EventConnect-Sponsor and Organiser Platform

A Project Report

Submitted by

Vedang Dharmadhikari	**112103036**
Soham Dorage	**112103040**
Aryan Thakre	**112103036**

in partial fulfilment for the award of the degree

of

TY (Computer Engineering))/

Under the guidance of

Dr. Tanuja R. Pattanshetti

COEP Technological University

DEPARTMENT OF COMPUTER ENGINEERING AND INFORMATION TECHNOLOGY,

COEP Technological University

CERTIFICATE

Certified that this project, titled “Event-Connect-Sponsor and Organiser Platform” has been successfully completed by

Vedang Dharmadhikari	**112103036**
Soham Dorage	**112103040**
Aryan Thakare	**112103014**

and is approved for the partial fulfilment of the requirements for the degree of “Software Engineering Mini Project”.

SIGNATURE	SIGNATURE
Tanuja Pattenshetti	**Pradeep Deshmukh**
Project Guide	**Head**
Department of Computer Engineering	**Department of Computer Engineering**
and Information Technology,	**and Information Technology,**

COEP Technological University,

Shivajinagar, Pune - 5.

COEP Technological Univesity,

Shivajinagar, Pune - 5.

Abstract

EventConnect is a dynamic web-based platform designed to bridge the gap between event sponsors and organizers, facilitating seamless connections and fostering mutually beneficial partnerships within the event industry. The existing challenges in this sector often hinder efficient collaboration between sponsors and organizers, necessitating a dedicated solution to streamline the process.

Through EventConnect, sponsors gain access to a tailored dashboard featuring event listings categorized by domain and price filters, along with a comprehensive profile form to specify their details, price range, and domain preferences. On the other hand, event organizers benefit from a dashboard displaying sponsor listings based on relevant criteria, as well as a profile form to outline their event details, price range, domain, and target audience.

Key features of the platform include the ability for organizers to create events with detailed information, including target audience demographics, and for sponsors to easily find events aligned with their objectives. Upon expressing interest in an event, organizers receive notifications, initiating further discussions via private chat rooms. The platform also facilitates feedback and rating systems, enabling sponsors to provide valuable insights on events and vice versa.

Built using the Python programming language and the Flask web development framework, EventConnect ensures a robust and scalable solution. HTML, CSS, and JavaScript are employed to create a user-friendly interface, enhancing the overall user experience.

In summary, EventConnect aims to revolutionize the event industry by providing a centralized platform for sponsors and organizers to connect, collaborate, and cultivate successful partnerships, ultimately fostering growth and innovation within the sector.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

In today's fast-paced world, the event industry serves as a crucial platform for businesses, organizations, and communities to connect, engage, and thrive. Events, ranging from conferences and trade shows to festivals and networking meetups, play a pivotal role in fostering relationships, sharing knowledge, and driving innovation across various sectors. Central to the success of any event are the partnerships forged between event sponsors and organizers, which provide essential resources, support, and visibility necessary for event execution and success.

However, despite the critical role of sponsor-organizer partnerships, the event industry often grapples with challenges in effectively connecting sponsors with organizers and vice versa. Existing methods of sponsor-organizer collaboration, such as manual outreach, networking events, and third-party agencies, are often inefficient, fragmented, and lack scalability. As a result, both sponsors and organizers face difficulties in identifying suitable partners, negotiating partnership agreements, and maximizing the mutual benefits of collaboration.

Moreover, in today's digital age, where data-driven insights and real-time communication are paramount, the lack of centralized platforms for sponsor-organizer connections presents a significant hurdle for the industry. Event sponsors and organizers struggle to access relevant data, analyze market trends, and communicate seamlessly with potential partners, leading to missed opportunities, suboptimal partnerships, and inefficient resource allocation.

To address these challenges and unlock the full potential of sponsor-organizer partnerships, we propose the development of EventConnect, a comprehensive Cloud-Based Platform designed to bridge the gap between event sponsors and organizers. EventConnect leverages the power of cloud technologies, data analytics, and real-time communication tools to streamline the process of sponsor-organizer connections, enabling efficient, transparent, and mutually beneficial partnerships within the event industry.

By providing a centralized platform for sponsor-organizer collaboration, EventConnect aims to revolutionize the way events are planned, executed, and experienced. Event sponsors gain access to a diverse range of event opportunities tailored to their preferences, objectives, and target audiences, while event organizers benefit from increased visibility, engagement, and support from sponsors aligned with their goals and values.

In essence, EventConnect represents more than just a platform—it embodies a vision of a collaborative, interconnected event ecosystem where sponsors and organizers come together to create impactful, memorable experiences that drive positive change and inspire communities. Through innovation, technology, and partnership, EventConnect seeks to empower the event industry and unleash its full potential for growth, creativity, and impact.

1.2 Aim of the Project

The aim of the EventConnect project is to develop a dedicated web-based platform that bridges the gap between event sponsors and organizers, facilitating efficient connections and fostering mutually beneficial partnerships within the event industry. By providing a centralized solution, the project aims to streamline the process of finding and engaging with sponsors or events, thereby enhancing collaboration, maximizing opportunities, and driving growth within the sector.

1.3 Methodology

The methodology for developing the EventConnect platform involves several key steps:

1.3.1 Requirement Analysis

Conduct a comprehensive analysis of the requirements by gathering insights from stakeholders including event sponsors, organizers, and potential users. Define the core features, functionalities, and user workflows based on identified needs and pain points within the event industry.

1.3.2 Design Phase

- **UI/UX Design**: Create wireframes and mockups to visualize the user interface and user experience of the platform. Design intuitive layouts, navigation systems, and interactive elements to ensure ease of use and engagement.

- **Database Design**: Design the database schema using MySQL, defining tables, relationships, and data attributes to efficiently store and manage user, event, and sponsorship data.

1.3.3 Development

- **Backend Development**: Utilize Python programming language and choose a suitable web framework such as Flask to develop the backend logic of the platform. Implement functionalities for user authentication, profile management, event creation, sponsorship listing, notifications, and messaging.
- **Frontend Development**: Implement the UI designs using HTML, CSS, and JavaScript to create responsive and visually appealing interfaces. Integrate frontend components with backend APIs to enable dynamic interactions and seamless user experiences.

1.3.4 Testing

- **Unit Testing**: Perform unit tests to ensure individual components and functionalities function as expected.
- **Integration Testing**: Test the integration of frontend and backend components to verify data flow and system interactions.

1.3.5 Maintenance and Iteration

- Monitor platform performance, user feedback, and analytics to identify areas for improvement.
- Implement iterative updates and enhancements based on user feedback, emerging trends, and industry requirements.
- Provide ongoing maintenance, support, and bug fixes to ensure the platform's stability and functionality.

By following this methodology, the EventConnect platform can be developed systematically, ensuring it meets the needs of its users while maintaining scalability, reliability, and usability.

1.4 Significance of the Work

The significance of the EventConnect project lies in its potential to revolutionize the event industry by addressing key challenges and enhancing the efficiency and effectiveness of sponsor-organizer partnerships. The following points highlight the significance of this work:

- **Efficiency Improvement**: By providing a centralized platform for sponsor-organizer connections, EventConnect streamlines the process of finding, engaging, and collaborating with sponsors or events. This efficiency improvement saves time and resources for both sponsors and organizers, enabling them to focus more on the strategic aspects of event planning and sponsorship management.

- **Opportunity Expansion**: EventConnect expands the opportunities for sponsors and organizers to connect with relevant partners beyond their immediate networks. By leveraging data-driven matchmaking algorithms and personalized recommendations, the platform facilitates meaningful connections between sponsors and events that align with their objectives, target audiences, and values.

- **Transparency and Trust**: The transparency and accountability built into EventConnect through feedback and rating systems foster trust and confidence among sponsors and organizers. By providing visibility into past partnerships and performance metrics, the platform enables informed decision-making and mitigates the risks associated with partnering with unknown entities.

- **Innovation Catalyst**: EventConnect serves as a catalyst for innovation within the event industry by promoting collaboration, creativity, and experimentation. By facilitating connections between sponsors and organizers from diverse backgrounds and industries, the platform encourages the exchange of ideas, best practices, and resources, leading to the development of innovative event experiences and formats.
- **Industry Growth**: By enabling more efficient and effective sponsor-organizer partnerships, EventConnect contributes to the overall growth and sustainability of the event industry. The platform empowers event organizers, particularly smaller or niche-focused ones, to access sponsorship opportunities and resources that may have been previously out of reach, thus fueling innovation and diversity within the industry.

In summary, the significance of the EventConnect project extends beyond the confines of individual events or sponsor-organizer partnerships. By providing a scalable, reliable, and user-friendly platform for connecting sponsors and organizers, EventConnect has the potential to transform the event industry, driving efficiency, innovation, and growth for years to come.

1.5 List of Abbreviations

EC EventConnect

DB Database

UI User Interface

UX User Experience

API Application Programming Interface

HTML Hypertext Markup Language

CSS Cascading Style Sheets

JS JavaScript

UAT User Acceptance Testing

SSL Secure Sockets Layer

CRUD Create, Read, Update, Delete

FL Flask

PDM Patient Data Management

FR Functional Requirement

TC Test Case

Chapter 2

Previous Solutions

2.1 Fragmented Sponsor-Organizer Connections

Previous approaches in the event industry have struggled to facilitate efficient connections between sponsors and organizers, leading to several challenges:

- **Lack of Centralized Platform**: Existing methods often relied on fragmented channels such as emails, phone calls, or personal networks, making it difficult for sponsors and organizers to find suitable partners.
- **Inefficient Communication**: Without a dedicated platform, communication between sponsors and organizers was often slow and disjointed, hindering the partnership development process.
- **Limited Visibility**: Sponsors and organizers had limited visibility into potential partnership opportunities, resulting in missed connections and suboptimal collaborations.

2.2 Manual Search and Engagement

Previous practices involved manual search and engagement efforts, which were prone to various shortcomings:

- **Time-Consuming Process**: Searching for suitable sponsors or events manually was a time-consuming process, requiring significant effort from both sponsors and organizers.
- **Difficulty in Targeting**: Without tailored search filters, sponsors and organizers struggled to find partners that matched their specific criteria such as domain, price range, and target audience.
- **Limited Reach**: Manual search methods often resulted in a limited reach, as sponsors and organizers were unable to explore a wide range of potential partnerships beyond their immediate networks.

2.3 Lack of Feedback Mechanisms

Previous approaches lacked effective feedback mechanisms, posing challenges for improving future interactions:

- **Absence of Evaluation**: Without a structured feedback system, sponsors and organizers had limited opportunities to provide feedback on their experiences, hindering the improvement of the partnership process.
- **Difficulty in Assessing Performance**: The absence of feedback mechanisms made it challenging for sponsors and organizers to assess the success of their partnerships and identify areas for improvement.

- **Missed Learning Opportunities**: Without actionable feedback, sponsors and organizers missed out on valuable learning opportunities that could drive more successful collaborations in the future.

Chapter 3

Actual Work Done

3.1 Username and Password Login Page

The username and password login page serves as the entry point for users to access the EventConnect platform. Below are the key features and functionalities implemented in this login page:

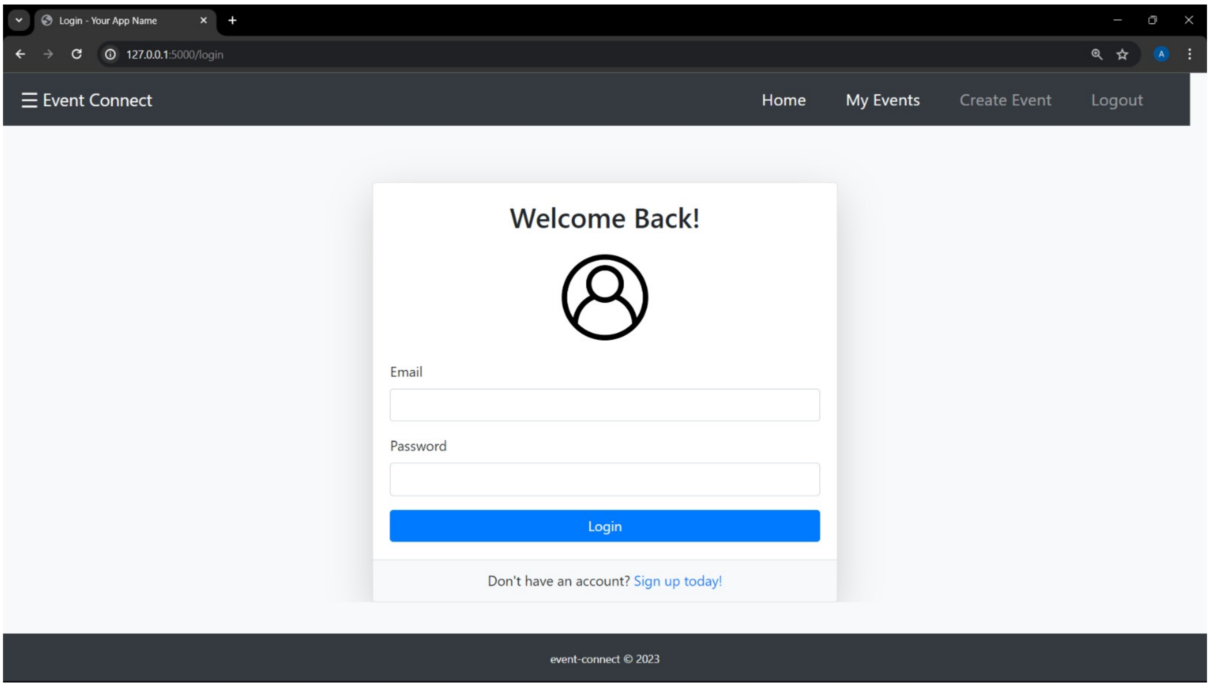


Figure 3.1: Username and Password Login Page

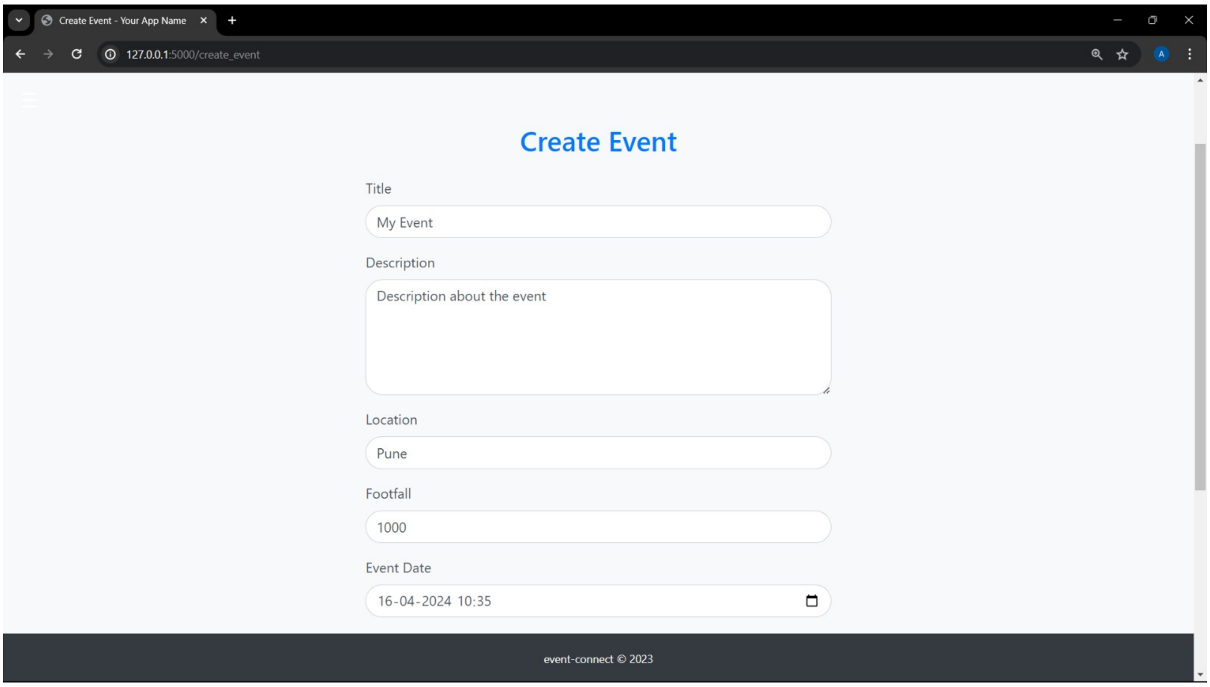
- **User Authentication:** Implemented a secure authentication system to verify the identity of users before granting access to the platform.

- **Username Input Field:** Provided a text input field where users can enter their unique usernames.
- **Password Input Field:** Implemented a password input field to ensure confidentiality, where users can securely input their passwords.
- **Login Button:** Integrated a login button that triggers the authentication process when clicked by the user.
- **Error Handling:** Implemented error handling mechanisms to display appropriate error messages in case of incorrect username or password inputs.
- **Session Management:** Implemented session management to maintain user sessions securely and ensure seamless navigation within the platform.

3.2 Event Creation

The event creation feature allows organizers to create and publish events on the Event-Connect platform. Below are the key features and functionalities implemented in this section:

- **Event Details Form:** Provided a form where organizers can input event details such as event name, date, location, description, target audience, and domain.
- **Price Range Selection:** Included options for organizers to specify the price range for tickets or sponsorship packages.
- **Validation and Error Handling:** Implemented validation checks to ensure all required fields are filled out correctly, with error messages displayed for any invalid inputs.



Create Event

Title

My Event

Description

Description about the event

Location

Pune

Footfall

1000

Event Date

16-04-2024 10:35

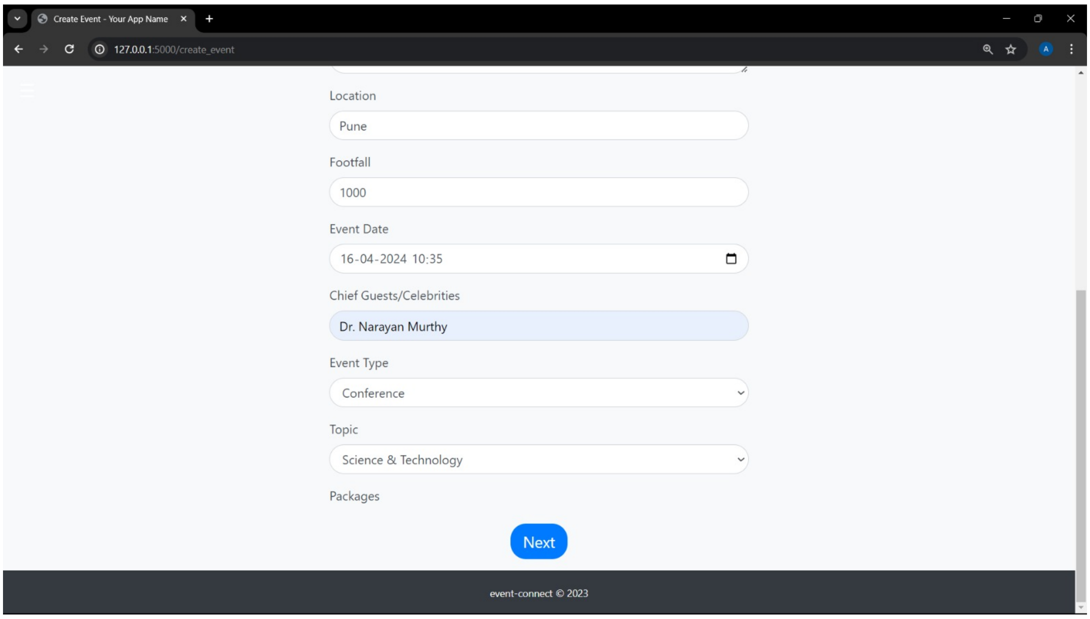
event-connect © 2023

Figure 3.2: Event Creation Interface

3.3 Event Listing and Filtering

The event listing and filtering functionality allow users to browse and filter events based on their preferences. Below are the key features and functionalities implemented in this section:

- **Event Listing:** Displays a list of events available on the platform, including details such as event name, date, location, and organizer.
- **Filtering Options:** Provides filtering options such as event domain and price range to help users narrow down their search results.
- **Search Bar:** Includes a search bar where users can enter keywords to find specific events quickly.
- **Sorting:** Allows users to sort the list of events based on parameters such as date, price, or popularity.



Location: Pune

Footfall: 1000

Event Date: 16-04-2024 10:35

Chief Guests/Celebrities: Dr. Narayan Murthy

Event Type: Conference

Topic: Science & Technology

Packages

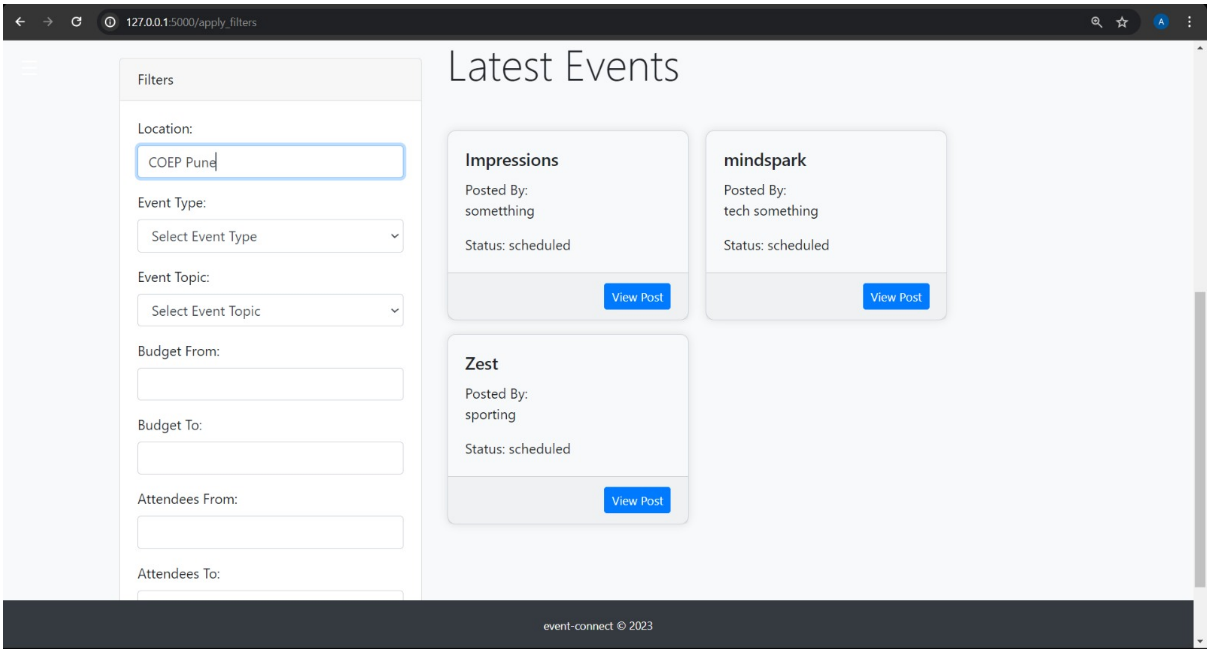
Figure 3.3: Event Creation Interface

- **Interactive Interface:** Offers an interactive interface for users to explore event details, view event organizers' profiles, and express interest in attending events.

3.4 Sponsor Listing

The sponsor listing feature allows organizers to showcase available sponsorship opportunities, while sponsors can browse and select events for sponsorship. Below are the key features and functionalities implemented in this section:

- **Sponsorship Packages:** Presented various sponsorship packages offered by event organizers, including details such as sponsorship tiers, benefits, and pricing.
- **Domain and Price Filters:** Provided filters for sponsors to refine their search based on sponsorship domain and price range.
- **Interactive Display:** Displayed sponsor listings in a visually appealing and informative manner, highlighting key sponsorship details and benefits.
- **Express Interest:** Enabled sponsors to express interest in sponsoring specific



127.0.0.1:5000/apply_filters

Filters

Location: COEP Pune

Event Type: Select Event Type

Event Topic: Select Event Topic

Budget From:

Budget To:

Attendees From:

Attendees To:

Latest Events

Impressions
Posted By:
somethting
Status: scheduled
View Post

mindspark
Posted By:
tech something
Status: scheduled
View Post

Zest
Posted By:
sporting
Status: scheduled
View Post

event-connect © 2023

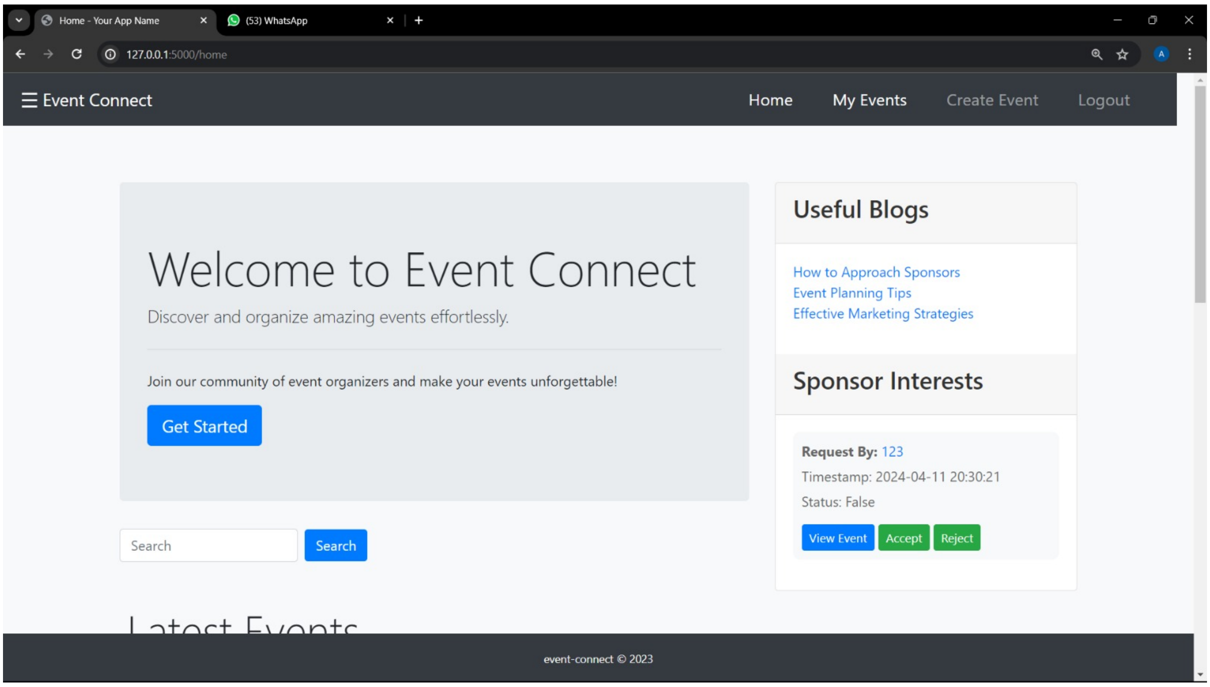
Figure 3.4: Event Listing and Filtering

events directly through the platform, initiating further communication with event organizers.

3.5 Signup Functionality

The signup functionality serves as the entry point for new users to register on the Event-Connect platform. Below are the key features and functionalities implemented in the signup process:

- **User Registration Form:** Implemented a user-friendly registration form where new users can input their details such as username, email address, password, and preferred domain.
- **Validation Checks:** Incorporated validation checks to ensure that the entered information meets the required criteria, such as password strength and unique username/email.
- **Registration Submission:** Integrated a submission mechanism to process the



Welcome to Event Connect

Discover and organize amazing events effortlessly.

Join our community of event organizers and make your events unforgettable!

Request By: 123

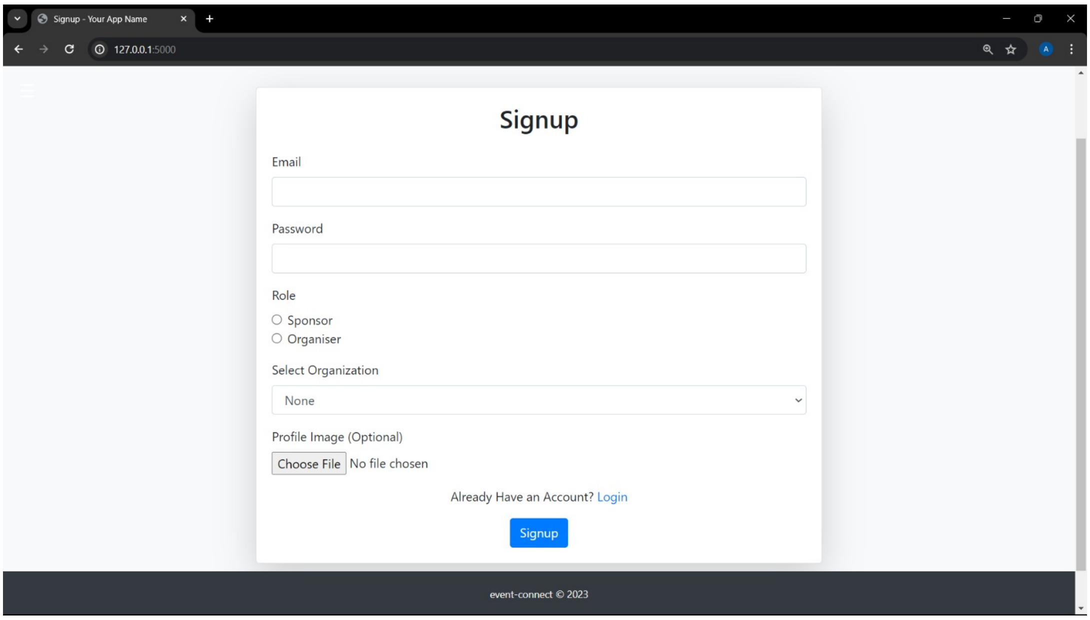
Timestamp: 2024-04-11 20:30:21

Status: False

Figure 3.5: Sponsor Listing Interface

registration form data and create new user accounts in the database.

- **Redirection to Login Page:** After registration, users are automatically redirected to the login page to access their newly created accounts.



Signup

Email

Password

Role

○ Sponsor

○ Organiser

Select Organization

None

Profile Image (Optional)

Choose File No file chosen

Already Have an Account? Login

Figure 3.6: Signup Page

Chapter 4

Project Management and Documentation

4.1 Gantt Chart

A Gantt chart is a visual representation of the project schedule, outlining tasks, dependencies, and timelines. It helps in planning, scheduling, and tracking project progress.

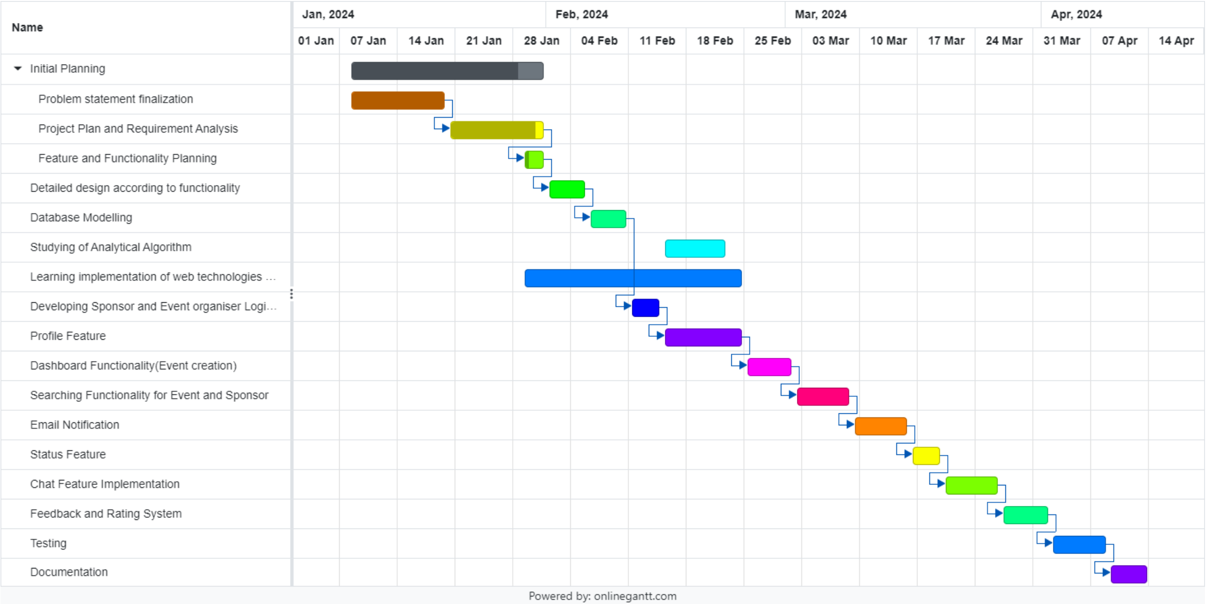
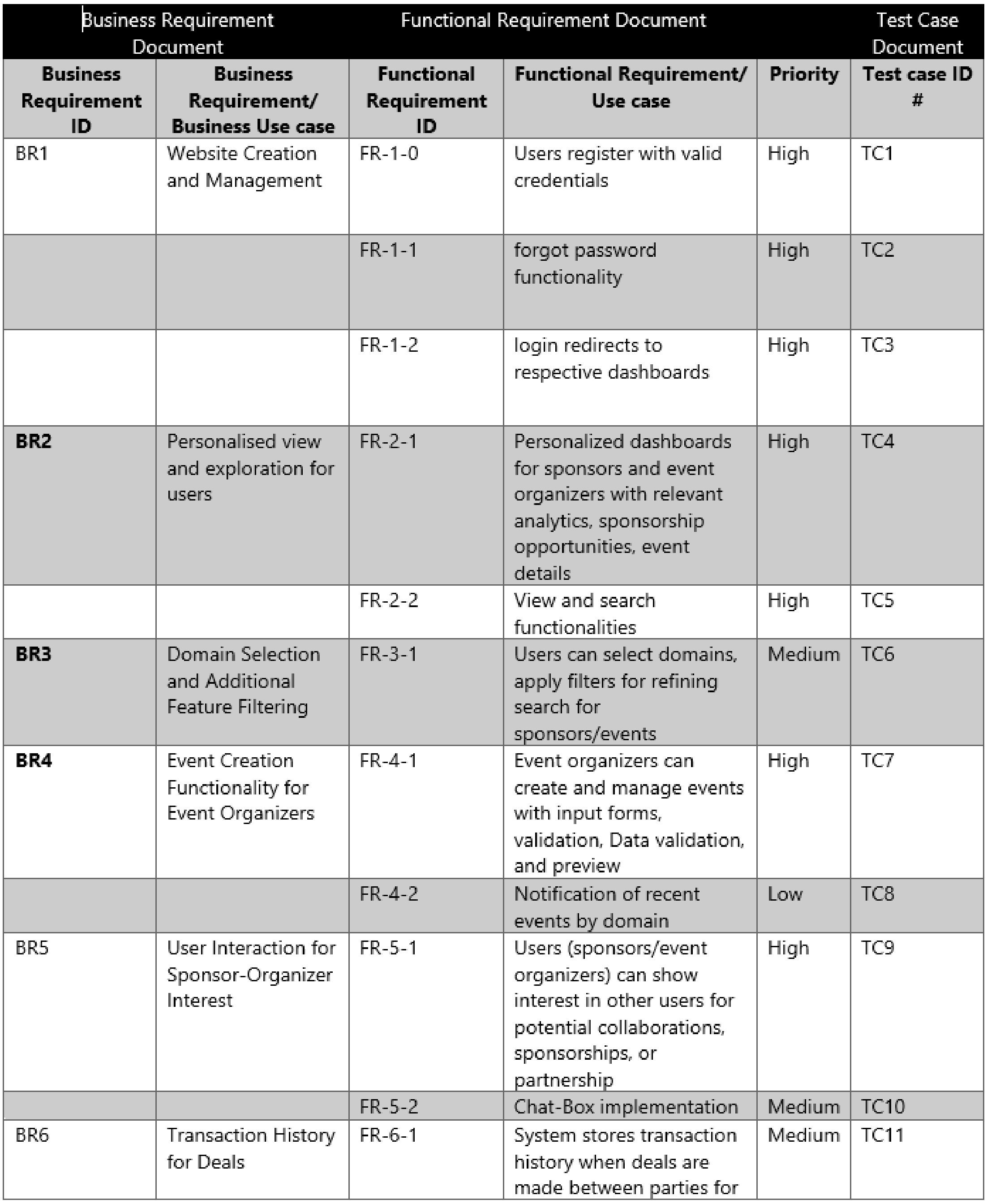


Figure 4.1: Gantt Chart for Project Plan

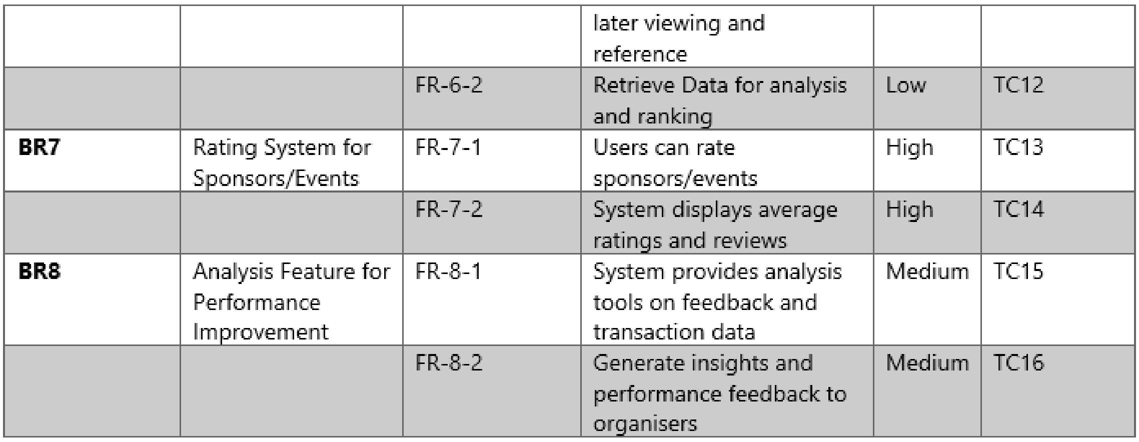
4.2 Traceability Matrix

The traceability matrix maps requirements to various project artifacts, such as design documents, test cases, and source code. It ensures that all requirements are adequately addressed and tested throughout the project lifecycle. The traceability matrix for the project is provided in Table .



Business Requirement Document		Functional Requirement Document			Test Case Document
Business Requirement ID	**Business Requirement/ Business Use case**	**Functional Requirement ID**	**Functional Requirement/ Use case**	**Priority**	**Test case ID #**
BR1	Website Creation and Management	FR-1-0	Users register with valid credentials	High	TC1
		FR-1-1	forgot password functionality	High	TC2
		FR-1-2	login redirects to respective dashboards	High	TC3
BR2	Personalised view and exploration for users	FR-2-1	Personalized dashboards for sponsors and event organizers with relevant analytics, sponsorship opportunities, event details	High	TC4
		FR-2-2	View and search functionalities	High	TC5
BR3	Domain Selection and Additional Feature Filtering	FR-3-1	Users can select domains, apply filters for refining search for sponsors/events	Medium	TC6
BR4	Event Creation Functionality for Event Organizers	FR-4-1	Event organizers can create and manage events with input forms, validation, Data validation, and preview	High	TC7
		FR-4-2	Notification of recent events by domain	Low	TC8
BR5	User Interaction for Sponsor-Organizer Interest	FR-5-1	Users (sponsors/event organizers) can show interest in other users for potential collaborations, sponsorships, or partnership	High	TC9
		FR-5-2	Chat-Box implementation	Medium	TC10
BR6	Transaction History for Deals	FR-6-1	System stores transaction history when deals are made between parties for	Medium	TC11

Figure 4.2: Traceability Matrix



			later viewing and reference		
		FR-6-2	Retrieve Data for analysis and ranking	Low	TC12
BR7	Rating System for Sponsors/Events	FR-7-1	Users can rate sponsors/events	High	TC13
		FR-7-2	System displays average ratings and reviews	High	TC14
BR8	Analysis Feature for Performance Improvement	FR-8-1	System provides analysis tools on feedback and transaction data	Medium	TC15
		FR-8-2	Generate insights and performance feedback to organisers	Medium	TC16

Figure 4.3: Traceability Matrix

4.3 Data Flow Diagrams

Data flow diagrams (DFDs) illustrate the flow of data within the system, including inputs, processes, data storage, and outputs. They provide a graphical representation of how data moves through the system.

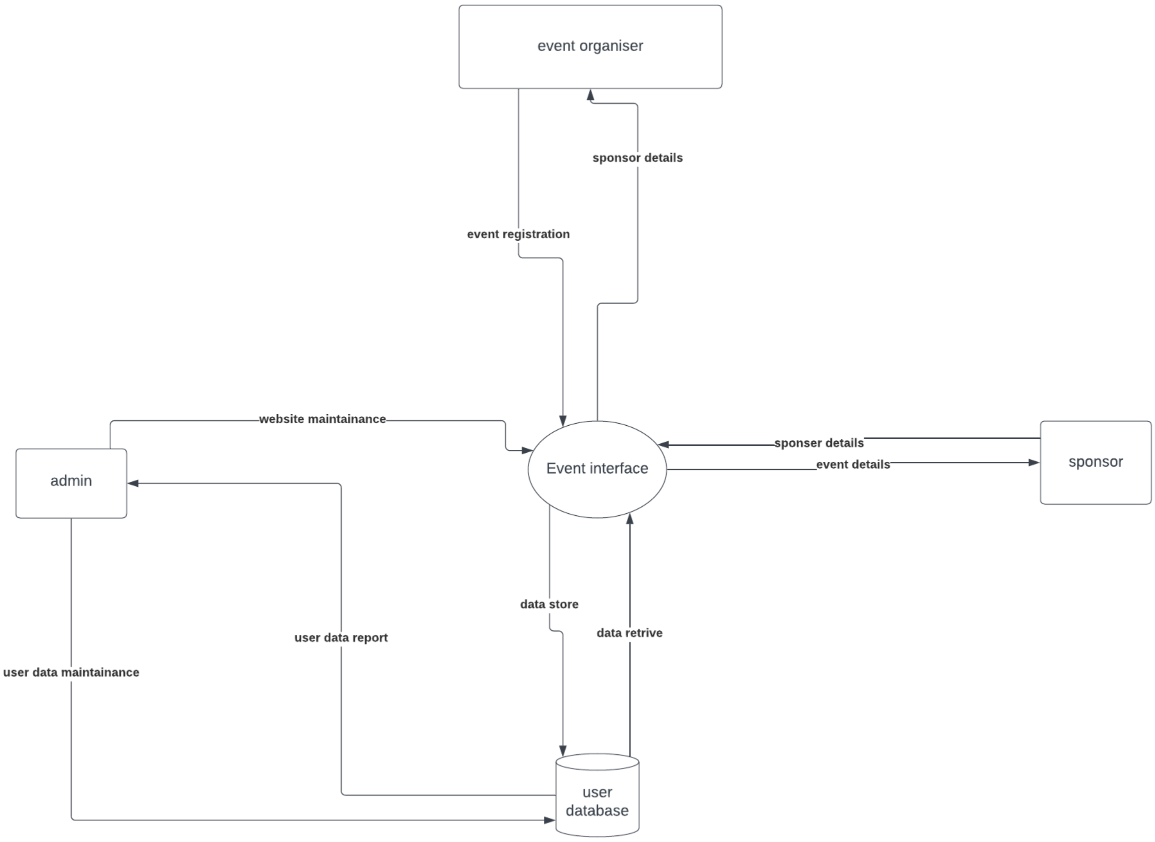


Figure 4.4: Level 0 Data Flow Diagram

A Level 0 Data Flow Diagram (DFD) provides a broad overview of the system, depicting the major processes or functions and their interrelationships, without diving into finer details. It shows the flow of data between external entities and the system's main processes, offering a high-level understanding of the system's functionality and data movement.

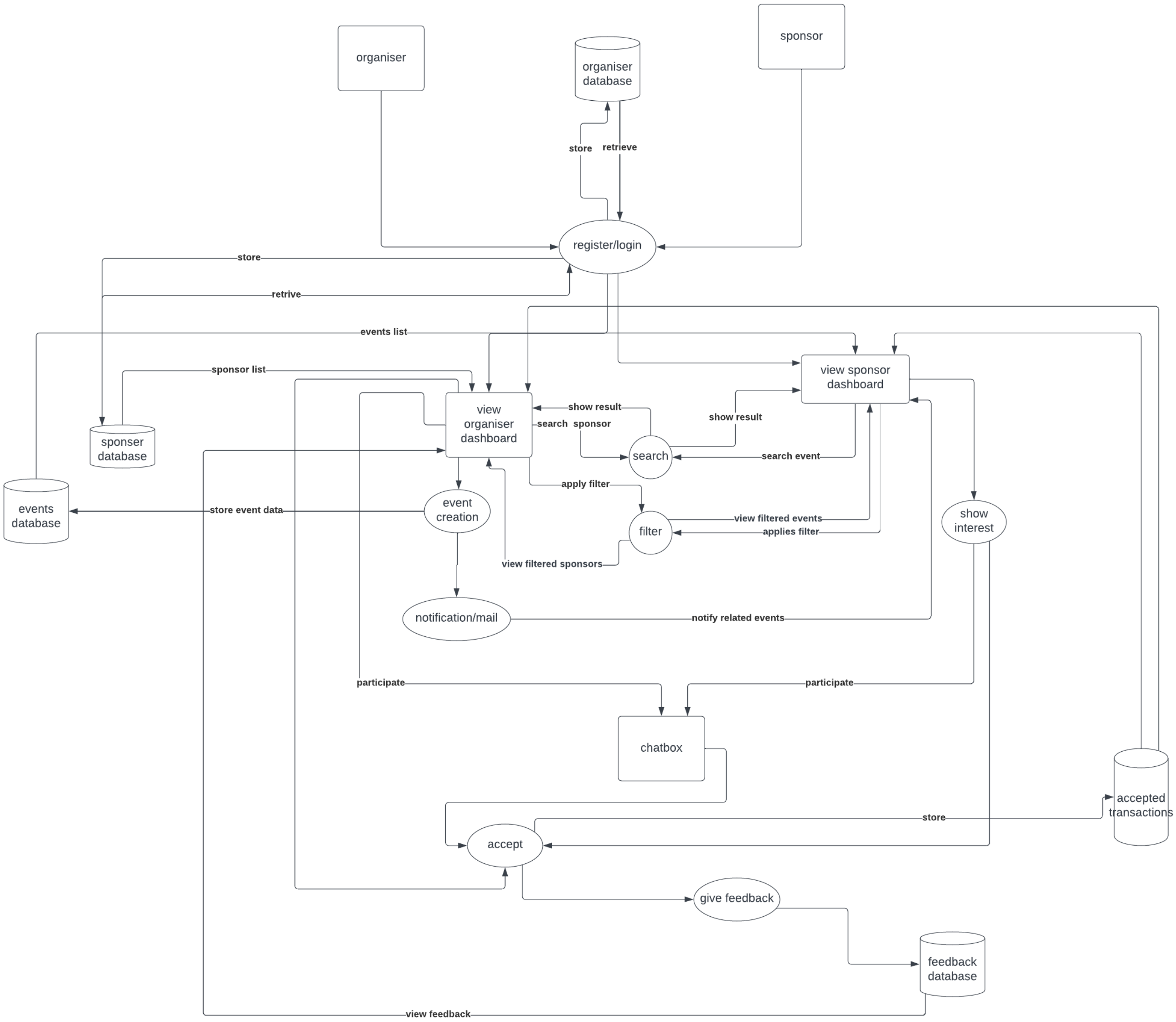


Figure 4.5: Level 1 Data Flow Diagram

A Level 1 Data Flow Diagram (DFD) provides more detailed insight compared to Level 0, breaking down the main processes identified in Level 0 into sub-processes or functions. It illustrates how data flows within each process, including inputs, outputs, and data stores, providing a more granular view of the system's functionality and data processing.

4.4 Entity-Relationship (ER) Diagram

The Entity-Relationship (ER) diagram serves as a visual representation of the database schema for the EventConnect platform. It illustrates the relationships between different entities and their attributes, providing insights into the data model and structure of the system.

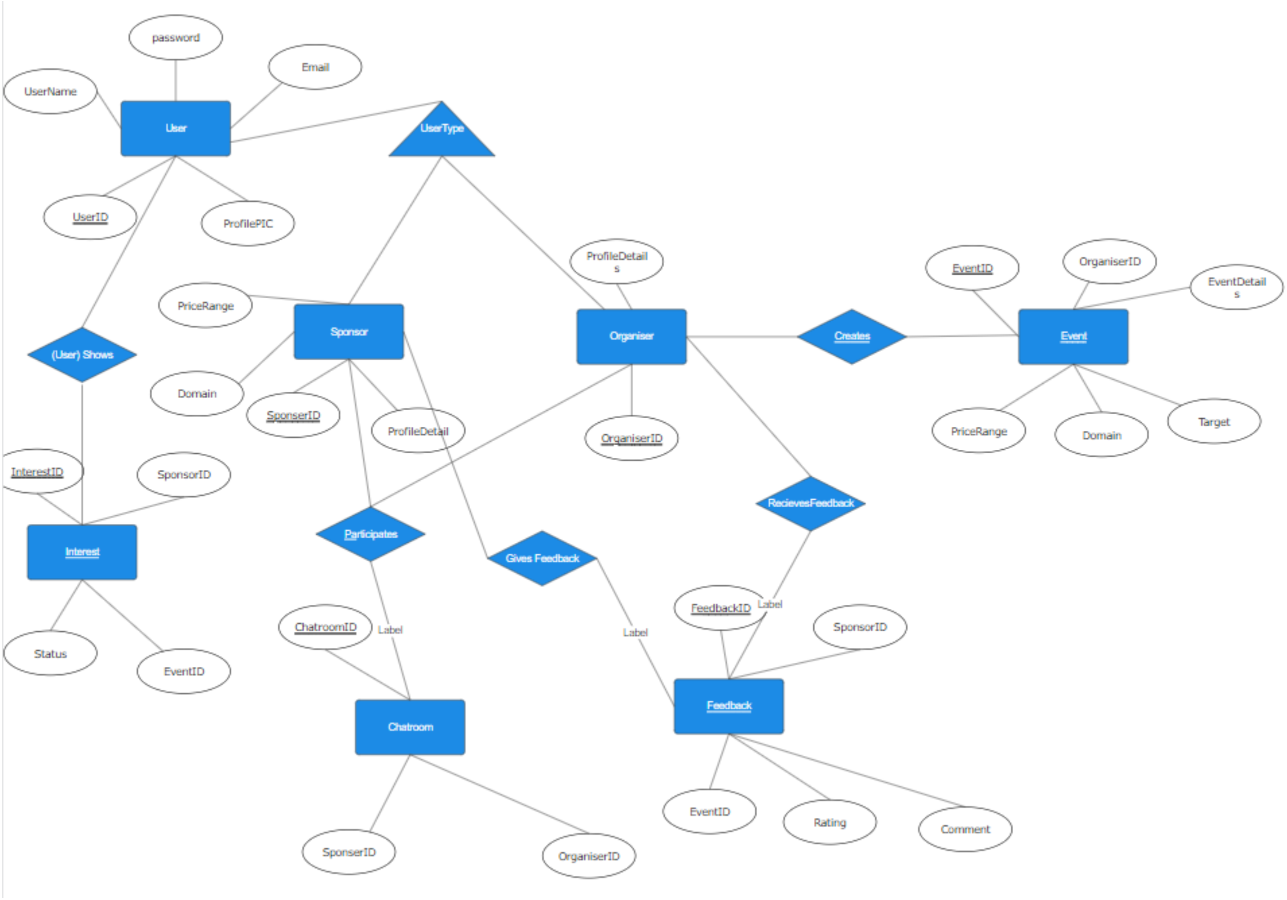


Figure 4.6: Entity-Relationship (ER) Diagram for EventConnect

Chapter 5

Detailed Design Document

In this chapter, we'll explore detailed design documentation essential for understanding our system's architecture, structure, and behavior, crucial for its implementation phase.

5.1 Use Case Diagram

A use case diagram illustrates the interactions between users and a system, showcasing various functionalities and how users interact with them. It visually represents actors, use cases, and their relationships, aiding in understanding system behavior and requirements.

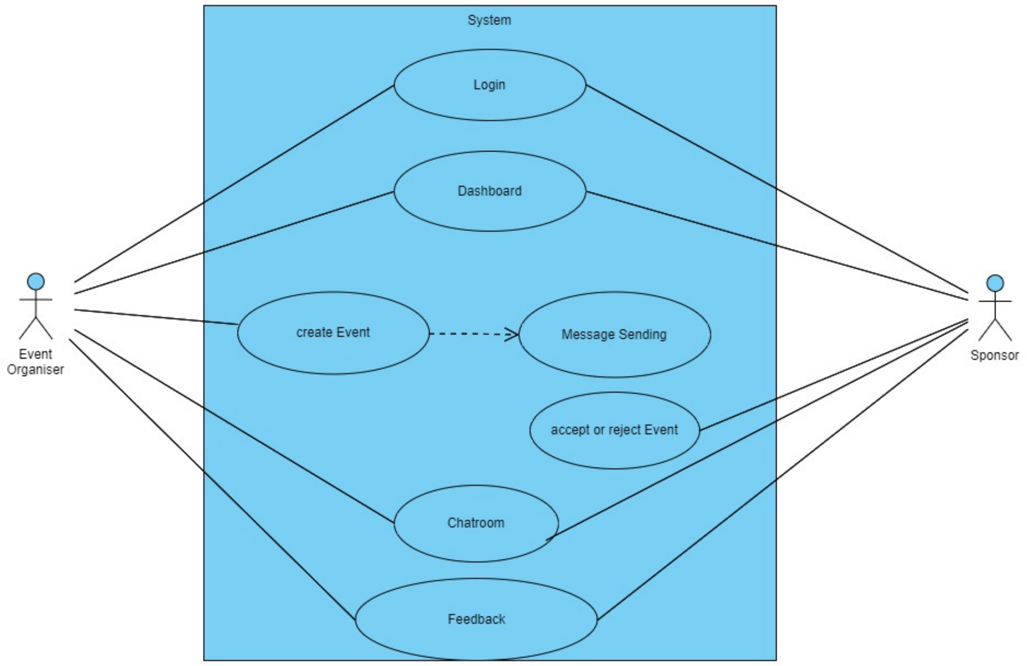


Figure 5.1: IPD Use Case Diagram

5.2 Class Diagram

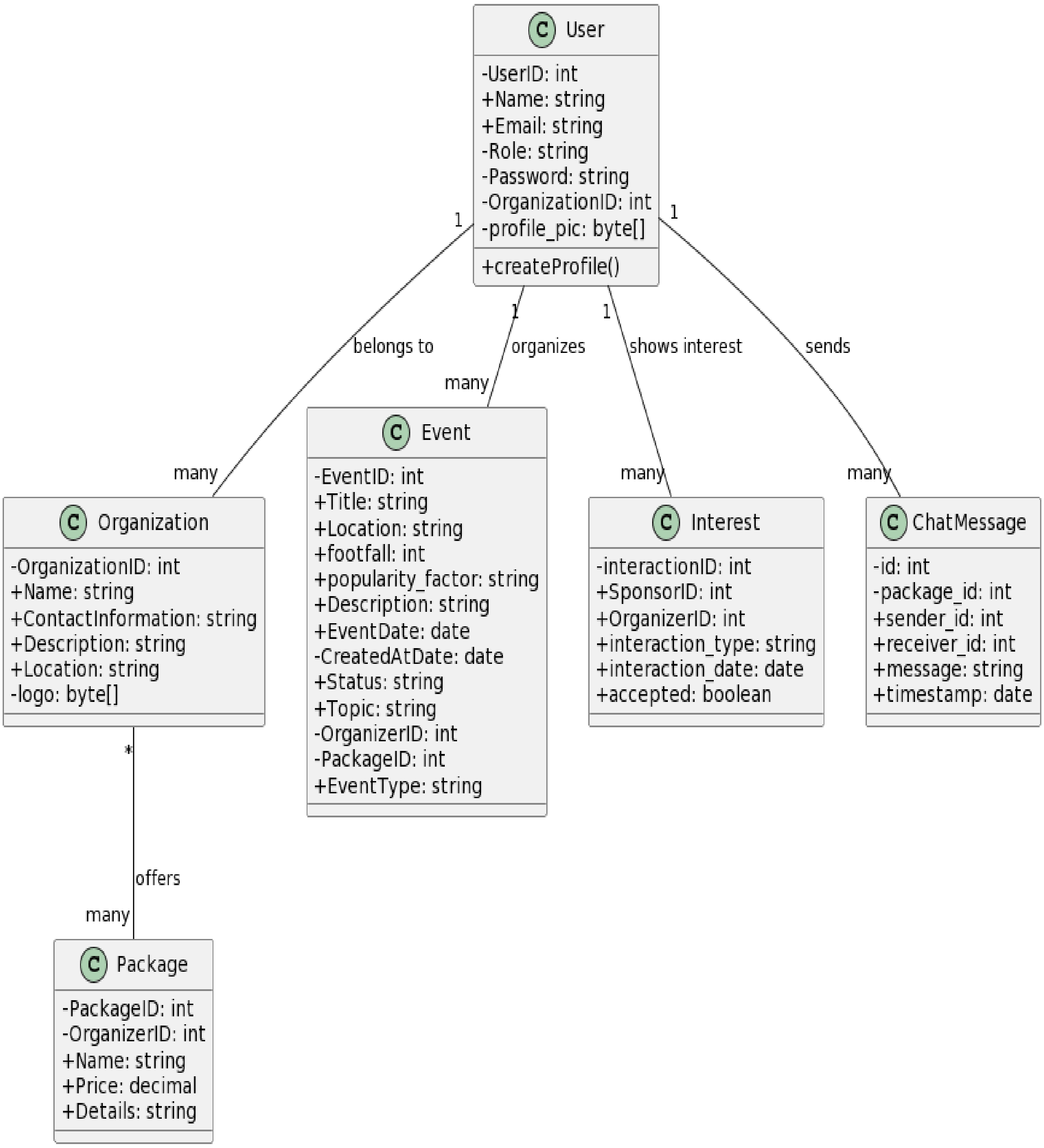


Figure 5.2: Class Diagram

5.3 Sequence Diagram

A sequence diagram is a type of interaction diagram that visualizes the interactions and messages exchanged between objects or components in a system over time. It illustrates the flow of control and data among various elements of the system, helping to understand the behavior of the system and the sequence of actions during runtime.

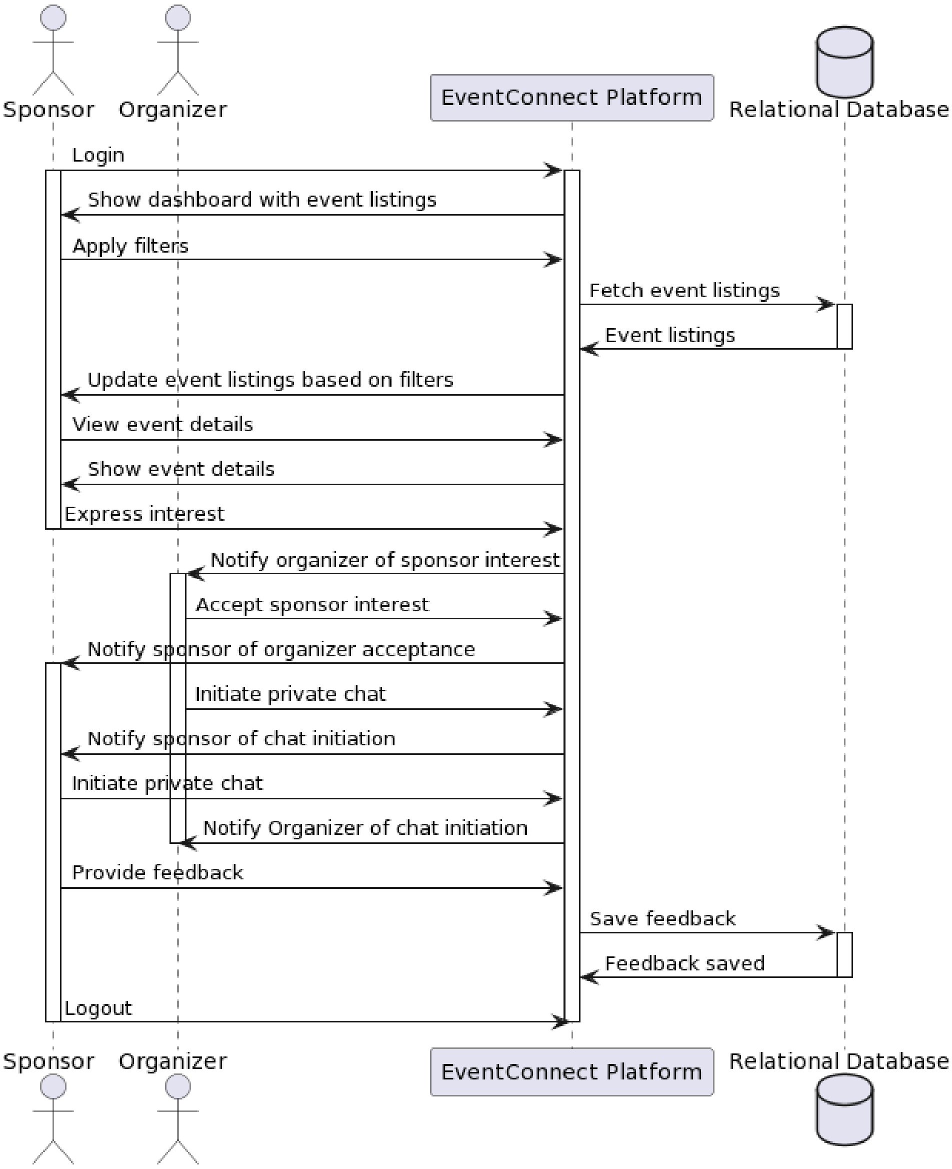


Figure 5.3: Sequence Diagram

5.4 Activity Diagram

An activity diagram is a type of behavioral diagram in the Unified Modeling Language (UML) that illustrates the flow of activities or actions within a system or process. It visually represents the sequential steps, decisions, and parallel activities involved in completing a task or achieving a goal.

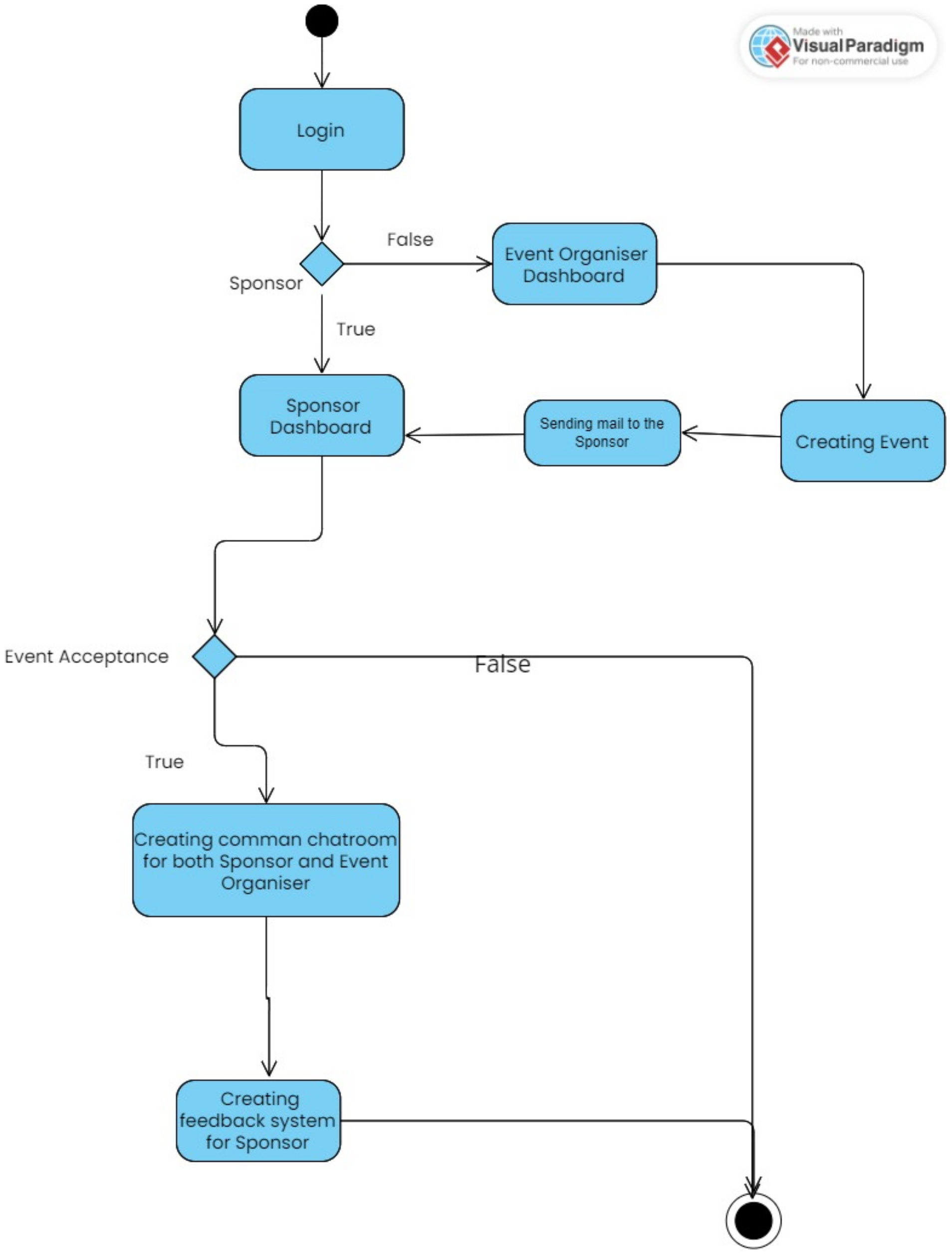

Login
Sponsor
False
Event Organiser Dashboard
True
Sponsor Dashboard
Sending mail to the Sponsor
Creating Event
Event Acceptance
False
True
Creating comman chatroom for both Sponsor and Event Organiser
Creating feedback system for Sponsor

Figure 5.4: Activity Diagram

5.5 Component Diagram

A component diagram is a type of structural diagram in UML that depicts the components of a system and their relationships. It illustrates the organization and dependencies between software components, providing a high-level view of the system's architecture. Components represent modular parts of the system that encapsulate functionality and can be replaced or extended independently.

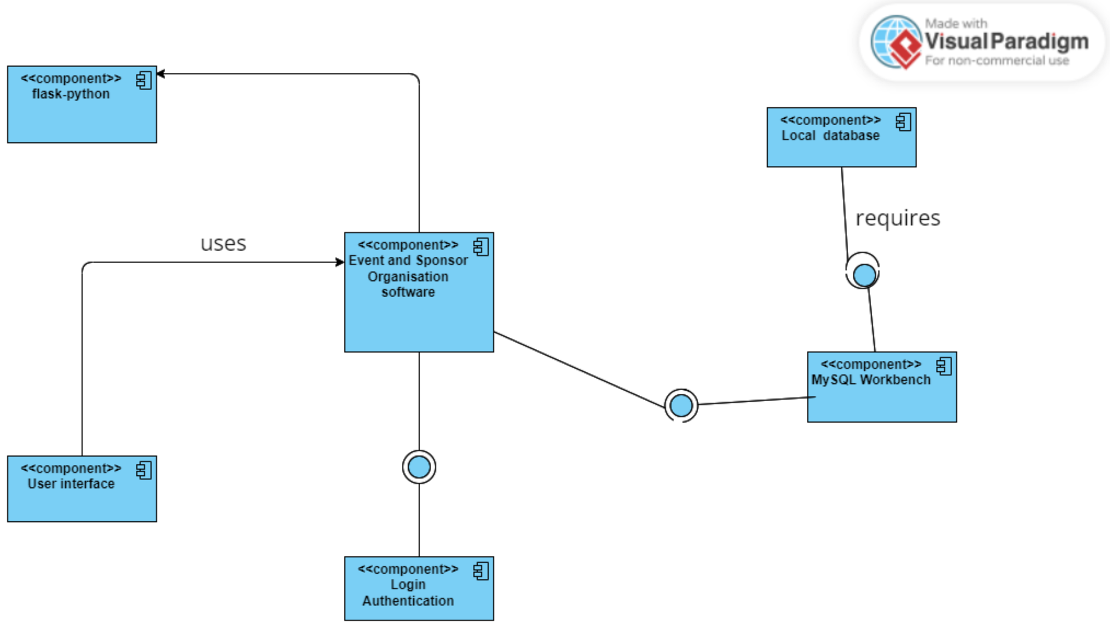

Made with Visual Paradigm For non-commercial use
<<component>> flask-python
<<component>> Local database
requires
uses
<<component>> Event and Sponsor Organisation software
<<component>> MySQL Workbench
<<component>> User interface
<<component>> Login Authentication

Figure 5.5: Component Diagram

5.6 Deployment Diagram

A deployment diagram illustrates the physical deployment of software components (such as servers, hardware devices, and software packages) within a network infrastructure. It shows how different components of a system are distributed across hardware nodes and interconnected through communication paths. Deployment diagrams are helpful in visualizing the deployment architecture and understanding how the system's physical components interact to support its functionality.

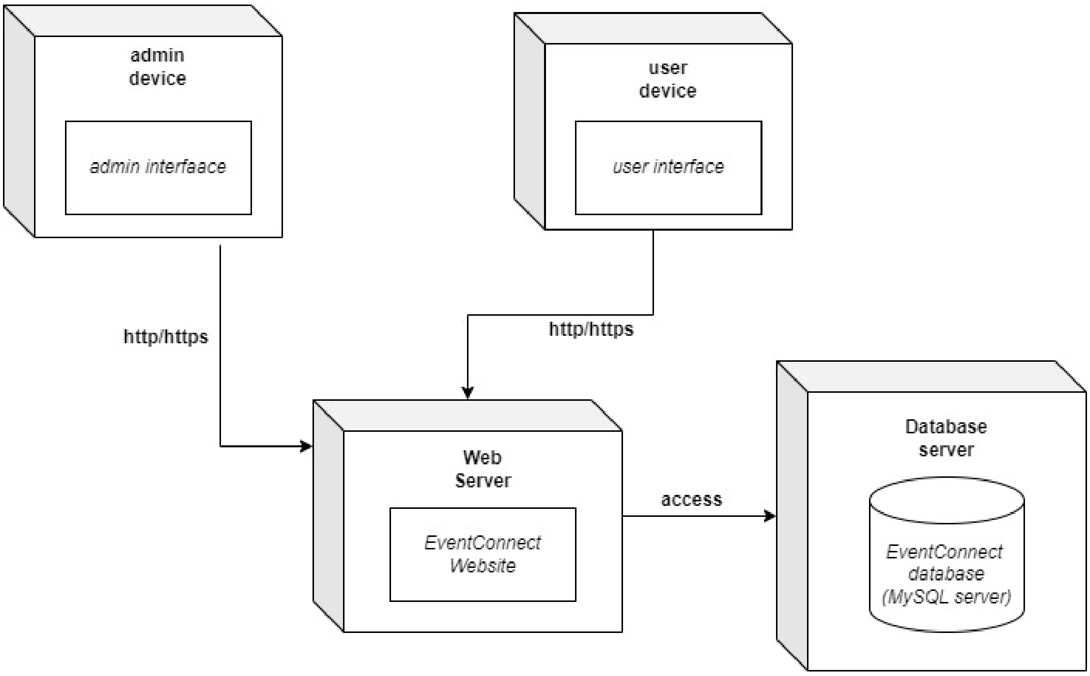
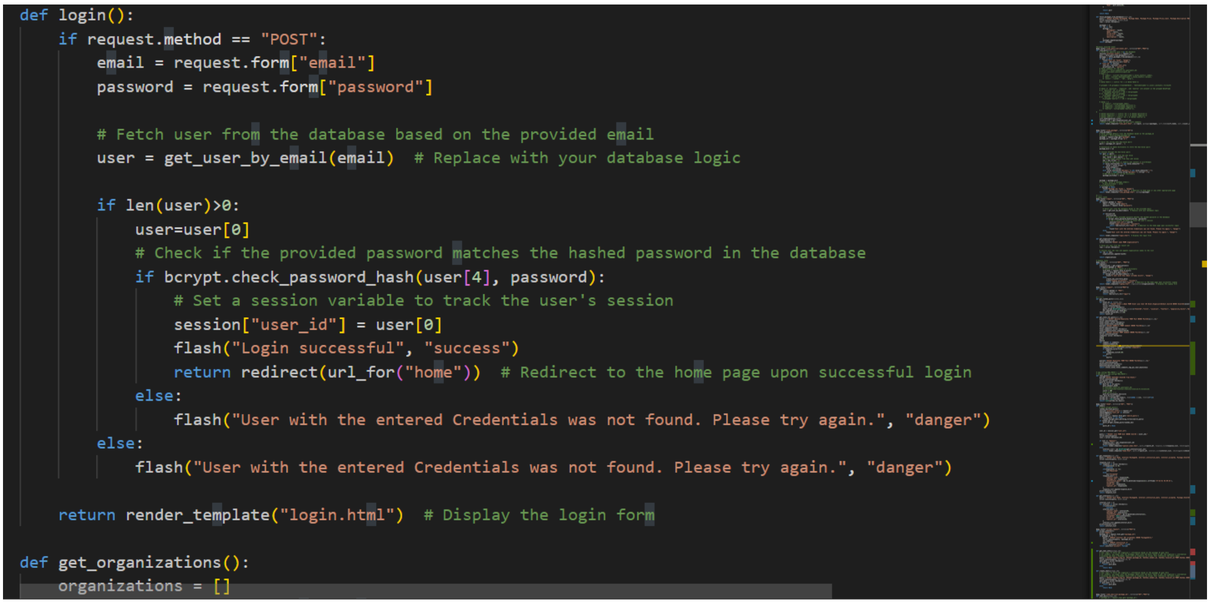


Figure 5.6: Deployment Diagram

Chapter 6

Coding Screenshots

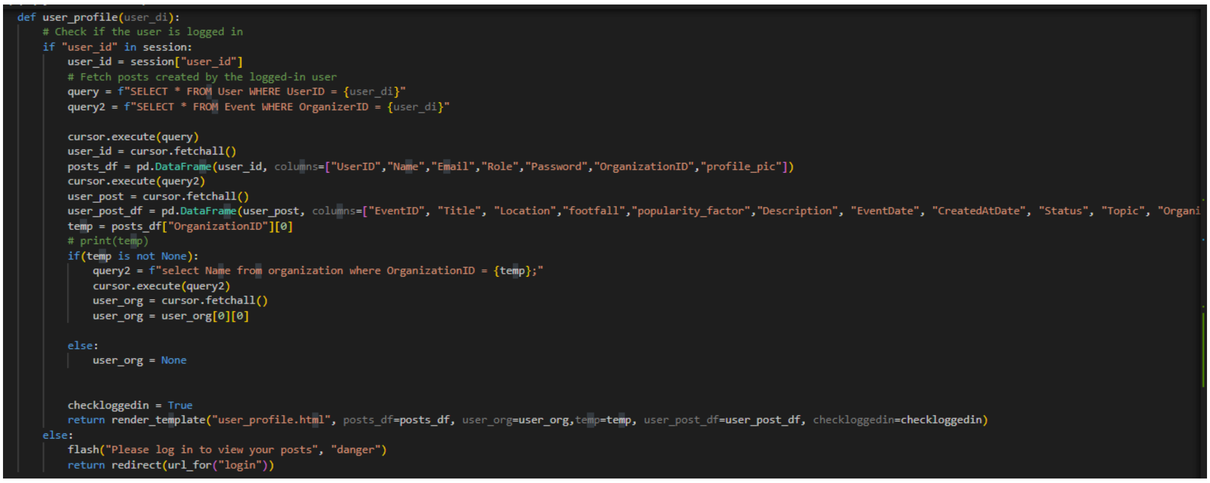
6.1 login



```
def login():
    if request.method == "POST":
        email = request.form["email"]
        password = request.form["password"]

        # Fetch user from the database based on the provided email
        user = get_user_by_email(email)  # Replace with your database logic

        if len(user)>0:
            user=user[0]
            # Check if the provided password matches the hashed password in the database
            if bcrypt.check_password_hash(user[4], password):
                # Set a session variable to track the user's session
                session["user_id"] = user[0]
                flash("Login successful", "success")
                return redirect(url_for("home"))  # Redirect to the home page upon successful login
            else:
                flash("User with the entered Credentials was not found. Please try again.", "danger")
        else:
            flash("User with the entered Credentials was not found. Please try again.", "danger")

    return render_template("login.html")  # Display the login form

def get_organizations():
    organizations = []
```

Figure 6.1: Code for login function

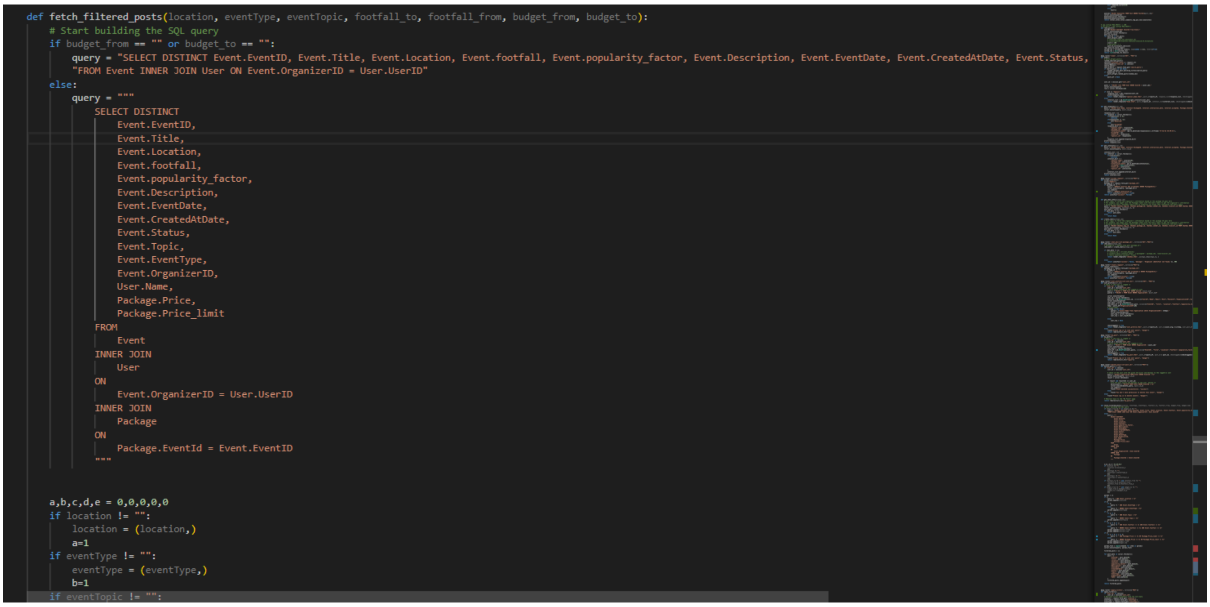
6.2 User profile



```python
def user_profile(user_di):
    # Check if the user is logged in
    if "user_id" in session:
        user_id = session["user_id"]
        # Fetch posts created by the logged-in user
        query = f"SELECT * FROM User WHERE UserID = {user_di}"
        query2 = f"SELECT * FROM Event WHERE OrganizerID = {user_di}"

        cursor.execute(query)
        user_id = cursor.fetchall()
        posts_df = pd.DataFrame(user_id, columns=["UserID","Name","Email","Role","Password","OrganizationID","profile_pic"])
        cursor.execute(query2)
        user_post = cursor.fetchall()
        user_post_df = pd.DataFrame(user_post, columns=["EventID", "Title", "Location","footfall","popularity_factor","Description", "EventDate", "CreatedAtDate", "Status", "Topic", "Organi
        temp = posts_df["OrganizationID"][0]
        # print(temp)
        if(temp is not None):
            query2 = f"select Name from organization where OrganizationID = {temp};"
            cursor.execute(query2)
            user_org = cursor.fetchall()
            user_org = user_org[0][0]

        else:
            user_org = None


        checkloggedin = True
        return render_template("user_profile.html", posts_df=posts_df, user_org=user_org,temp=temp, user_post_df=user_post_df, checkloggedin=checkloggedin)
    else:
        flash("Please log in to view your posts", "danger")
        return redirect(url_for("login"))
```

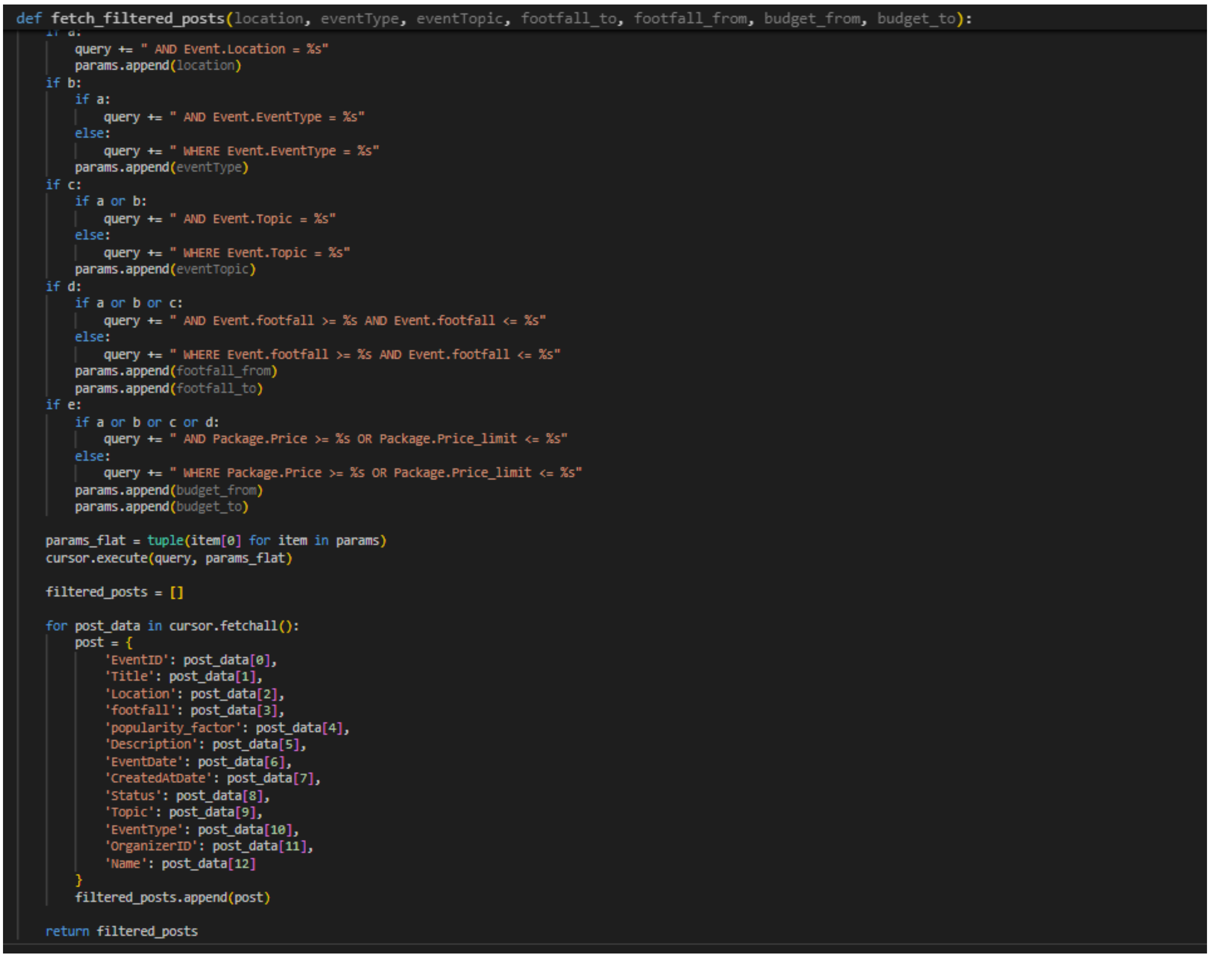
Figure 6.2: Code for User profile function of the Software

6.3 Event Filtering



```python
def fetch_filtered_posts(location, eventType, eventTopic, footfall_to, footfall_from, budget_from, budget_to):
    # Start building the SQL query
    if budget_from == "" or budget_to == "":
        query = "SELECT DISTINCT Event.EventID, Event.Title, Event.Location, Event.footfall, Event.popularity_factor, Event.Description, Event.EventDate, Event.CreatedAtDate, Event.Status,
        "FROM Event INNER JOIN User ON Event.OrganizerID = User.UserID"
    else:
        query = """
            SELECT DISTINCT
                Event.EventID,
                Event.Title,
                Event.Location,
                Event.footfall,
                Event.popularity_factor,
                Event.Description,
                Event.EventDate,
                Event.CreatedAtDate,
                Event.Status,
                Event.Topic,
                Event.EventType,
                Event.OrganizerID,
                User.Name,
                Package.Price,
                Package.Price_limit
            FROM
                Event
            INNER JOIN
                User
            ON
                Event.OrganizerID = User.UserID
            INNER JOIN
                Package
            ON
                Package.EventId = Event.EventID
            """


    a,b,c,d,e = 0,0,0,0,0
    if location != "":
        location = (location,)
        a=1
    if eventType != "":
        eventType = (eventType,)
        b=1
    if eventTopic != "":
```

Figure 6.3: Code for Event Filtering section of the Software



```python
def fetch_filtered_posts(location, eventType, eventTopic, footfall_to, footfall_from, budget_from, budget_to):
    if a:
        query += " AND Event.Location = %s"
        params.append(location)
    if b:
        if a:
            query += " AND Event.EventType = %s"
        else:
            query += " WHERE Event.EventType = %s"
        params.append(eventType)
    if c:
        if a or b:
            query += " AND Event.Topic = %s"
        else:
            query += " WHERE Event.Topic = %s"
        params.append(eventTopic)
    if d:
        if a or b or c:
            query += " AND Event.footfall >= %s AND Event.footfall <= %s"
        else:
            query += " WHERE Event.footfall >= %s AND Event.footfall <= %s"
        params.append(footfall_from)
        params.append(footfall_to)
    if e:
        if a or b or c or d:
            query += " AND Package.Price >= %s OR Package.Price_limit <= %s"
        else:
            query += " WHERE Package.Price >= %s OR Package.Price_limit <= %s"
        params.append(budget_from)
        params.append(budget_to)

    params_flat = tuple(item[0] for item in params)
    cursor.execute(query, params_flat)

    filtered_posts = []

    for post_data in cursor.fetchall():
        post = {
            'EventID': post_data[0],
            'Title': post_data[1],
            'Location': post_data[2],
            'footfall': post_data[3],
            'popularity_factor': post_data[4],
            'Description': post_data[5],
            'EventDate': post_data[6],
            'CreatedAtDate': post_data[7],
            'Status': post_data[8],
            'Topic': post_data[9],
            'EventType': post_data[10],
            'OrganizerID': post_data[11],
            'Name': post_data[12]
        }
        filtered_posts.append(post)

    return filtered_posts
```

Figure 6.4: Code for Event Filtering section of the Software

Chapter 7

Technical Dependencies

For the EventConnect project, there are several technical dependencies required to develop and deploy the platform successfully. These dependencies include programming languages, frameworks, libraries, and tools needed to implement various features and functionalities.

7.1 Programming Language

- **Python:** Python will serve as the primary programming language for developing the backend logic, data processing, and server-side functionalities of the EventConnect platform.

7.2 Web Framework

- **Flask:** Flask will be used as the web framework to facilitate rapid development, URL routing, request handling, and interaction with the backend database.

7.3 Database Management System

- **MySQL:** MySQL will be utilized as the relational database management system (RDBMS) to store and manage data related to event listings, sponsor profiles, organizer details, user accounts, messages, and other relevant information.

7.4 Frontend Technologies

- **HTML/CSS/JavaScript:** These frontend technologies will be used to design and develop the user interface (UI) of the EventConnect platform, including web pages, forms, navigation elements, and interactive components.
- **Bootstrap or Materialize CSS:** Bootstrap or Materialize CSS frameworks may be utilized to streamline the frontend development process, provide responsive design capabilities, and ensure consistency across different devices and screen sizes.

7.5 User Authentication

- **Authentication Libraries:** Authentication libraries (e.g., Flask-Login, Authentication) will be integrated into the backend to implement user authentication and authorization mechanisms, allowing users to register, log in, and access restricted features based on their roles (e.g., sponsors, organizers).

By addressing these technical dependencies and leveraging appropriate tools and technologies, the EventConnect platform can be developed

Chapter 8

Software Testing

This chapter presents a comprehensive overview of the test cases conducted to ensure the functionality and reliability of the EventConnect platform.

8.1 Test Cases

- **TC1: User Registration**
  - **Meaning:** This test case verifies that users can successfully register with valid credentials.
  - **Status:** Done
- **TC2: Dashboard Redirection**
  - **Meaning:** Upon successful login, users should be directed to their personalized dashboards.
  - **Status:** Done
- **TC3: Tailored Dashboards for Sponsors and Organizers**
  - **Meaning:** Ensure sponsors and event organizers have access to tailored dashboards with pertinent data.

  - **Status:** Done

- **TC4: Efficient Sponsor/Event Viewing and Searching**
  - **Meaning:** Allow users to view and search for sponsors/events efficiently.
  - **Status:** Done

- **TC5: Criteria-Based Filtering**
  - **Meaning:** Enable users to narrow down sponsor/event options based on specific criteria.
  - **Status:** Done

- **TC6: Comprehensive Event Management Tools**
  - **Meaning:** Provide event organizers with comprehensive tools to create and manage events seamlessly.
  - **Status:** Done

- **TC7: Effective Communication Through Chat Interface**
  - **Meaning:** Enable users to communicate effectively through a chat interface.
  - **Status:** Done

- **TC8: Transaction Record Preservation**
  - **Meaning:** Preserve a record of transactions for future reference and analysis.
  - **Status:** Done

- **TC9:‘ Data Access for Analysis and Ranking**
  - **Meaning:** Enable users to access data for analysis and ranking purposes.
  - **Status:** Done

- **TC10: Feedback Mechanism for Users**
  - **Meaning:** Allow users to provide feedback on sponsors and events.
  - **Status:** Done
- **TC11: Analysis Tools for Feedback and Data**
  - **Meaning:** Offer users tools to analyze feedback and transactional data.
  - **Status:** Done

Chapter 9

Results, Conclusions, and Future Work

In this section, we present the results obtained from the implementation of EventConnect, draw conclusions based on these results, and discuss potential future enhancements to the platform.

9.1 Results

The implementation of EventConnect has yielded the following results:

- Successful development and deployment of a web-based platform facilitating efficient connections between event sponsors and organizers.
- Implementation of user-friendly dashboards for sponsors and organizers, allowing easy navigation and interaction with the platform.
- Integration of advanced search and filtering mechanisms, enabling sponsors and organizers to find relevant partnership opportunities based on domain, price range, and other criteria.

- Seamless communication features,including private chat rooms, facilitating effective collaboration between sponsors and organizers.
- Implementation of a feedback and rating system, enabling users to provide and receive feedback on partnership experiences, fostering continuous improvement.

These results demonstrate the successful achievement of the platform's objectives and functionalities, enhancing the efficiency and effectiveness of sponsor-organizer connections in the event industry.

9.2 Conclusions

Based on the results obtained, we draw the following conclusions:

- EventConnect effectively addresses the existing gaps in the event industry by providing a centralized platform for sponsors and organizers to connect and collaborate.
- The user-friendly interface and intuitive design of the platform enhance user experience and usability, leading to increased engagement and satisfaction among users.
- The integration of advanced search and communication features streamlines the partnership development process, enabling sponsors and organizers to find suitable partners more efficiently.
- The feedback and rating system promotes transparency and accountability, fostering trust and confidence among users and improving the overall quality of partnerships.

Overall, EventConnect demonstrates significant potential to revolutionize sponsor-organizer connections in the event industry and drive mutually beneficial partnerships.

9.3 Future Work

While the current version of EventConnect meets the immediate requirements, there is scope for further enhancement and expansion in the following areas:

- Integration of machine learning algorithms to provide personalized recommendations for sponsors and organizers based on their preferences and past interactions.
- Enhancement of data analytics capabilities to derive actionable insights from user behavior and partnership outcomes, enabling continuous optimization of the platform.
- Expansion of the platform's reach and user base through strategic marketing initiatives and partnerships with industry stakeholders.

By addressing these areas of improvement, future iterations of EventConnect can further solidify its position as the premier platform for sponsor-organizer connections in the event industry.

Chapter 10

Acknowledgments

We would like to extend our heartfelt appreciation to Akash Nale, CEO at OMart and Flyyo.india, for their invaluable contributions and support throughout the development of EventConnect.

Akash Nale's expertise and guidance were instrumental in shaping the vision and direction of this project. Their insights and recommendations played a crucial role in the design and implementation of key features, ensuring the platform's effectiveness and usability.

We are deeply grateful for Akash Nale's dedication and commitment to this project, as well as their willingness to share their knowledge and expertise with our team. Their mentorship has been invaluable in overcoming challenges and achieving our goals.

Finally, we extend our thanks to all our users and stakeholders who provided feedback and support throughout the development process. Your input has been instrumental in shaping EventConnect into a valuable resource for the event industry.