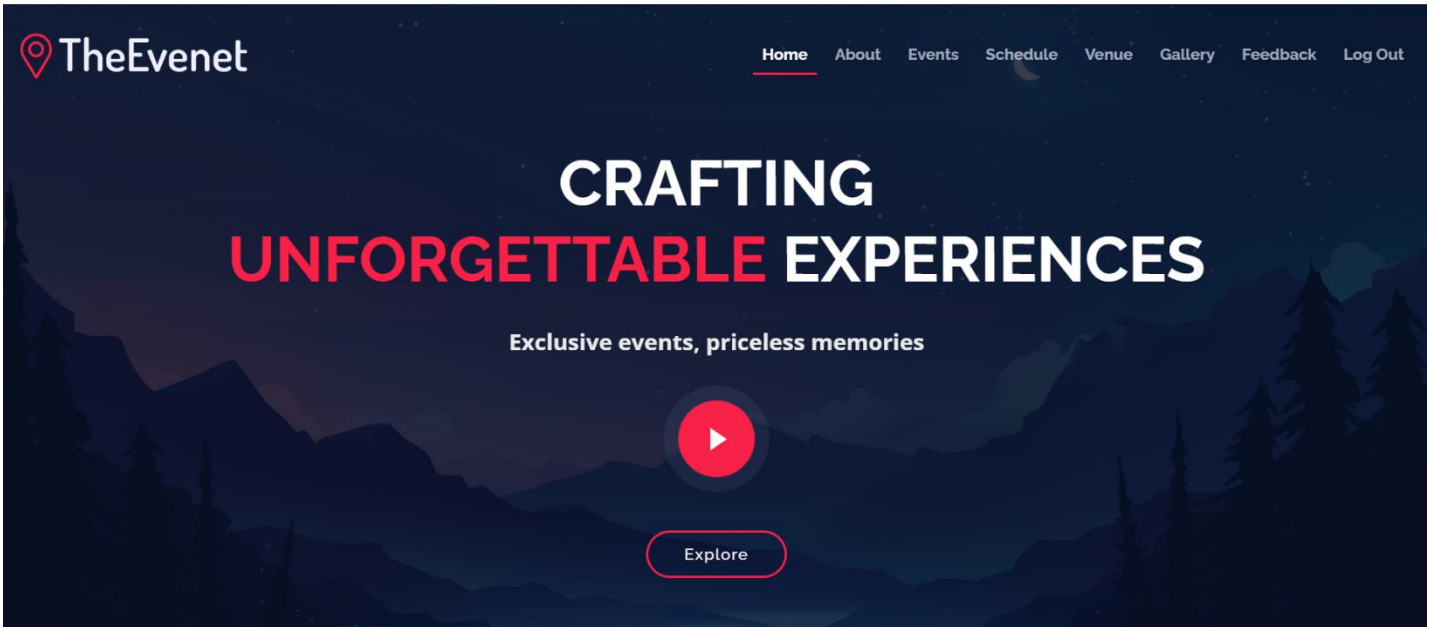


Fig.4.1.3 Home Page

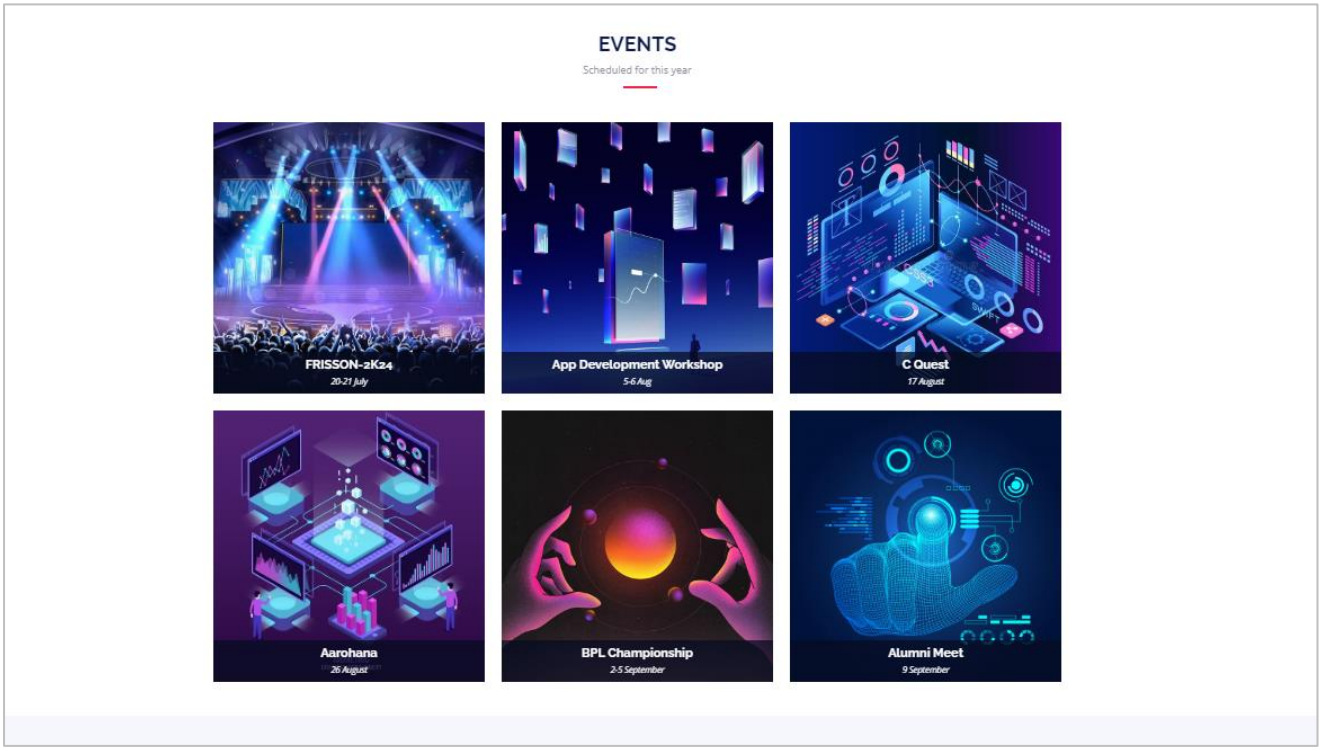


Fig. 4.1.4 Events List

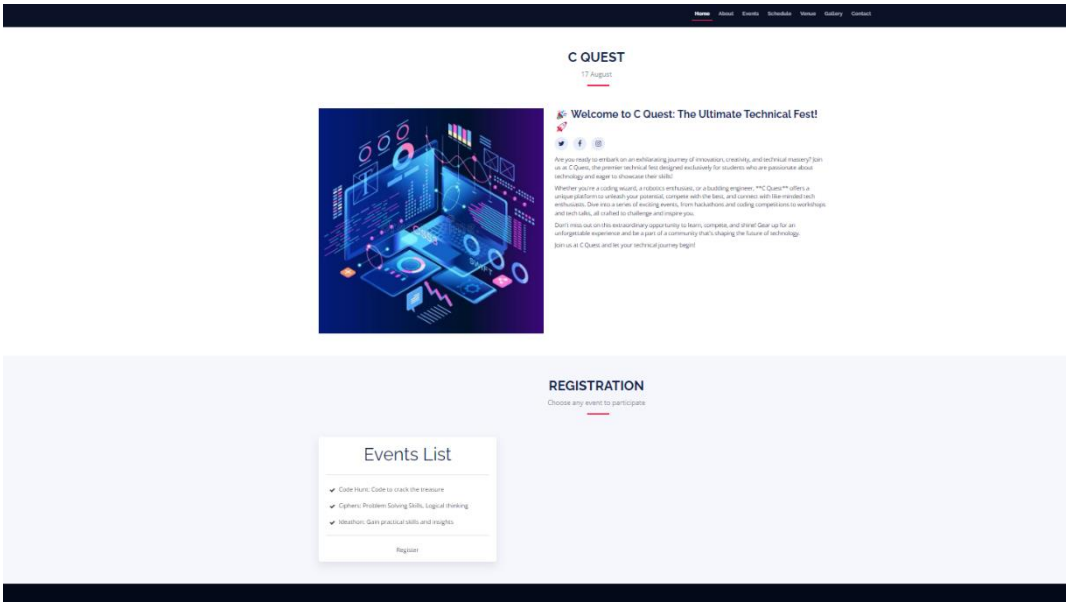


Fig.4.1.5 Event Selection Page

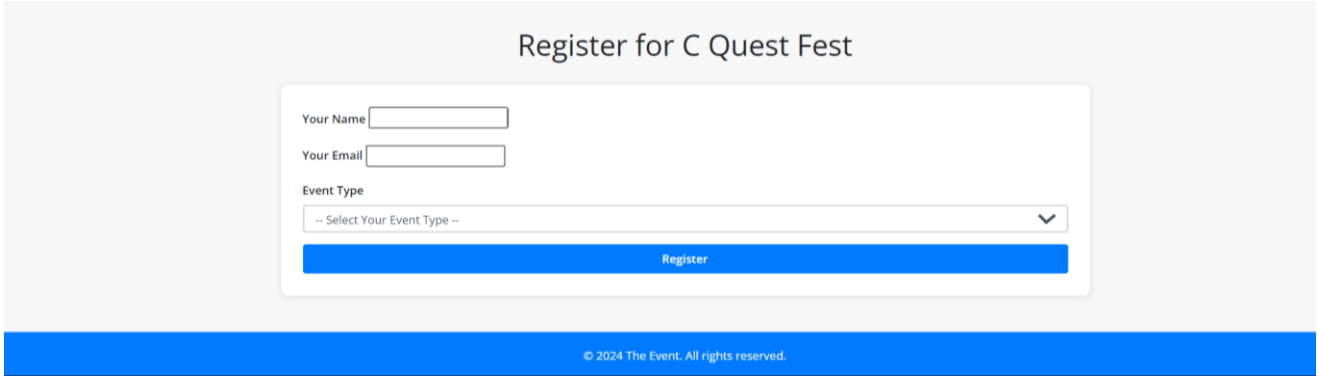


Fig. 4.1.6 Event Registration Page

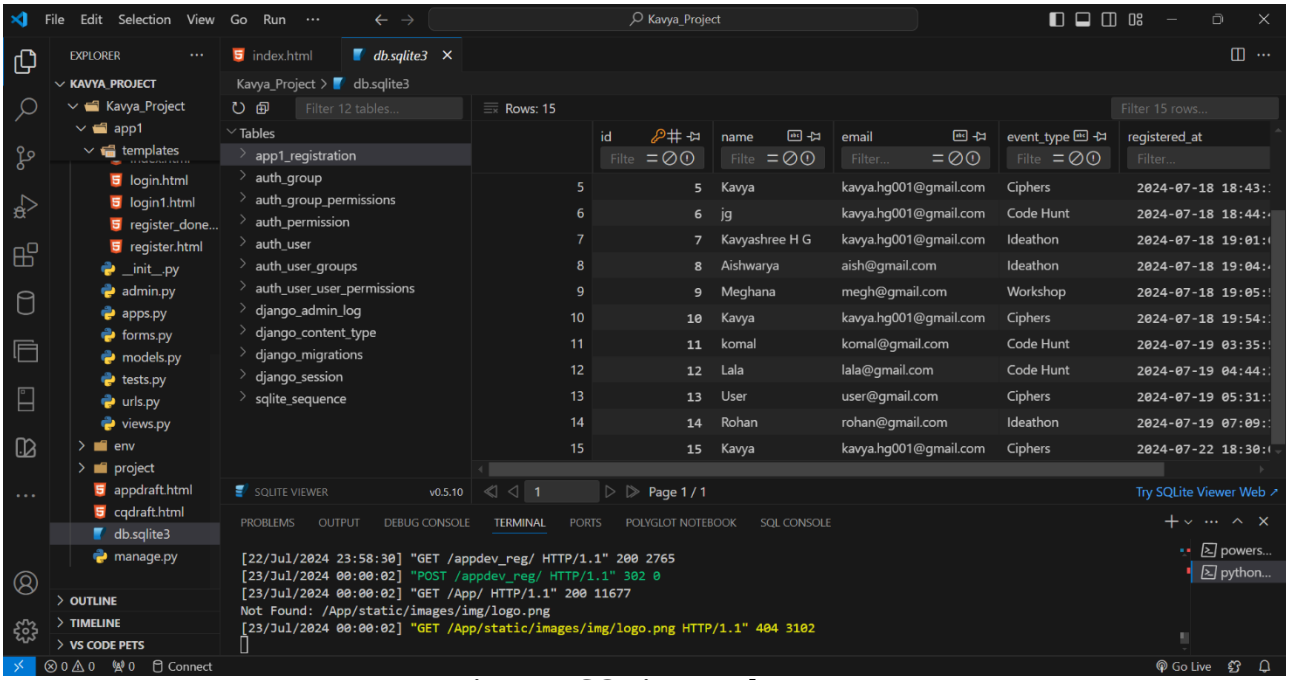


Fig.4.1.7 SQLite Database

4.2 ANALYSIS OF RESULTS

The Departmental Event Management System is designed to streamline the organization and management of departmental events, workshops, and seminars. The system facilitates event creation, registration, scheduling, and feedback collection, providing a centralized platform for all event-related activities. By leveraging Django for backend development and SQLite for data management, the system aims to enhance efficiency, data integrity, and user experience. Below are the advantages and disadvantages of implementing this system.

ADVANTAGES:

- ♣ Centralized control simplifies event creation, updating, and deletion, streamlined registration and feedback processes save time and effort.
- ♣ Integrity of data transactions is maintained.
- ♣ Scalable to accommodate more users and events without significant additional costs.

DISADVANTAGES:

- ♣ Initial implementation may lack advanced analytics, search, and filtering options, absence of user authentication limits personalized experiences and security.
- ♣ Requires a stable internet connection to function, no offline access or functionality is available.

4.3 COMPARISON WITH EXPECTATIONS

The Departmental Event Management System met most expectations by simplifying event management, providing an intuitive interface, and ensuring reliable data management. Feedback collection and cost-effectiveness through open-source tools were also successful. However, it fell short in advanced features like analytics and user authentication, experienced performance issues with SQLite under heavy loads, required a stable internet connection, and needed better security measures. User adaptation involved a learning curve and some resistance from traditional methods. Overall, while the system met key goals, it has areas for enhancement in features, performance, and security.

CONCLUSION

5.1 SUMMARY OF FINDINGS

The Departmental Event Management System effectively streamlined event management processes, offering a user-friendly interface and reliable data handling. The system successfully facilitated event creation, registration, and feedback collection, and proved cost-effective by utilizing open-source tools. However, several areas for improvement were identified. The system's initial limitations included a lack of advanced features such as detailed analytics and user authentication, performance constraints with SQLite under high loads, and dependency on stable internet connectivity. Additionally, security measures and user adaptation posed challenges. Addressing these issues through future enhancements will improve system functionality, performance, and user experience, ensuring the system better meets its goals and user needs.

5.2 LIMITATIONS

Despite its advantages, the implemented event management system has several limitations:

1. **Limited Advanced Features**: Lacks detailed analytics and advanced search options.
2. **Performance Constraints**: SQLite may struggle with high loads and large datasets.
3. **Dependency on Internet Connectivity**: Requires a stable internet connection; no offline access.
4. **Security Concerns**: Initial setup lacks advanced security features.
5. **User Authentication**: No user accounts for personalized experiences.
6. **Learning Curve**: Users may face challenges adapting from traditional methods.

5.3 FUTURE WORK

Future improvements include adding advanced features such as analytics and search options, upgrading to PostgreSQL for better performance, enabling offline access, enhancing security with encryption and user authentication, and providing user training for smoother adaptation. Enhancing security with data encryption and user authentication will protect sensitive information and ensure compliance.

REFERENCES

1. **"Django for Professionals: Production websites with Python & Django"** by William S. Vincent
    - A comprehensive guide to building professional websites using Django.
2. **"Two Scoops of Django 3.x: Best Practices for the Django Web Framework"** by Audrey Roy Greenfeld and Daniel Roy Greenfeld
    - Offers best practices and tips for working with Django.
3. **"Web Development with Django"** by Ben Shaw and Saurabh Badhwar
    - A practical book on web development using Django.
4. **"The Definitive Guide to Django: Web Development Done Right"** by Adrian Holovaty and Jacob Kaplan-Moss
    - Authored by the creators of Django, this book provides a deep dive into the framework's architecture and best practices.
5. **Stack Overflow**
   https://stackoverflow.com/questions/tagged/django
6. **GitHub - Django Repositories**
   https://github.com/django/django