



**FINAL YEAR PROJECT REPORT
BS (COMPUTER SCIENCE)**

**EVENT FLOW (ONLINE TICKETING EVENT
MANAGEMENT SYSTEM)**

SUBMITTED BY

| | |
|-------------------------------------|--------------|
| SHEIKH MOHAMMAD OSAMA FAYYAZ | 50605 |
| HUZAIFA AHMAD | 50922 |
| MUHAMMAD SAMEER | 50551 |

SUPERVISOR

SIR MAAZ AHMED

COORDINATOR

DR AARIJ MAHMOOD HUSSAAN

FACULTY OF ENGINEERING, SCIENCE AND TECHNOLOGY

IQRA UNIVERSITY, KARACHI

MARCH 2024

ABSTRACT

This project focuses on developing a user-friendly website for online event management and ticket sales; we use agile methodology for flexibility and responsiveness during development. The initiative addresses challenges faced by event organizers handling tasks manually, such as ticket management and attendee registration. Existing systems fall short in simplifying event management and enhancing attendee satisfaction.

The proposed Online Ticketing Event Management System aims to streamline event planning, automate tasks, and boost efficiency for both organizers and attendees. Leveraging agile methodology, the project breaks down tasks into user stories, prioritizes based on feedback, and delivers working increments iteratively.

Key features include customizable ticket options, invitation campaigns, access control, and diverse payment methods. The system also offers a calendar view of events and analytics for tracking ticket sales and engagement. With a user-friendly interface, the system aims to transform event management into an intuitive and enjoyable experience for all involved parties.

We have approved this manuscript for submission and presentation as fulfillment of Bachelor of Computer Science/ Software Engineering.

Supervisor: SIR MAAZ AHMED
Date: 29-03-2024

Project Coordinator: DR AARIJ MAHMOOD HUSSAAN
Date: 29-03-2024

DECLARATION

I hereby declare that the work has been done by myself to fulfill the requirement of the BS (Computer Science) and no portion of the work contained in this report has been submitted in support of any web application for any other degree or qualification of this or any other university or institute of learning.

I hereby further declare that in the event of any infringement of the provision of the Act whether knowingly or unknowingly the university shall not be liable for the same in any manner whatsoever and undertake to indemnify and keep the university indemnified against all such claims and actions.

© SHEIKH MOHAMMAD OSAMA FAYYAZ [50605]

© HUZAIFA AHMAD [50922]

©MUHAMMAD SAMEER [50551]

ACKNOWLEDGEMENT

First, we thank Almighty Allah who praise us with the ability to think, work and deliver what we are assigned to do. Secondly, we must be grateful to our supervisor "**Sir Maaz Ahmed**" who helps us in this project. We also acknowledge our teachers that throughout our studies helps us and guides us, departmental staff, university staff or other then this. We are also thankful to the FYP instructor "**Dr. Aarij Mahmood Hussaan**" for his precious support throughout the tenure as he is the best instructor for FYP who makes every student to be updated with the project progress and lead to the completion with great success within the time period given. We are also grateful to our family and friends, for supporting and encouraging us to complete this project. Finally, we would like to thank all the colleagues of IQRA University who have been with us in all difficult times with suggestions and supportive words, which carry us to make this project a reality.

LIST OF ACRONYMS

EF =Event Flow

OTEM = Online Ticketing event management system

TABLE OF CONTENTS

Contents

| | |
|--|-----------|
| CHAPTER – 1: | 1 |
| 1.0. INTRODUCTION: | 1 |
| 1.1. PROBLEM STATEMENT: | 2 |
| 1.3. OBJECTIVE: | 3 |
| <i>1.3.1. Streamlining ticket customization:</i> | 3 |
| <i>1.3.2. Enhancing promotional efforts:</i> | 3 |
| <i>1.3.3. Improving attendee registration experience:</i> | 3 |
| <i>1.3.4. Ensuring access control:</i> | 3 |
| <i>1.3.5. Simplifying check-in process:</i> | 3 |
| <i>1.3.6. Facilitating seat assignment:</i> | 3 |
| <i>1.3.7. Providing flexible payment options:</i> | 3 |
| <i>1.3.8. Ensuring purchase protection:</i> | 3 |
| 1.4. CHALLENGES: | 4 |
| <i>1.4.1. Limited technology literacy:</i> | 4 |
| <i>1.4.2. Technical difficulties during development:</i> | 4 |
| <i>1.4.3. Managing event logistics:</i> | 4 |
| <i>1.4.4. Security concerns:</i> | 4 |
| <i>1.4.5. Meeting user requirements:</i> | 4 |
| 1.5. STRUCTURE OF REPORT: | 5 |
| <i>1.5.1. Chapter 2: Technology Background:</i> | 5 |
| <i>1.5.2. Chapter 3: Requirements & Methodology:</i> | 5 |
| <i>1.5.3. Chapter 4: Project Plan & Initial Design:</i> | 5 |
| <i>1.5.4. Chapter 5: Project Design & Development:</i> | 5 |
| <i>1.5.5. Chapter 6: Testing:</i> | 5 |
| <i>1.5.6. Chapter 7: Conclusion:</i> | 5 |
| CHAPTER – 2: | 6 |
| 2.0. TECHNOLOGY OVERVIEW: | 6 |
| <i>2.0.1. Technology Stack:</i> | 6 |
| <i>2.0.2. Web Application Overview:</i> | 7 |
| 2.1. LITERATURE REVIEW: ONLINE TICKETING EVENT MANAGEMENT SYSTEMS | 8 |
| <i>2.1.1. Literature Review:</i> | 8 |
| <i>2.1.2. Introduction:</i> | 8 |
| <i>2.1.3. Existing Platforms:</i> | 8 |
| <i>2.1.4. Comparison and Conclusion:</i> | 9 |
| CHAPTER – 3 | 10 |
| 3.0. INTRODUCTION: | 10 |
| 3.1. PROJECT PLAN: | 11 |
| 3.2. GANT CHART: | 12 |
| 3.3. FUNCTIONAL REQUIREMENTS: | 13 |
| <i>3.3.1. TICKET CUSTOMIZATION:</i> | 13 |
| <i>3.3.2. INVITATION CAMPAINGS:</i> | 13 |
| <i>3.3.3. REGISTRATION QUESTION FORM & CONFIRMATION EMAIL:</i> | 13 |
| <i>3.3.4. ACCESS CONTROL:</i> | 13 |
| <i>3.3.5. NAME BADGES & QR CODE:</i> | 13 |

| | |
|---|------------|
| 3.3.6. SEAT ASSIGNMENT: | 13 |
| 3.3.7. CALENDAR VIEW: | 13 |
| 3.3.8. PAYMENT OPTIONS:..... | 13 |
| 3.3.9. PURCHASE PROTECTION..... | 13 |
| 3.3.10: TICKET RESERVATIONS:..... | 14 |
| 3.3.12. SURVEY QUESTIONS: | 14 |
| 3.4. NON-FUNCTIONAL REQUIREMENTS: | 14 |
| 3.4.1. PERFORMANCE | 14 |
| 3.4.2. SCALABILITY: | 14 |
| 3.4.3. SECURITY:..... | 14 |
| 3.4.4. RELIABILITY:..... | 14 |
| 3.4.5. USABILITY: | 14 |
| 3.4.6. INTEGRATION:..... | 14 |
| 3.5. HARDWARE REQUIREMENTS: | 14 |
| 3.6. SUMMARY: | 15 |
| CHAPTER – 4 | 16 |
| 4.0. INTRODUCTION:..... | 16 |
| 4.1. STATE DIAGRAM: | 17 |
| 4.2. ENTITY RELATIONSHIP DIAGRAM | 19 |
| 4.3. USE CASE DIAGRAM..... | 20 |
| 4.4. USE CASES:..... | 21 |
| 4.5. SUMMARY: | 32 |
| CHAPTER – 5 | 33 |
| 5.0. INTRODUCTION:..... | 33 |
| 5.1. PROTOTYPE DESIGN:..... | 34 |
| 5.2 .DATABASE QUERIES: | 40 |
| 5.3. EXTERNAL LIBRARIES:..... | 51 |
| 5.3.1. Bootstrap: | 51 |
| 5.3.2. PHP Mailer:..... | 51 |
| 5.3.3. jQuery:..... | 51 |
| 5.3.4. PayPal API:..... | 51 |
| 5.3.5. Full Calendar:..... | 51 |
| 5.4. SCREENSHOTS:..... | 52 |
| <i>Attendee look of website:</i> | 52 |
| <i>Organizer look of website:</i> | 61 |
| <i>Admin look of website:</i> | 75 |
| 5.5. SUMMARY: | 79 |
| CHAPTER – 6 | 80 |
| 6.0. INTRODUCTION:..... | 80 |
| 6.1. PROJECT TEST CASES: | 81 |
| <i>Attendee Test Cases:</i> | 81 |
| <i>Organizer Test Cases:</i> | 90 |
| 6.2. SUMMARY: | 99 |
| CHAPTER – 7 | 100 |
| 7.0. INTRODUCTION:..... | 100 |
| 7.1. SYSTEM LIMITATIONS AND CHALLENGES: | 100 |

| | |
|--|------------|
| <i>7.1.1. Internet Connection required*:</i> | 100 |
| <i>7.1.2. Tech Know-How*:</i> | 100 |
| <i>7.1.3. Parental Controls*:</i> | 100 |
| <i>7.1.4. Camera Challenges*:</i> | 100 |
| 7.2. FUTURE WORK: | 101 |
| <i>7.2.1. More Features</i> | 101 |
| <i>7.2.2. Better User Experience:</i> | 101 |
| <i>7.2.3. Improved Security:</i> | 101 |
| <i>7.2.4. Recurring Billing:</i> | 101 |
| <i>7.2.5. QR Code Scanner Machine:</i> | 101 |
| <i>7.2.6. Seat Mapping Tools:</i> | 101 |
| 7.3. CONCLUSION: | 101 |
| REFERENCES | 102 |
| APPENDIX | 103 |
| BUSINESS CANVAS: | 103 |
| DETAILED GANTT CHART: | 104 |
| SOFTWARE MANUAL: | 105 |
| INTRODUCTION | 105 |
| <i>Attendee User Manual:</i> | 105 |
| 1. Creating an Account: | 105 |
| 2. Signing In: | 105 |
| 3. Viewing Events: | 105 |
| 4. Viewing Calendars: | 105 |
| 5. Buying Tickets: | 105 |
| 6. Checkout: | 105 |
| 7. Ticket Confirmation: | 105 |
| 8. Managing Tickets: | 106 |
| 9. Refund Policies: | 106 |
| 10. Wallet Management: | 106 |
| 11. Event Categories: | 106 |
| <i>Organizer User Manual:</i> | 106 |
| 1. Creating an Account: | 106 |
| 2. Signing In: | 106 |
| 3. Creating Main Event: | 106 |
| 4. Creating Sub-Events: | 106 |
| 5. Generating Ticket Types: | 106 |
| 6. Generating Tickets: | 106 |
| 7. Creating Blogs: | 107 |
| 8. Adding Add-ons: | 107 |
| 9. Ticket Templates: | 107 |
| 10. Wallet Management: | 107 |
| 11. Viewing Orders: | 107 |
| 12. Event Management: | 107 |

LIST OF TABLES

| | |
|--|----|
| TABLE 1: PROJECT PLAN | 11 |
| ACTOR = ATTENDEE USE CASES | 21 |
| TABLE 2: USE CASE1=CREATE ACCOUNT | 21 |
| TABLE 3: USE CASE2=LOGIN ACCOUNT | 21 |
| TABLE 4: USE CASE3=VIEW EVENT | 21 |
| TABLE 5: USE CASE4=CALENDAR VIEW | 22 |
| TABLE 6: USE CASE5=TICKET TYPE | 22 |
| TABLE 7: USE CASE6=PAYMENT | 22 |
| TABLE 8: USE CASE7=E-TICKET | 23 |
| TABLE 9: USE CASE8=REFUND | 23 |
| TABLE 10: USE CASE9=PAYMENT DETAIL | 23 |
| TABLE 11: USE CASE10=CHECK-IN..... | 24 |
| TABLE 12: USE CASE11=ATTENDANCE..... | 24 |
| ACTOR = ORGANIZER USE CASES..... | 24 |
| TABLE 13: USE CASE12=REGISTER ACCOUNT | 24 |
| TABLE 14: USE CASE13=LOGIN ACCOUNT | 25 |
| TABLE 15: USE CASE14=TICKET CUSTOMIZATION | 25 |
| TABLE 16: USE CASE15=RESERVED TICKET..... | 25 |
| TABLE 17: USE CASE16=CREATE EVENT | 26 |
| TABLE 18: USE CASE17=GENERATE REPORT | 26 |
| TABLE 19: USE CASE18=VIEW EVENT DETAIL | 26 |
| TABLE 20: USE CASE19=EDIT EVENT DETAIL | 27 |
| TABLE 21: USE CASE20=GET PAYMENT | 27 |
| TABLE 22: USE CASE21=INVITATION CAMPAIGN..... | 27 |
| TABLE 23: USE CASE22=SURVEY QUESTIONS | 28 |
| TABLE 24: USE CASE23= PROCESS PAYMENT..... | 28 |
| TABLE 25: USE CASE24= PAYMENT REFUND | 28 |
| TABLE 26: USE CASE25= GENERATE QR CODE | 29 |
| TABLE 27: USE CASE26= PASTE THE QR CODE IN TICKET..... | 29 |
| TABLE 28: USE CASE27= CHECK-IN WITH QR CODE..... | 29 |
| TABLE 29: USE CASE28= SEND TICKETS | 30 |
| TABLE 30: USE CASE29= SEND SURVEY QUESTIONS | 30 |

| | |
|--|-----------|
| TABLE 31: USE CASE30= SEND INVITATION CAMPAIGN..... | 30 |
| ACTOR=ADMIN USE CASES..... | 31 |
| TABLE 32: USE CASE31= LOGIN..... | 31 |
| TABLE 33: USE CASE32= MANAGE SYSTEM | 31 |
| TABLE 34: USE CASE33= MANAGE EVENTS..... | 31 |
| TABLE 35: USE CASE34= ASSIGN SEATS | 32 |
| TABLE 36: PROTOTYPE1= REGISTER PAGE..... | 34 |
| TABLE 37: PROTOTYPE2= LOGIN PAGE..... | 35 |
| TABLE 38: PROTOTYPE3= MAIN WINDOW | 36 |
| TABLE 39: PROTOTYPE4= MANAGE EVENT WINDOW | 37 |
| TABLE 40: PROTOTYPE5= CUSTOMER WINDOW..... | 38 |
| TABLE 41: PROTOTYPE6= ANALYTICS REPORT WINDOW | 39 |
| TABLE 42: ATTENDEE TEST CASE 1 | 81 |
| TABLE 43: ATTENDEE TEST CASE 2 | 81 |
| TABLE 44: ATTENDEE TEST CASE 3 | 82 |
| TABLE 45: ATTENDEE TEST CASE 4 | 82 |
| TABLE 46: ATTENDEE TEST CASE 5 | 83 |
| TABLE 47: ATTENDEE TEST CASE 6 | 83 |
| TABLE 48: ATTENDEE TEST CASE 7 | 84 |
| TABLE 49: ATTENDEE TEST CASE 8 | 84 |
| TABLE 50: ATTENDEE TEST CASE 9 | 85 |
| TABLE 51: ATTENDEE TEST CASE 10 | 85 |
| TABLE 52: ATTENDEE TEST CASE 11 | 86 |
| TABLE 53: ATTENDEE TEST CASE 12 | 86 |
| TABLE 54: ATTENDEE TEST CASE 13 | 87 |
| TABLE 55: ATTENDEE TEST CASE 14 | 87 |
| TABLE 56: ATTENDEE TEST CASE 15 | 88 |
| TABLE 57: ATTENDEE TEST CASE 16 | 88 |
| TABLE 58: ATTENDEE TEST CASE 17 | 89 |
| TABLE 59: ORGANIZER TEST CASE 1..... | 90 |
| TABLE 60: ORGANIZER TEST CASE 2..... | 90 |
| TABLE 61: ORGANIZER TEST CASE 3..... | 91 |
| TABLE 62: ORGANIZER TEST CASE 4..... | 91 |
| TABLE 63: ORGANIZER TEST CASE 5..... | 92 |
| TABLE 64: ORGANIZER TEST CASE 6..... | 92 |

| | |
|---------------------------------------|----|
| TABLE 65: ORGANIZER TEST CASE 7..... | 93 |
| TABLE 66: ORGANIZER TEST CASE 8..... | 93 |
| TABLE 67: ORGANIZER TEST CASE 9..... | 94 |
| TABLE 68: ORGANIZER TEST CASE 10..... | 94 |
| TABLE 69: ORGANIZER TEST CASE 11..... | 95 |
| TABLE 70: ORGANIZER TEST CASE 12..... | 95 |
| TABLE 71: ORGANIZER TEST CASE 13..... | 96 |
| TABLE 72: ORGANIZER TEST CASE 14..... | 96 |
| TABLE 73: ORGANIZER TEST CASE 15..... | 97 |
| TABLE 74: ORGANIZER TEST CASE 16..... | 97 |
| TABLE 75: ORGANIZER TEST CASE 17..... | 98 |
| TABLE 76: ORGANIZER TEST CASE 18..... | 98 |
| TABLE 77: ORGANIZER TEST CASE 19..... | 99 |
| TABLE 78: ORGANIZER TEST CASE 20..... | 99 |

LIST OF FIGURES

| | |
|--|----|
| FIGURE.1: GANT CHART | 12 |
| FIGURE 2: ATTENDEE STATE DIAGRAM | 17 |
| FIGURE 3: ORGANIZER STATE DIAGRAM | 18 |
| FIGURE.4: ENHANCED ENTITY RELATIONSHIP DIAGRAM | 19 |
| FIGURE.5: USE CASE DIAGRAM | 20 |
| FIGURE6: SIGNUP AND LOGIN PAGE:..... | 40 |
| FIGURE7: CREATE, EDIT AND DELETE ORGANIZER: | 41 |
| FIGURE8: USER PROFILE PAGE:..... | 42 |
| FIGURE9: CREATING NEW EVENTS:..... | 43 |
| FIGURE10: EDITING AND DELETING MAIN EVENTS: | 44 |
| FIGURE11: EDITING AND DELETING SUB EVENTS: | 45 |
| FIGURE12: CREATING AND DELETING BLOGS:..... | 46 |
| FIGURE13: CREATING AND DELETING EVENT CATEGORIES:..... | 46 |
| FIGURE14: CREATING EDITING AND DELETING EVENT TICKETS:..... | 47 |
| FIGURE15: CREATING EDITING AND DELETING EVENT TICKET TYPES:..... | 48 |
| FIGURE16: CREATING EDITING AND DELETING TICKET ADD-ON:..... | 49 |
| FIGURE17: CREATING QUESTION FORM: | 50 |
| FIGURE18: CREATING AND DELETING REGISTRATION QUESTIONS: | 50 |
| FIGURE19: SCREEN SHOT 1: MAIN PAGE (HEADER AND BANNER) | 52 |
| FIGURE20: SCREEN SHOT 2: MAIN PAGE (FEATURED EVENTS) | 52 |
| FIGURE21: SCREEN SHOT 3: MAIN PAGE(FOOTER) | 53 |
| FIGURE22: SCREEN SHOT 4: ABOUT PAGE..... | 53 |
| FIGURE23: SCREEN SHOT 5: EVENTS PAGE | 54 |
| FIGURE34: SCREEN SHOT 6: POLICY PAGE | 54 |
| FIGURE35: SCREEN SHOT 7: EVENT CATEGORIES PAGE | 55 |
| FIGURE36: SCREEN SHOT 8:SUB EVENTS PAGE..... | 55 |
| FIGURE37: SCREEN SHOT 9: EVENT TICKET PAGE | 56 |
| FIGURE38: SCREEN SHOT 10: CART PAGE..... | 56 |
| FIGURE39: SCREEN SHOT 11: AFTER CHECKOUT PAGE | 57 |
| FIGURE40: SCREEN SHOT 12: WALLET PAGE..... | 57 |
| FIGURE41: SCREEN SHOT 13: ATTENDEE BOUGHT TICKETS PAGE | 58 |
| FIGURE42: SCREEN SHOT 14: ATTENDEE BOUGHT TICKET STATUS PAGE | 58 |
| FIGURE43: SCREEN SHOT 15: SHOWN TICKET PAGE | 59 |

| | |
|---|-----------|
| FIGURE44: SCREEN SHOT 16: ATTENDEE PROFILE PAGE | 59 |
| FIGURE45: SCREEN SHOT 17: CALENDAR VIEW PAGE | 60 |
| FIGURE46: SCREEN SHOT 18: MATCHES TICKET PAGE | 60 |
| FIGURE47: SCREEN SHOT 19: ORGANIZER MY EVENT PAGE1..... | 61 |
| FIGURE48: SCREEN SHOT 20: ORGANIZER MY EVENT PAGE2..... | 61 |
| FIGURE49: SCREEN SHOT 21: ORGANIZER DASHBOARD | 62 |
| FIGURE50: SCREEN SHOT 22: VIEW MAIN EVENTS..... | 62 |
| FIGURE51: SCREEN SHOT 23: CREATE MAIN EVENT | 63 |
| FIGURE52: SCREEN SHOT 24: EDIT MAIN EVENT | 63 |
| FIGURE53: SCREEN SHOT 25: VIEW SUB EVENT | 64 |
| FIGURE54: SCREEN SHOT 26: CREATE SUB EVENT..... | 64 |
| FIGURE55: SCREEN SHOT 27: EDIT SUB EVENT | 65 |
| FIGURE56: SCREEN SHOT 28: CREATE NEW TICKET | 65 |
| FIGURE57: SCREEN SHOT 29: VIEW DIFFERENT TICKETS..... | 66 |
| FIGURE58: SCREEN SHOT 30: VIEW QUANTITY OF EACH TICKET | 66 |
| FIGURE59: SCREEN SHOT 31: CREATE TICKET TYPE | 67 |
| FIGURE60: SCREEN SHOT 32: VIEW ANALYTICS AND REPORTING | 67 |
| FIGURE61: SCREEN SHOT 33: VIEW ADD-ONS | 68 |
| FIGURE62: SCREEN SHOT 34: CREATE ADD-ONS..... | 68 |
| FIGURE63: SCREEN SHOT 35: VIEW BLOGS | 69 |
| FIGURE64: SCREEN SHOT 36: CREATE BLOGS..... | 69 |
| FIGURE65: SCREEN SHOT 37: VIEW QUESTION FORM..... | 70 |
| FIGURE66: SCREEN SHOT 38: CREATE QUESTION FORM..... | 70 |
| FIGURE67: SCREEN SHOT 39: CREATE FORM QUESTIONS | 71 |
| FIGURE68: SCREEN SHOT 40: VIEW FORM QUESTION..... | 71 |
| FIGURE69: SCREEN SHOT 41: VIEW ORDERS | 72 |
| FIGURE70: SCREEN SHOT 42: VIEW TICKET TEMPLATES | 72 |
| FIGURE71: SCREEN SHOT 43: VIEW TICKET TEMPLATES 1 | 73 |
| FIGURE72: SCREEN SHOT 44: VIEW TICKET TEMPLATES 2 | 73 |
| FIGURE73: SCREEN SHOT 45: VIEW TICKET TEMPLATES 3 | 74 |
| FIGURE74: SCREEN SHOT 46: VIEW TICKET TEMPLATES 4 | 74 |
| FIGURE75: SCREEN SHOT 47: VIEW USERS | 75 |
| FIGURE76: SCREEN SHOT 48: CREATE USERS..... | 75 |
| FIGURE77: SCREEN SHOT 49: EDIT USERS | 76 |
| FIGURE78: SCREEN SHOT 50: VIEW USERS | 76 |

| | |
|---|-----------|
| FIGURE79: SCREEN SHOT 51: EDIT USERS | 77 |
| FIGURE80: SCREEN SHOT 52: EDIT USERS | 77 |
| FIGURE81: SCREEN SHOT 53: CONFIRMATION EMAIL | 78 |
| FIGURE82: SCREEN SHOT 54: SEAT SELECTION | 79 |

CHAPTER – 1:

1.0. Introduction:

In today's fast-paced event Organization, efficiency and customization are key. Enter "Event Flow," an Online Ticketing Event Management System designed to transform event planning, execution, and experience. With its user-friendly features, Event Flow offers a seamless solution for organizers and attendees alike.

Event Flow puts organizers in control, allowing them to customize every aspect of ticketing to suit their needs. From ticket types and prices to capacity and discounts, organizers can tailor their events effortlessly. The system also simplifies communication by enabling organizers to send invitations directly from the platform.

Personalization is central to Event Flow. Organizers can create customized registration forms and automate confirmation emails, enhancing attendee engagement. The system also offers robust access control, ensuring a tailored experience for each attendee based on their ticket type.

Beyond traditional ticketing, Event Flow introduces advanced features like QR code-generated name badges for easy check-in and lead generation. Organizers can also assign seats and manage future events through a convenient calendar view.

Event Flow sets a new standard in event management, empowering organizers to create unforgettable experiences while streamlining the entire event lifecycle. The Online Ticketing Event Management System project aims to revolutionize event organization, providing a modern and intuitive platform for managing events seamlessly. Leveraging web-based technology, this system enhances event planning, ticket sales, attendee management, and promotion, ultimately elevating the overall event experience for organizers and attendees alike.

1.1. Problem Statement:

Traditional event management processes are often cumbersome and inefficient, leading to challenges such as manual ticket sales, limited promotional reach, and difficulty in managing attendee information. Organizers face hurdles in effectively customizing ticket types, managing registrations, and ensuring seamless access control during events. Additionally, the lack of flexible payment options and purchase protection measures can deter potential attendees from purchasing tickets.

1.2. Motivation:

The motivation behind the Online Ticketing Event Management System project stems from the need to address the shortcomings of traditional event management practices and embrace modern technology to enhance efficiency and attendee satisfaction. By providing a comprehensive solution that automates various aspects of event organization, the project aims to empower organizers to streamline their processes, maximize ticket sales, and deliver exceptional event experiences. Moreover, the project aims to cater to the growing demand for secure and flexible online ticketing solutions, especially in light of the evolving needs of event attendees.

1.3. Objective:

The main goal of the Online Ticketing Event Management System project is to simplify event organization and ticketing processes for both organizers and attendees. Specifically, the objectives include:

1.3.1. Streamlining ticket customization:

Make it easy for organizers to set prices, define ticket capacities, and offer discounts and add-ons to attract attendees.

1.3.2. Enhancing promotional efforts:

Enable organizers to create personalized invitation campaigns and send them directly from the system to reach a wider audience.

1.3.3. Improving attendee registration experience:

Allow organizers to customize registration question forms and confirmation emails to collect relevant information and provide essential details to attendees.

1.3.4. Ensuring access control:

Control access to event content based on ticket types purchased and implement proper validation to prevent unauthorized access.

1.3.5. Simplifying check-in process:

Generate name badges with personal QR codes for attendees to facilitate efficient check-in and lead generation.

1.3.6. Facilitating seat assignment:

Provide options for organizers to assign seats for individuals and groups, and manage seat availability effectively.

1.3.7. Providing flexible payment options:

Support Wallet payment methods, which can lead to easily refunds, to offer attendees more flexibility.

1.3.8. Ensuring purchase protection:

Implement features to handle refunds and cancellations effectively, including purchase protection for attendees.

1.4. Challenges:

1.4.1. Limited technology literacy:

Many people faced challenges due to a lack of familiarity with smartphones and web-based platforms, making it difficult for them to navigate and utilize the online ticketing system.

1.4.2. Technical difficulties during development:

The team encountered various challenges while creating the system, such as ensuring compatibility across different devices and optimizing performance to minimize storage and memory usage.

1.4.3. Managing event logistics:

Organizers often struggled with coordinating various aspects of event logistics, including ticket sales, registration, access control, and attendee management, which required careful planning and coordination.

1.4.4. Security concerns:

Ensuring the security and reliability of payment processing and data handling posed significant challenges, requiring robust encryption and validation measures to protect sensitive information.

1.4.5. Meeting user requirements:

Balancing the need for advanced features with user-friendly design and intuitive functionality was a challenge, as the system needed to cater to both organizers and attendees with diverse needs and preferences.

1.5. Structure of Report:

This marks the completion of chapter one of the Event Flow (OTEM). This chapter highlighted the overall concept of the Event Flow. The introduction section provides a detailed review regarding major details of idea hunting. The problem statement specifies what kind of issue we are tackling with this web application and how it will be beneficial for the society. The motivation heading provides a clear definition of what motivated us to come up with the idea of Event Flow. As we go further into the content of chapter one the major descriptions regarding the objective of the project can be seen. This is where you will find vital objectives.

Furthermore, we have provided content regarding the challenges that can occur with the progression of the web application as well as in the usage of the web application. Lastly, the remaining structure of the project report is given as follows:

1.5.1. Chapter 2: Technology Background:

This chapter will consist of our well-researched literature review regarding all the related prior work of our project subject and technologies.

1.5.2. Chapter 3: Requirements & Methodology:

This Chapter will discuss basic models of the system; in addition to that, the chapter will also host functional and nonfunctional requirements of our project.

1.5.3. Chapter 4: Project Plan & Initial Design:

This chapter will consist of all the detailed designs of the project that will help the developer in understanding the project implementation and creating an easy route in development of the system.

1.5.4. Chapter 5: Project Design & Development:

This is the most significant chapter since it details the actual design and implementation of the concept. i.e., the phases of design and development.

1.5.5. Chapter 6: Testing:

We will construct test cases in this chapter:

- i. Perform front end (design testing), which may include user control testing, spelling checks, and alignment, among other things.
- ii. Carry out backend testing (source code)
- iii. Use a tool to conduct testing and incorporate the results in report.

1.5.6. Chapter 7: Conclusion:

In this last chapter, we will conclude our work, share results including facts and figures, tables, and graphs to show your findings.

- i Discuss limitations and challenges.
- ii Discuss the work that will be done in the future.

CHAPTER – 2:

2.0. Technology Overview:

In this section, we will delve into the technologies utilized in our Online Ticketing Event Management System web application. We will provide an overview of the technologies employed, their functionalities, and the benefits they bring to our project.

2.0.1. Technology Stack:

Our web application is built using a combination of HTML, CSS, JavaScript, PHP, and MySQL. Each technology plays a crucial role in the functionality and user experience of our event management system.

HTML (Hypertext Markup Language):

HTML is used to structure the content of our web pages, defining the layout and hierarchy of elements. It forms the backbone of our web application's frontend.

Advantages of HTML:

- Provides a standardized markup language for creating web pages, ensuring cross-browser compatibility.
- Enables semantic markup, making the content accessible to search engines and assistive technologies.
- Supports the integration of multimedia elements, enhancing the interactivity and engagement of the application.

CSS (Cascading Style Sheets):

CSS is utilized to style and design the user interface of our web application. It allows us to create visually appealing and responsive layouts, enhancing the overall user experience.

Advantages of CSS:

- Enables consistent styling across the application.
- Facilitates responsive design for various devices and screen sizes.
- Helps in improving the visual appeal of the application, thus attracting more users.

JavaScript:

JavaScript is employed for client-side scripting, adding interactivity and dynamic behavior to our web application. It enhances user interaction and enables real-time updates without page reloads.

Advantages of JavaScript:

- Enables the creation of responsive and interactive user interfaces, enhancing user engagement.
- Facilitates asynchronous data retrieval and processing, improving the performance and responsiveness of the application.
- Supports the integration of third-party libraries and frameworks, expanding the functionality and capabilities of our web application.

PHP (Hypertext Preprocessor):

PHP is used as the server-side scripting language in our project. It handles dynamic content generation, database interactions, and server-side processing of user requests.

Advantages of PHP:

- Provides robust server-side functionality for processing forms, handling user authentication, and managing database operations.
- Offers excellent compatibility with MySQL database, making data management efficient and secure.
- Supports modular development, allowing for easy integration of new features and functionalities.

MySQL:

MySQL is employed as the relational database management system (RDBMS) to store and manage event-related data, including user information, ticket details, and event schedules.

Advantages of MySQL:

- Offers efficient data storage and retrieval capabilities, ensuring fast performance even with large datasets.
- Provides strong data security features, including encryption and access control, to safeguard sensitive information.
- Supports transactions and ACID properties, ensuring data integrity and consistency.

2.0.2. Web Application Overview:

Our Online Ticketing Event Management System web application features a user-friendly interface for event organizers and attendees. It allows organizers to create, promote, and manage events seamlessly, while attendees can securely purchase tickets, receive electronic tickets, and stay updated with event notifications.

The web application includes features such as customizable ticket types, invitation campaigns, registration forms, seat assignment, payment options, and purchase protection. It prioritizes security, scalability, and customization to provide a reliable and efficient solution for event management.

Overall, our web application aims to automate various aspects of event management, reduce manual efforts, improve attendee satisfaction, and enable organizers to successfully plan, promote, and execute events.

2.1. Literature Review: Online Ticketing Event Management Systems

In recent years, the event management industry has witnessed a significant transformation due to technological advancements, leading to the emergence of Online Ticketing Event Management Systems (OTEMS). These systems have become essential tools for event organizers, offering a comprehensive platform to efficiently plan, organize, and manage events while providing attendees with a seamless ticketing experience. In this literature review, we delve into the evolution of OTEMs, their key features, benefits, challenges, and future prospects.

The concept of online ticketing and event management dates back to the early 2000s, coinciding with the rise of e-commerce platforms and online booking systems. Initially, these systems offered basic functionalities such as ticket sales and seat reservations. However, with advancements in technology and the growing demand for sophisticated event management solutions, OTEMs have evolved significantly over time.

Today, OTEMs encompass a wide range of features and capabilities, including ticket customization, registration management, access control, payment processing, promotional tools, and analytics. These systems leverage web-based technologies such as HTML, CSS, JavaScript, MySQL, and PHP to provide users with an intuitive and seamless experience.

2.1.1. Literature Review:

Our literature review will explore existing literature and platforms related to online ticketing and event management systems. Additionally, we will compare our project with popular platforms such as Eventbrite, Ticketwala, Book me, and Bookkaru to identify similarities, differences, strengths, and weaknesses.

2.1.2. Introduction:

Online ticketing and event management systems have become indispensable tools for organizers seeking efficient event planning and execution. These platforms offer convenience for both organizers and attendees, facilitating seamless ticket purchase, event promotion, and attendee management.

2.1.3. Existing Platforms:

Several platforms dominate the online ticketing and event management industry, including Eventbrite, Ticketwala, Book me, and Bookkaru. These platforms offer a variety of features and services to cater to the diverse needs of event organizers and attendees.

Eventbrite:

Eventbrite is a leading online ticketing platform globally, providing event organizers with a comprehensive suite of tools for event planning, promotion, and ticket sales. Attendees can easily discover and purchase tickets for a wide range of events, from concerts to conferences.

Comparison with Our Project:

While EventBrite offers similar features to our project, such as customizable ticket types and event promotion tools, our project prioritizes security, scalability, and customization, which may provide a more robust solution for event organizers.

Ticketwala:

Ticketwala is a growing online ticketing platform known for its user-friendly interface and affordable ticket prices. It caters to various events, including movies, sports, and cultural performances, focusing on providing a seamless ticket purchasing experience.

Comparison with Our Project:

While Ticketwala shares similarities with our project in terms of user interface design and ticket purchasing functionalities, our project offers additional features such as invitation campaigns and purchase protection, enhancing the overall event management experience.

Book me:

Book me is a popular online ticketing platform in the entertainment industry, offering streamlined ticket booking for movies, theaters, and concerts. It emphasizes real-time seat availability and secure payment options to simplify the ticket purchasing process.

Comparison with Our Project:

Although Book me specializes in entertainment events, our project caters to a broader range of events and offers advanced features such as seat assignment and payment options, making it a comprehensive solution for event organizers.

Bookkaru:

Bookkaru is a niche online ticketing platform focusing on travel and tourism-related events, providing travelers with a convenient way to book tickets for tours and activities. It prioritizes user experience and accessibility in its platform design.

Comparison with Our Project

While our project does not focus specifically on travel-related events like Bookkaru, it offers similar features such as customizable ticket types and event promotion tools, making it a versatile solution for event organizers across various industries.

2.1.4. Comparison and Conclusion:

In conclusion, our project aims to provide a comprehensive and user-friendly platform for event organizers to efficiently manage their events and ticketing process. While existing platforms offer similar features and services, our project distinguishes itself through its emphasis on security, scalability, and customization. By comparing our project with these platforms, we can identify opportunities for improvement and innovation in the online ticketing and event management industry. Our project seeks to address the limitations of existing platforms and provide a reliable and efficient solution for event organizers and attendees alike.

CHAPTER – 3

3.0. Introduction:

In this chapter, we will discuss about how much work is done on the development of our project according to the project plan. This chapter will cover the in-detail process and objective of the project. As our web application is built on windows that are being reused within the system itself, the developers had to take a systematic approach for the web application to work smoothly. Our project plan is strategically planned with the Gantt chart and other organizational tools. Each activity has specific time period allotted according to the complexity of the task, which is why the days in work may vary.

We will also discuss in detail about the Functional, Non-Functional and Hardware requirements of our project. The functional requirement are taken in to full consideration as they are the necessary part in order to get the basic requirement by the project such as, ticket customization, invitation campaigns, registration question form, confirmation email, access control, name badges & QR Code, seat assignment, calendar view, payment options, purchase protection, ticket reservations, survey questions, while on the other hand, the non-functional requirement such as, performance, scalability, security, reliability, usability, integration are also thoroughly planned.

3.1. Project Plan:

| FINAL YEAR PROJECT Schedule | | | | | |
|-----------------------------|----------------------------------|------------|------------|---|--|
| Task # | Task Description | Start date | End date | Deliverables | Features |
| 1 | Research & Planning | 25/10/2023 | 31/10/2023 | Project requirements, and design outline | Project Initiation |
| 2 | Database design & Setup | 1/11/2023 | 7/11/2023 | Completed database schema | MySQL, Database Setup |
| 3 | Dashboard for users | 8/11/2023 | 21/11/2023 | Dashboard for Admin, Organizer and Attendee (Completed) | Dashboard for Users |
| 7 | Ticket page & Ticket types | 22/11/2023 | 28/11/2023 | Ticket view, types(completed) | Ticket customization with guest reservation and ticket types |
| 4 | Creating Events | 29/11/2023 | 5/12/2023 | Events dashboard for Organizers(Completed) | create ,edit ,delete event access for Organizer |
| 5 | Calendar View | 6/11/2023 | 12/12/2023 | Added calendar view(completed) | Calendar View |
| 6 | Registration Form | 13/12/2023 | 19/12/2023 | Event registration form (completed) | Registration Form |
| 8 | E-ticket & name badges & QR code | 20/12/2023 | 26/12/2023 | E-ticket & implement Name Badges(completed) | Access to E-tickets & have a name badge & QR code on it. |
| 9 | Analytics and report | 27/12/2023 | 2/1/2024 | Make Analytics & Reporting | Make Analytics & reporting Dashboard for organizer help in future events |
| 10 | Survey question | 3/1/2024 | 9/1/2024 | Survey for attendee feedback | Feedback for future plans |
| 11 | Wallets | 10/1/2024 | 16/1/2024 | Wallet for Organizer and Attendee | Refund |
| 12 | Testing & Debugging | 17/1/2024 | 23/1/2024 | Improvements and Enhancements | Make the project better |
| 13 | Deployment | 24/1/2024 | 30/1/2024 | FULL Website | |

Table.1: Project Plan

3.2. Gant Chart:

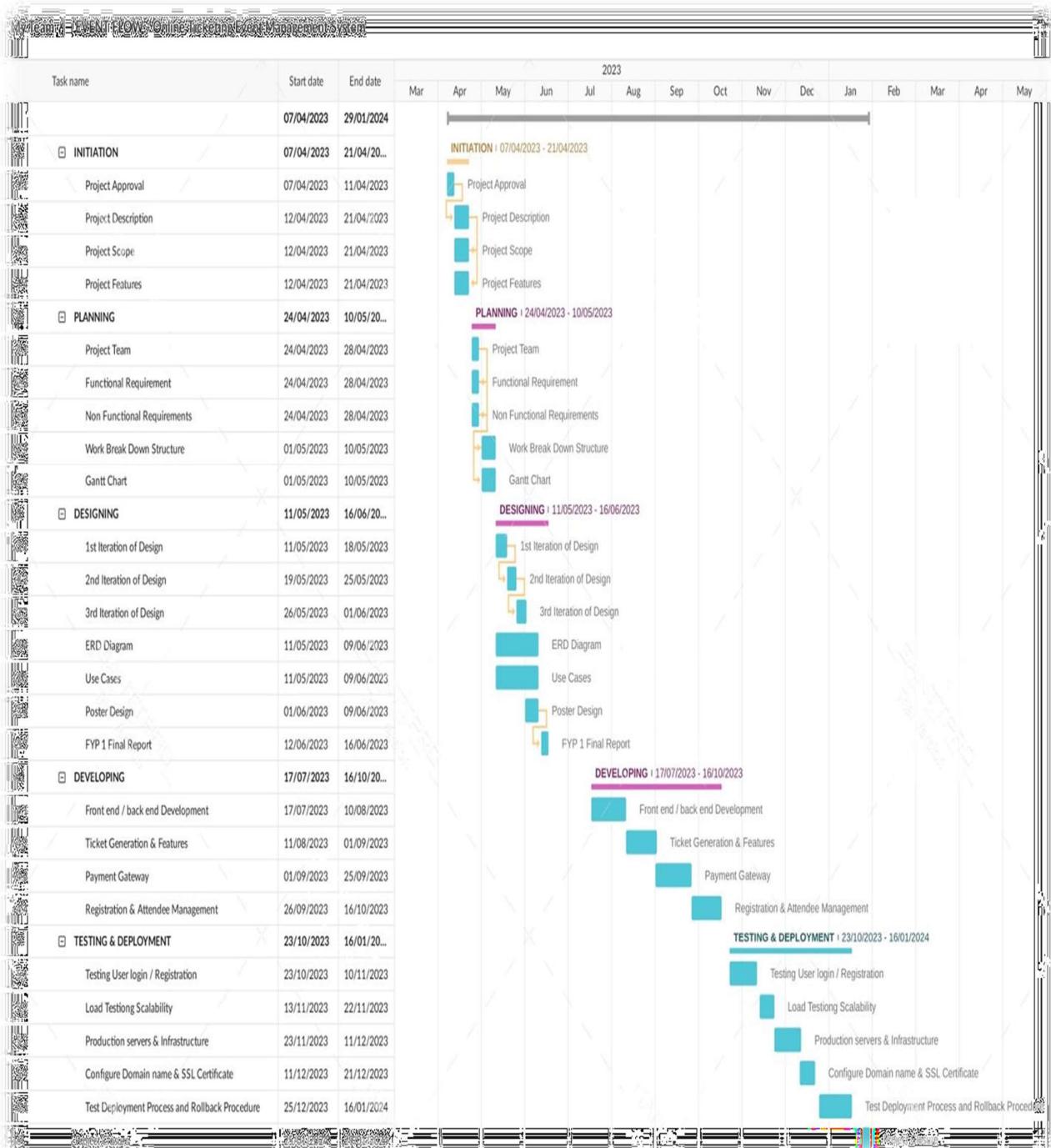


Figure.1: Gant Chart

3.3. Functional Requirements:

3.3.1. TICKET CUSTOMIZATION:

- Set prices for each ticket type with the flexibility to change them based on demand or promotional offers.
- Define ticket capacity for each ticket type to limit the number of tickets available for purchase.
- Enable discounts and add-ons to be applied to specific ticket types or for a limited time period.

3.3.2. INVITATION CAMPAIGNS:

- Enable personalized invitation campaigns with the ability to customize email content
- Allow event organizers to send invitation campaigns directly from the system.

3.3.3. REGISTRATION QUESTION FORM & CONFIRMATION EMAIL:

- Allow event organizers to create a registration question form to collect relevant information from attendees during registration.
- Provide options for organizers to customize automatic confirmation email for each ticket purchased, including ticket details, event information, and payment confirmation.

3.3.4. ACCESS CONTROL:

- Control access to event content based on the type of tickets purchased, such as VIP tickets granting access to exclusive areas or content.
- Ensure proper validation and verification of tickets to prevent unauthorized access.

3.3.5. NAME BADGES & QR CODE:

- Generate name badges/Ticket ID for attendees with personal QR codes for efficient check-in and lead generation.

3.3.6. SEAT ASSIGNMENT:

- Allow event organizers to assign seats for both individuals and groups when tickets are purchased.
- Provide options for organizers to manage seat availability.

3.3.7. CALENDAR VIEW:

- Provide a calendar view that includes future events to allow attendees to easily browse and select events of interest.
- Display event details, dates, and ticket availability in the calendar view.

3.3.8. PAYMENT OPTIONS:

- Support multiple payment options besides credit cards, such as PayPal, to provide flexibility for attendees.
- Ensure secure and reliable payment processing with proper encryption and validation.

3.3.9. PURCHASE PROTECTION

- Provide features for organizers to handle refunds and cancellations effectively, including purchase protection for attendees.

3.3.10: TICKET RESERVATIONS:

- Allow event organizers to reserve tickets for guests and VIPs
- Provide options for organizers to manage ticket reservations, allocations, and guest lists.

3.3.12. SURVEY QUESTIONS:

- Allow organizers to add survey questions during registration to collect feedback or gather additional information from attendees.
- Provide options for organizers to customize survey questions and manage responses.

3.4. Non-Functional Requirements:

3.4.1. PERFORMANCE

- System will be able to handle a minimum of 7,000 simultaneous users without any significant decrease in performance.
- The average response time for each page will be less than 10 seconds.
- The system will be able to handle a minimum of 50,000 ticket transactions per day.

3.4.2. SCALABILITY:

- The system will be able to handle a minimum of 100 new events added per month.
- The system will be able to handle a minimum of 33% growth in the number of users and transactions per year.

3.4.3. SECURITY:

- The system will have a minimum of 97% uptime to ensure continuous availability to users.
- The system will be able to recover from any failure or disaster within 4 hours.

3.4.4. RELIABILITY:

- The system will be able to process transactions with a minimum of 98% accuracy
- The system will be able to handle a minimum of 99% of all tickets scanned successfully at events.

3.4.5. USABILITY:

- The system will have a user-friendly interface, and users will be able to complete common tasks without requiring extensive training.
- The system will be compatible with common web browsers and devices to ensure broad accessibility.

3.4.6. INTEGRATION:

- Support integration with other relevant systems or platforms, such as payment gateways, CRM systems, email-marketing tools.

3.5. Hardware Requirements:

It is a web application so it does not need any specialized tool just normal hardware can be used to run this web application.

- a. Can be accessible from android/ desktop through google/ chrome.
- b. Need a continuous and good internet connection to operate it.
- c. A QR code scanner must be needed for scanning tickets.

3.6. Summary:

In this chapter, a detailed Project Plan, Functional, Non-Functional requirements and other planning mechanisms are discussed in detail that will be required in our project. We have also mentioned an introduction regarding our web application how we can perform our task so we make a milestone chart in the first we describe our task week wise in summary activity and then we make a Gantt chart according to summary activity. In Gantt chart we were describing task name or duration for implementation of our “Event Flow web application” After Gantt chart we describe the functional and Non-Functional requirements which we need to achieve and the hardware we required to implement our project.

CHAPTER – 4

4.0. Introduction:

In this chapter, we are going to discuss about the design and specification of our project, in which we elaborate our project deeply with the help of diagrams like we gather all the information related to our web application then set the framework to show the flow of the web application so that the its flow will be easily understand. We have used different diagrams for the complete flow of our web application to make it understandable to user. In this phase, we also discussing the State diagrams, entity relationship diagram (ERD'S), UML in detailed as according to our web application. These diagrams show the system workflow and specification of our web application to make it user friendly. It also shows how every screen workflow is working with the help of diagram. After all information has been gathered and design has been created so now, the development has started in order to make sure that user can use it.

The purpose for making the data flow diagram and entity diagram to guide the direction of our system that how we perform each and every thing and also show the flow of our web application specifically like in implementation. In detail, this will provide a clear understanding of the overall coding of the system for the people who are on user bases. each diagram is detail with all functional input and output of the system, making sure that the system runs smoothly.

4.1. State Diagram:

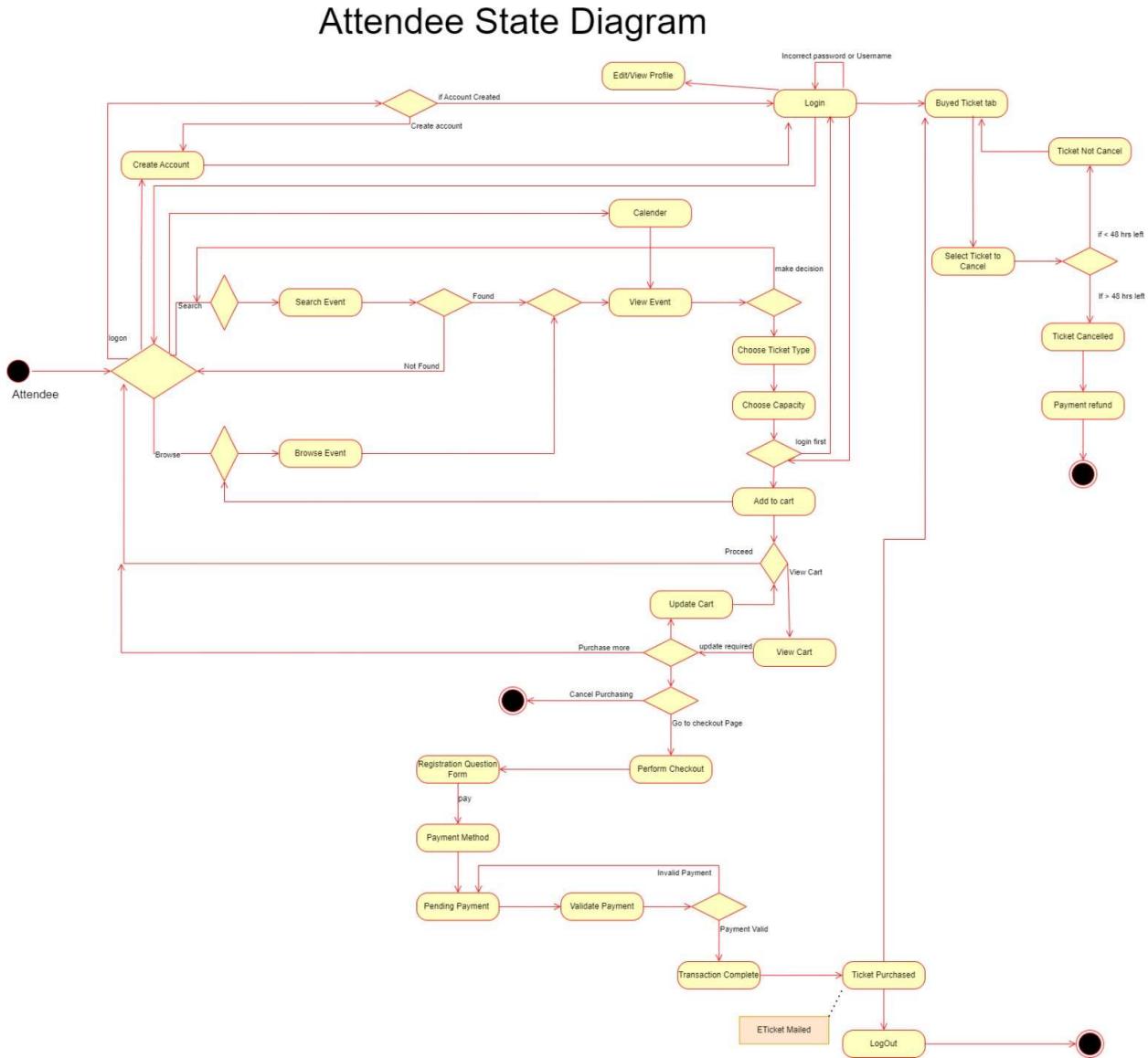


Figure 2: Attendee State Diagram

Organizer State Diagram

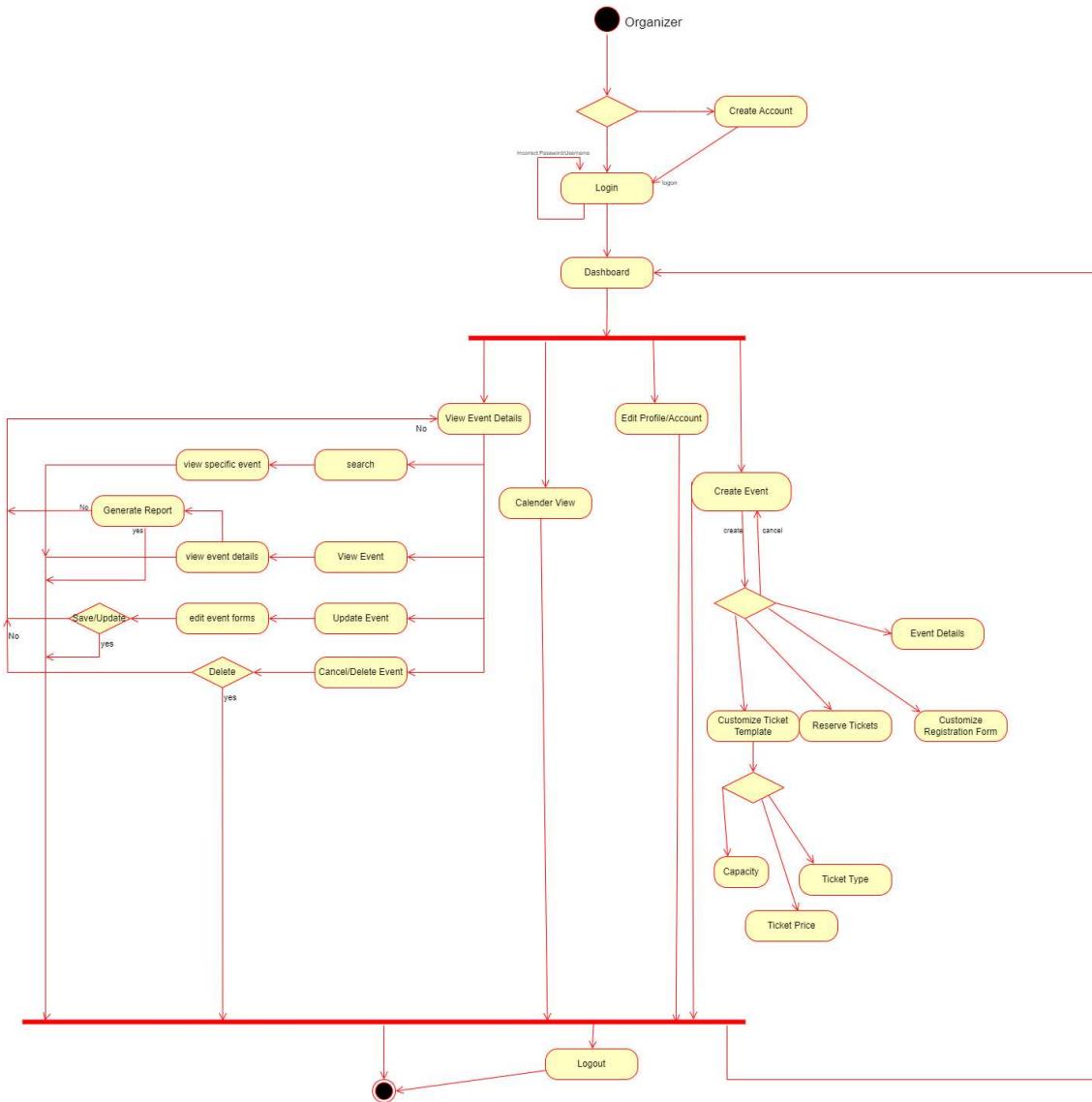


Figure 3: Organizer State Diagram

4.2. Entity Relationship Diagram:

EVENT FLOW : Online Ticketing Event Management System

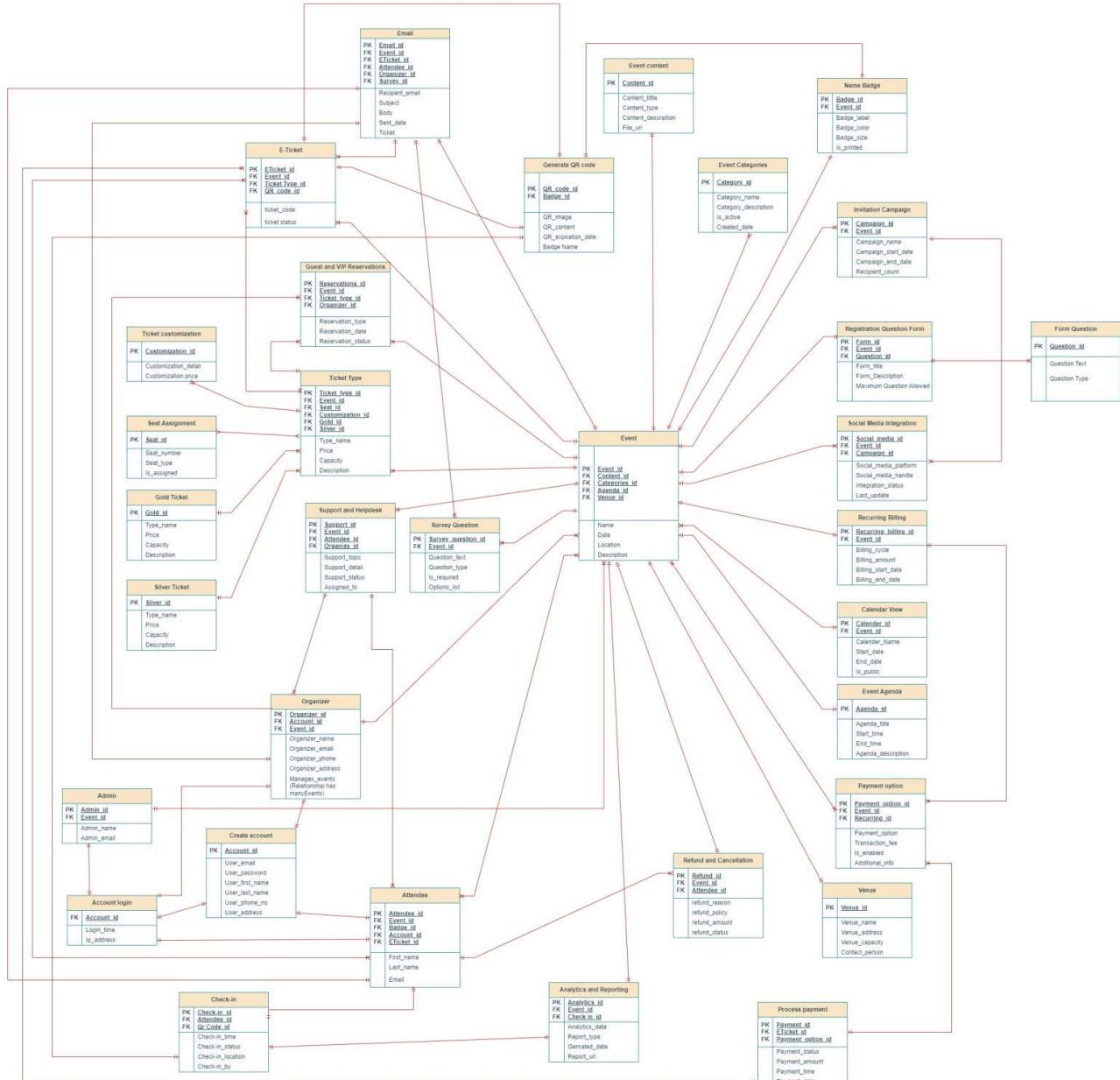


Figure.4: Enhanced Entity Relationship Diagram

4.3. Use Case diagram:

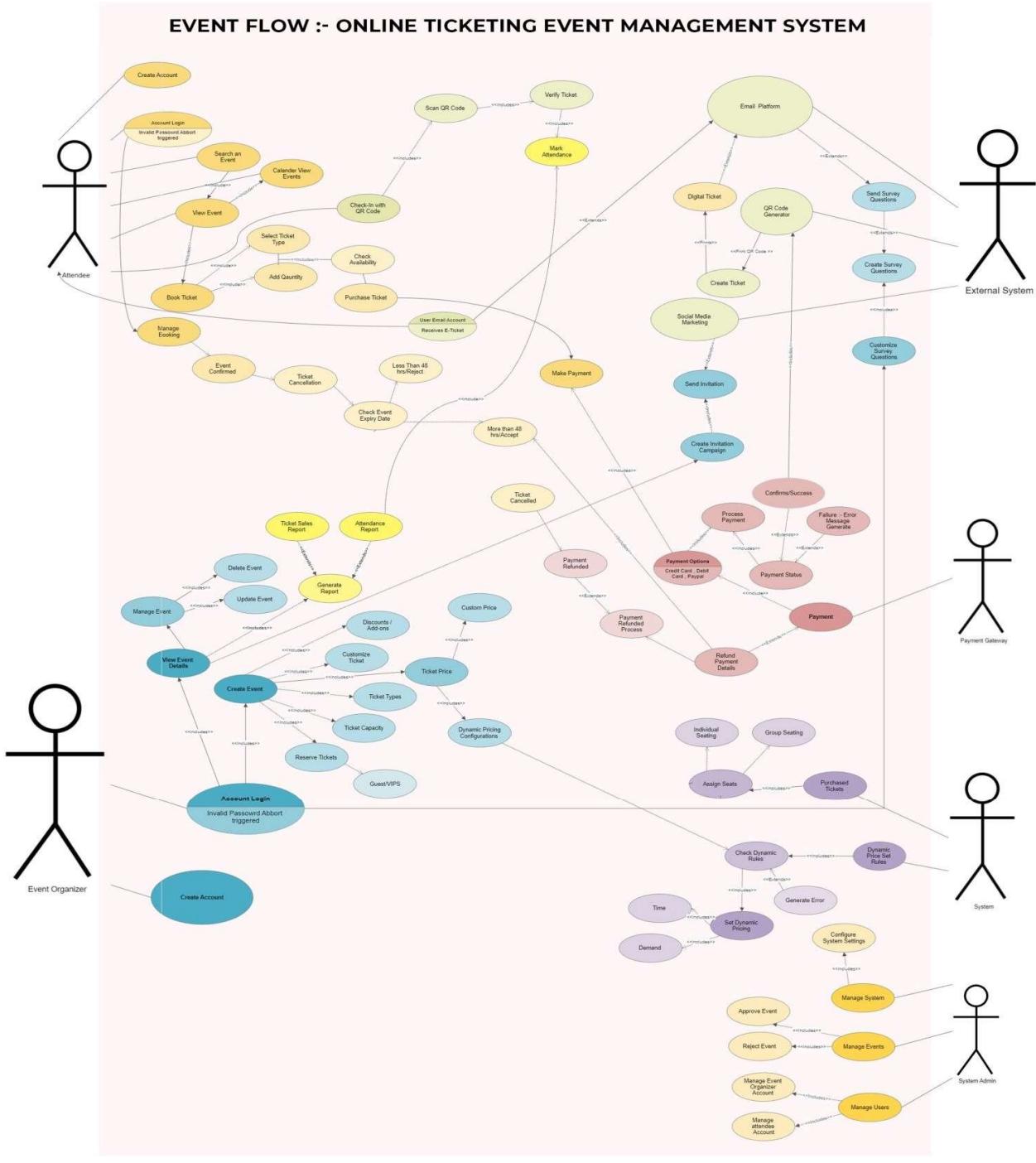


Figure.5: Use Case Diagram

4.4. Use Cases:

Actor = Attendee Use Cases

| | | |
|---------------------------------------|---|---|
| 1.1.1 Create Account/Register Account | | |
| Use Case Name: | Create Account/Register Account | |
| ID: | UC101 | |
| Actors Involved: | Attendee | |
| Brief Description: | Process of creating a new account for an attendee. | |
| Pre-Condition: | None. | |
| Post-Condition: | The attendee has a registered account. | |
| Normal Flow of Events: | Actor Action The attendee provides the necessary information to create a new account. | System Response The system validates the provided information and creates a new account for the attendee. |

Table 2: Use Case1=Create Account

| | | |
|-------------------------------|---|---|
| 1.1.2 Login Account | | |
| Use Case Name: | Login Account | |
| ID: | UC102 | |
| Actors Involved: | Attendee | |
| Brief Description: | Process of logging in to the attendee's account. | |
| Pre-Condition: | The attendee must have a registered account. | |
| Post-Condition: | The attendee is logged in and gains access to their account. | |
| Normal Flow of Events: | Actor Action The attendee provides their login credentials. | System Response The system verifies the provided credentials and grants access to the attendee's account. |

Table 3: Use Case2=Login Account

| | | |
|-------------------------------|---|---|
| 1.1.3 View Event | | |
| Use Case Name: | View Event | |
| ID: | UC103 | |
| Actors Involved: | Attendee | |
| Brief Description: | Process of viewing event details. | |
| Pre-Condition: | None | |
| Post-Condition: | The attendee can view the event details. | |
| Normal Flow of Events: | Actor Action The attendee selects an event to view. | System Response The system displays the event details, including date, time, location, description. |

Table 4: Use Case3=View Event

| | | |
|-------------------------------|---|--|
| 1.1.4 Calendar View | | |
| Use Case Name: | <u>Calendar View</u> | |
| ID: | UC104 | |
| Actors Involved: | Attendee | |
| Brief Description: | Process of viewing a calendar displaying future events. | |
| Pre-Condition: | The attendee must be logged in to their account. | |
| Post-Condition: | The attendee can see the calendar with upcoming events. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The attendee selects the calendar option. | The system presents a calendar view with the dates and names of future events. |

Table 5: Use Case4=Calendar View

| | | |
|-------------------------------|---|---|
| 1.1.5 Select Ticket Type | | |
| Use Case Name: | <u>Select Ticket Type</u> | |
| ID: | UC105 | |
| Actors Involved: | Attendee | |
| Brief Description: | Process of selecting the type of ticket for an event. | |
| Pre-Condition: | 1) The attendee must be logged in to their account. 2) The attendee must be viewing the event details. | |
| Post-Condition: | The attendee selects the desired ticket type. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The attendee chooses a ticket type from the available options. | The system records the selected ticket type for the attendee. |

Table 6: Use Case5=Ticket Type

| | | |
|-------------------------------|--|--|
| 1.1.6 Make Payment | | |
| Use Case Name: | <u>Make Payment</u> | |
| ID: | UC106 | |
| Actors Involved: | Attendee, Payment Gateway | |
| Brief Description: | Process of making a payment for the selected ticket. | |
| Pre-Condition: | The attendee must have selected a ticket type. | |
| Post-Condition: | The attendee's payment is Processed. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The attendee provides payment details. | The system securely sends the payment details to the payment gateway for Processing. |

Table 7: Use Case6=Payment

| | | |
|-------------------------------|--|--|
| 1.1.7 E-Ticket | | |
| Use Case Name: | E-Ticket | |
| ID: | UC107 | |
| Actors Involved: | Attendee, Email Platform | |
| Brief Description: | Process of receiving the E-ticket via email. | |
| Pre-Condition: | The attendee's payment must be successfully Processed. | |
| Post-Condition: | The attendee receives the E-ticket in their email. | |
| Normal Flow of Events: | Actor Action | System Response |
| | None. | The system generates the E-ticket and sends it to the attendee's email address via the email platform. |

Table 8: Use Case7=E-Ticket

| | | |
|-------------------------------|--|---|
| 1.1.8 Payment Refund | | |
| Use Case Name: | Payment Refund | |
| ID: | UC108 | |
| Actors Involved: | Attendee, Payment Gateway | |
| Brief Description: | Process of receiving a refund for a ticket. | |
| Pre-Condition: | The attendee must have requested a refund. | |
| Post-Condition: | The attendee's refund payment is Processed and sent to their bank account. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The attendee requests a refund. | The system initiates the refund Process with the payment gateway, which transfers the refund amount to the attendee's bank account. |

Table 9: Use Case8=Refund

| | | |
|-------------------------------|--|---|
| 1.1.9 Save Payment Details | | |
| Use Case Name: | Save Payment Details | |
| ID: | Attendee | |
| Actors Involved: | UC109 | |
| Brief Description: | Process of saving the attendee's payment details for future use. | |
| Pre-Condition: | The attendee must have made a successful payment. | |
| Post-Condition: | The attendee's payment details are securely stored in their account. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The attendee selects the option to save payment details. | The system securely stores the attendee's payment details in their account for future transactions. |

Table 10: Use Case9=Payment Detail

| | | |
|-------------------------------|---|--|
| 1.1.10 Check-in with QR Code | | |
| Use Case Name: | <u>Check-in with QR Code</u> | |
| ID: | UC110 | |
| Actors Involved: | Attendee, External System (QR Code Generator) | |
| Brief Description: | Process of checking in to an event using a QR code. | |
| Pre-Condition: | The event must be ongoing, and the attendee must have their E-ticket. | |
| Post-Condition: | Verified Ticket | |
| Normal Flow of Events: | Actor Action | System Response |
| | The attendee presents their E-ticket with the QR code. | The system uses an external QR code generator to scan the QR code. |

Table 11: Use Case10=check-in

| | | |
|-------------------------------|---|--|
| 1.1.11 Mark Attendance | | |
| Use Case Name: | <u>Mark Attendance</u> | |
| ID: | UC111 | |
| Actors Involved: | Attendee, External System (QR Code Generator) | |
| Brief Description: | Process of marking attendance using QR code Scanning. | |
| Pre-Condition: | The event must be ongoing, and the attendee must have their E-ticket. | |
| Post-Condition: | The attendee's attendance is marked. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The attendee presents their E-ticket with the QR code. | The system uses an external QRcode generator to scan the QR code and mark the attendee's attendance. |

Table 12: Use Case11=Attendance

Actor = Organizer Use Cases

| | | |
|---------------------------------------|---|--|
| 1.2.1 Create Account/Register Account | | |
| Use Case Name: | <u>Create Account/Register Account</u> | |
| ID: | UC201 | |
| Actors Involved: | Event Organizer | |
| Brief Description: | Process of creating a new account for an event organizer. | |
| Pre-Condition: | None. | |
| Post-Condition: | The event organizer has a registered account. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer provides the necessary information to create a new account. | The system validates the provided information and creates a new account for the event organizer. |

Table 13: Use Case12=Register Account

| | | |
|-------------------------------|---|--|
| 1.2.2 Login Account | | |
| Use Case Name: | <u>Login Account</u> | |
| ID: | UC202 | |
| Actors Involved: | Event Organizer | |
| Brief Description: | Process of logging in to the event organizer's account. | |
| Pre-Condition: | The event organizer must have a registered account. | |
| Post-Condition: | The event organizer is logged in and gains access to their account. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer provides their login credentials. | The system verifies the provided credentials and grants access to the event organizer's account. |

Table 14: Use Case13=Login Account

| | | |
|-------------------------------|--|---|
| 1.2.3 Customize Ticket | | |
| Use Case Name: | <u>Customize Ticket</u> | |
| ID: | UC203 | |
| Actors Involved: | Event Organizer | |
| Brief Description: | Process of customizing ticket options, such as types, capacity, discounts, and dynamic pricing | |
| Pre-Condition: | The event organizer must be logged in to their account. | |
| Post-Condition: | The ticket options are customized according to the event organizer's preferences. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer selects the desired customization options for the tickets. | The system applies the selected customization options to the ticket settings for the event. |

Table 15: Use Case14=Ticket Customization

| | | |
|-------------------------------|---|---|
| 1.2.4 Reserve Ticket | | |
| Use Case Name: | <u>Reserve Ticket</u> | |
| ID: | UC204 | |
| Actors Involved: | Event Organizer | |
| Brief Description: | Process of reserving tickets specifically for guests and VIPs. | |
| Pre-Condition: | The event organizer must be logged in to their account. | |
| Post-Condition: | Guest and VIP tickets are reserved separately. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer selects the number of guest and VIP tickets to reserve. | The system sets aside the specified number of tickets for guests and VIPs, separate from the regular ticket pool. |

Table 16: Use Case15=Reserved Ticket

| | | |
|-------------------------------|--|---|
| 1.2.5 Create Event | | |
| Use Case Name: | Create Event | |
| ID: | UC205 | |
| Actors Involved: | Event Organizer | |
| Brief Description: | Process of creating a new event. | |
| Pre-Condition: | The event organizer must be logged in to their account. | |
| Post-Condition: | The new event is created and listed in the system. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer provides the necessary information to create the event, such as event name, date, time, location, description, and ticket details. | The system validates the provided information and creates a new event listing with the specified details. |

Table 17: Use Case16=Create Event

| | | |
|-------------------------------|---|--|
| 1.2.6 Generate Report | | |
| Use Case Name: | Generate Report | |
| ID: | UC206 | |
| Actors Involved: | Event Organizer | |
| Brief Description: | Process of generating reports related to ticket sales and attendee attendance. | |
| Pre-Condition: | The event organizer must be logged in to their account and have access to event data. | |
| Post-Condition: | The event organizer obtains the requested reports. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer selects the desired report type, such as ticket sales or attendee attendance. | The system processes the request and generates the requested report based on the available event data. |

Table 18: Use Case17=Generate Report

| | | |
|-------------------------------|--|--|
| 1.2.7 View Event Details | | |
| Use Case Name: | View Event Details | |
| ID: | UC207 | |
| Actors Involved: | Event Organizer | |
| Brief Description: | Process of viewing event details for a specific event. | |
| Pre-Condition: | The event organizer must be logged in to their account. | |
| Post-Condition: | The event organizer can view the detailed information of the selected event. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer selects an event to view. | The system displays the event details, including date, time, location, description, ticket types, and current ticket availability. |

Table 19: Use Case18=View Event Detail

| | | |
|-------------------------------|---|--|
| 1.2.8 Edit Event Details | | |
| Use Case Name: | <u>View Event Details</u> | |
| ID: | UC208 | |
| Actors Involved: | Event Organizer | |
| Brief Description: | Process of managing an event, including updating or deleting it. | |
| Pre-Condition: | The event organizer must be logged in to their account and have created the event. | |
| Post-Condition: | The event details are updated or the event is removed from the system. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer selects the option to manage an event and chooses to update or delete it. | The system allows the event organizer to modify the event details or permanently remove the event from the system, based on the chosen action. |

Table 20: Use Case19=Edit Event Detail

| | | |
|-------------------------------|--|---|
| 1.2.9 Get Payment | | |
| Use Case Name: | <u>Get Payment</u> | |
| ID: | UC209 | |
| Actors Involved: | Event Organizer, Payment Gateway | |
| Brief Description: | Process of receiving payments for ticket sales through the event organizer's bank account. | |
| Pre-Condition: | The event organizer must have successfully sold tickets. | |
| Post-Condition: | The payment for ticket sales is processed and deposited into the event organizer's bank account. | |
| Normal Flow of Events: | Actor Action | System Response |
| | Request for payment. | The system facilitates the payment processing through the selected payment gateway, which transfers the funds to the event organizer's account. |

Table 21: Use Case20=Get Payment

| | | |
|-------------------------------|--|--|
| 1.2.10 Invitations Campaigns | | |
| Use Case Name: | <u>Invitations Campaigns</u> | |
| ID: | UC210 | |
| Actors Involved: | Event Organizer, External System (Social Media Platform) | |
| Brief Description: | Process of creating invitation campaigns for the event and sending them through a social media platform. | |
| Pre-Condition: | The event organizer must be logged in to their account and have access to the desired social media platform. | |
| Post-Condition: | The invitations are created and sent to the intended audience through the chosen social media platform. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer creates an invitation campaign, including content for lead generation. | The system utilizes the external social media platform to send the invitations to the specified audience as per the campaign settings. |

Table 22: Use Case21=Invitation Campaign

| | | |
|-------------------------------|---|--|
| 1.2.11 Survey Questions | | |
| Use Case Name: | <u>Survey Questions</u> | |
| ID: | UC211 | |
| Actors Involved: | Event Organizer, External System (Email Platform) | |
| Brief Description: | Process of creating survey questions and sending them through an email platform to collect feedback from attendees. | |
| Pre-Condition: | The event organizer must be logged in to their account and have access to the desired email platform. | |
| Post-Condition: | The survey questions are created and sent to the attendees via email. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer creates survey questions, including the question content and format. | The system utilizes the external email platform to send the survey questions to the attendees' email addresses, allowing them to provide feedback. |

Table 23: Use Case22=Survey Questions

| | | |
|-------------------------------|--|--|
| 1.3.1 Process Payment | | |
| Use Case Name: | <u>Process Payment</u> | |
| ID: | UC301 | |
| Actors Involved: | Payment Gateway | |
| Brief Description: | Process of Processing payments through PayPal or a bank. | |
| Pre-Condition: | The attendee must have provided valid payment details. | |
| Post-Condition: | The payment is Processed and confirmed. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The attendee provides payment details and selects the preferred payment method (PayPal or bank). | The system securely communicates the payment details to the payment gateway, which Processes the payment using the selected method and confirms the transaction. |

Table 24: Use Case23= Process Payment

| | | |
|-------------------------------|--|---|
| 1.3.2 Payment Refund | | |
| Use Case Name: | <u>Payment Refund</u> | |
| ID: | UC302 | |
| Actors Involved: | Payment Gateway | |
| Brief Description: | Process of refunding a payment. | |
| Pre-Condition: | The refund request must be initiated by the event organizer or attendee. | |
| Post-Condition: | The payment is refunded to the original payment source. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event organizer or attendee requests a refund. | The system communicates the refund request to the payment gateway, which Processes the refund and returns the funds to the original payment source. |

Table 25: Use Case24= Payment Refund

| | | |
|-------------------------------|---|--|
| 1.4.1 Generate QR Code | | |
| Use Case Name: | <u>Generate QR Code</u> | |
| ID: | UC401 | |
| Actors Involved: | External System (QR Code Generator) | |
| Brief Description: | Process of generating a QR code. | |
| Pre-Condition: | The relevant data, such as ticket details, must be available. | |
| Post-Condition: | A QR code is generated for the specified data. | |
| Normal Flow of Events: | Actor Action | System Response |
| | None. | The system sends the necessary data to the external QR code generator, which generates a unique QR code. |

Table 26: Use Case25= Generate QR Code

| | | |
|-----------------------------------|---|---|
| 1.4.2 Paste the QR Code in Ticket | | |
| Use Case Name: | <u>Paste the QR Code in Ticket</u> | |
| ID: | UC402 | |
| Actors Involved: | External System (QR Code Generator) | |
| Brief Description: | Process of integrating the generated QR code into the ticket. | |
| Pre-Condition: | The QR code must be generated. | |
| Post-Condition: | The ticket contains the embedded QR code. | |
| Normal Flow of Events: | Actor Action | System Response |
| | None. | The system inserts the generated QR code into the ticket design or template, ensuring it is visible and scanable. |

Table 27: Use Case26= Paste the QR Code in Ticket

| | | |
|--|---|---|
| 1.4.3 Check-in with QR Code for Attendance | | |
| Use Case Name: | <u>Check-in with QR Code for Attendance</u> | |
| ID: | UC403 | |
| Actors Involved: | External System (QR Code Generator) | |
| Brief Description: | Process of checking in an attendee using their QR code for attendance tracking. | |
| Pre-Condition: | The event must be ongoing, and the attendee must have their ticket with the embedded QR code. | |
| Post-Condition: | The attendee's attendance is recorded. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The event staff or system operator scans the attendee's QR code. | The external QR code generator processes the scanned QR code, verifies its authenticity, and marks the attendee's attendance. |

Table 28: Use Case27= Check-in with QR Code

| | | |
|-------------------------------|---|--|
| 1.5.1 Send Tickets | | |
| Use Case Name: | <u>Send Tickets</u> | |
| ID: | UC501 | |
| Actors Involved: | External System (Email Platform) | |
| Brief Description: | Process of sending tickets to attendees via email. | |
| Pre-Condition: | The attendee's email address and valid tickets must be available. | |
| Post-Condition: | The tickets are sent to the attendees' email addresses. | |
| Normal Flow of Events: | Actor Action | System Response |
| | None. | The system utilizes the external email platform to send the tickets to the attendees' email addresses, ensuring secure delivery and accessibility. |

Table 29: Use Case28= Send Tickets

| | | |
|-------------------------------|--|--|
| 1.5.2 Send Survey Questions | | |
| Use Case Name: | <u>Send Survey Questions</u> | |
| ID: | UC502 | |
| Actors Involved: | External System (Email Platform) | |
| Brief Description: | Process of sending survey questions to attendees via email. | |
| Pre-Condition: | The attendee's email address and valid survey questions must be available. | |
| Post-Condition: | The survey questions are sent to the attendees' email addresses. | |
| Normal Flow of Events: | Actor Action | System Response |
| | None. | The system utilizes the external email platform to send the survey questions to the attendees' email addresses, allowing them to provide feedback and responses. |

Table 30: Use Case29= Send Survey Questions

| | | |
|--------------------------------|---|---|
| 1.6.1 Send Invitation Campaign | | |
| Use Case Name: | <u>Send Invitation Campaign</u> | |
| ID: | UC601 | |
| Actors Involved: | External System (Social Media Platform) | |
| Brief Description: | Process of sending invitation campaigns through a social media platform. | |
| Pre-Condition: | The event organizer must have created the invitation campaign and have access to the desired social media platform. | |
| Post-Condition: | The invitation campaign is sent to the intended audience through the chosen social media platform. | |
| Normal Flow of Events: | Actor Action | System Response |
| | None. | The system utilizes the external social media platform to send the invitation campaign, ensuring wide reach and engagement among the target audience. |

Table 31: Use Case30= Send Invitation Campaign

Actor=Admin Use Cases

| 1.7.1 Login | | |
|-------------------------------|--|---|
| Use Case Name: | <u>Login</u> | |
| ID: | UC701 | |
| Actors Involved: | System Admin | |
| Brief Description: | Process of logging in to the system as a system admin using a password. | |
| Pre-Condition: | The system admin account must exist. | |
| Post-Condition: | The system admin gains access to the administrative functionalities of the system. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The system admin provides their username and password. | The system verifies the provided credentials and grants access to the system admin account. |

Table 32: Use Case31= Login

| 1.7.2 Manage System | | |
|-------------------------------|--|---|
| Use Case Name: | <u>Manage System</u> | |
| ID: | UC702 | |
| Actors Involved: | System Admin | |
| Brief Description: | Process of managing the overall system and its functionalities. | |
| Pre-Condition: | The system admin must be logged in to their account. | |
| Post-Condition: | The system settings and configurations are modified as per the system admin's actions. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The system admin performs various administrative tasks, such as modifying system settings, managing user accounts, or monitoring system performance. | The system implements the changes and updates based on the system admin's actions, ensuring the smooth operation and maintenance of the overall system. |

Table 33: Use Case32= Manage System

| 1.7.3 Manage Events (Approve/Decline) | | |
|---------------------------------------|---|---|
| Use Case Name: | <u>Manage Events (Approve/Decline)</u> | |
| ID: | UC703 | |
| Actors Involved: | System Admin, Event Organizer | |
| Brief Description: | Process of managing events by approving or declining event submissions. | |
| Pre-Condition: | The system admin must be logged in to their account, and event submissions must exist. | |
| Post-Condition: | The event submissions are either approved or declined based on the system admin's decision. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The system admin reviews the event submissions and selects the appropriate action (approve or decline). | The system updates the status of the event submissions accordingly, notifying the event organizers of the decision. |

Table 34: Use Case33= Manage Events

| | | |
|---|--|--|
| 1.8.1 Assign Seats for Individual & Group | | |
| Use Case Name: | Assign Seats for Individual & Group | |
| ID: | UC801 | |
| Actors Involved: | System | |
| Brief Description: | Process of assigning seats for individual and group ticket purchases inthe online ticketing event management system. | |
| Pre-Condition: | Tickets must be purchased. Available seats must exist. | |
| Post-Condition: | Seats are assigned to the purchased tickets, considering the preferences and availability of seats. | |
| Normal Flow of Events: | Actor Action | System Response |
| | The system analyzes the purchased tickets and assigns seats based on the preferencesand availability of seats. | The system checks the purchasedtickets and determines suitable seats for individual and group bookings. It takes into account the seating preferences providedby the attendees and ensures that the seats are allocated efficiently and according to the event's seating arrangements. |

Table 35: Use Case34= Assign Seats

4.5. Summary:

In this chapter, we met with the State Diagrams, Enhanced Entity relationship Diagrams and UML Diagram. Our Web application covered the all-major modules, which were used to fulfill the requirements so the State Diagram and Enhanced entity relationship Diagram (EERD's) elaboratethis in detailed because when we gathered the information so it show the flow of our system specifically like in implementation we make sure that the development is start or not then afterthis testing process will be occurred so if there is any error occur so it must be solved at that time then we monitor our Web application by time to time to make sure that everything is completeor not that why we make State diagram and Enhanced entity relationship diagram to focus on our mistake. In spite of everything, information has been accumulated and layout has been createdso now, the implementation has been initiated according to the Web applications' requirement to make sure that it is beneficial for the user on a long run

CHAPTER – 5

5.0. Introduction:

In this chapter we are discussing aspects which are used in our project, and prototype design which is generally used to evaluate a new design to enhance precision by system analysts and users and frontend and backend design of our project. We are also discussing about the database queries which are used in firebase and some external libraries and we are showing screenshots of our application as it fulfills user requirements. We have briefly provided a few clarifications about the sort of functionalities available on the system, source code of validation and etc.

5.1. Prototype Design:

Project Title: Event Flow Web Application

Date: 18/06/2023

Screen Name: Register Screen

Screen: < 1 of 6 >

Link from screen: Login Window

Link to screen: Login Window

Screen Description:

Contain Fields such as: Username, password, first name, last name, Email Address, Phone no., Registration type

Functionality/Interactivity:

User Register himself by input the required credentials mention in screen

Screen Design:

The screenshot displays the Event Flow web application interface. At the top, there is a dark header bar with contact information (+92 336273926, support@eventflow.com), a 'SIGN IN | SIGN UP' button, and a user icon showing '0 ..'. Below the header is a navigation bar with links for EVENT, CALENDAR VIEW, SEARCH, CREATE AN EVENT, and CONTACT. The main content area has a title 'Register'. It contains several input fields for registration: Username, Password, First Name, Last Name, CNIC (Government Issued ID Card), Mailing Address, E-mail Address, and Phone no. There is also a 'Registration Type' section with two checkboxes: 'Register as a Attendee' and 'Register as Organizer'. At the bottom of the form are two buttons: 'REGISTER' (orange) and 'LOGIN' (grey). Below the register form is another screenshot of the Event Flow dashboard, which includes social media links (Facebook, Twitter, LinkedIn, Google, Instagram) and a 'DASHBOARD' section with 'Contact us' and 'Subscriber Login' links.

Background: white

Audio: none

Color scheme: white, Dark blue, black, lilac, yellow, grey, green, and Orange, red, light blue

Video: none

Text attributes: Aclonica(24sp) Arial(20dp)

Still images: Registration page

Table 36: prototype1= Register page

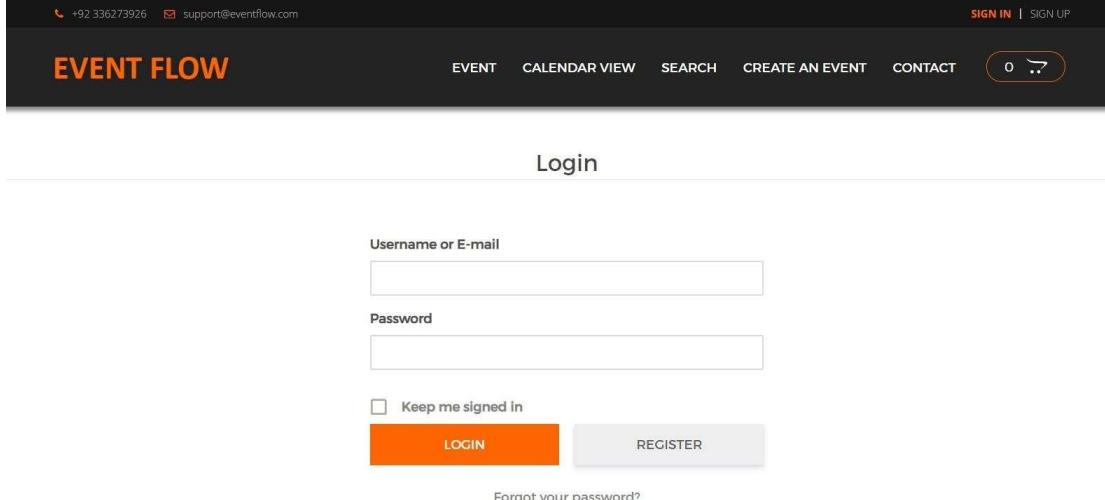
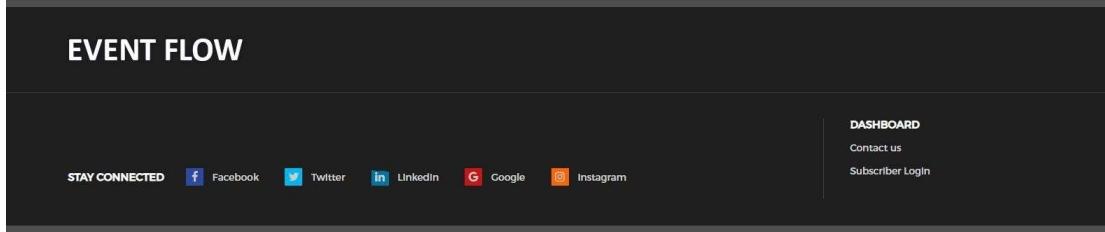
| | |
|--|---|
| <p>Project Title: Event Flow Web Application Date: 18/06/2023 Screen Name: login Panel</p> | |
| <p>Screen: < 2 of 6 > Link from screen: Main Window Link to screen: Event Flow Web Application window and Register window</p> | <p>Screen Description: Email textbox, Password textbox, loginbutton, Text, Image.</p> |
| <p>Functionality/Interactivity: Use inputs data to use the app. If he/she is a new user, user chooses the register button.</p> | |
| <p>Screen Design:</p>  | |
|  | |
| <p>Background: textbox, Password textbox, login button, Text, Image.</p> | <p>Audio: none</p> |
| <p>Color scheme: white, Dark blue, black, lilac, yellow, grey, green, and Orange, red, light blue</p> | <p>Video: none</p> |
| <p>Text attributes: Aclonica(24sp) Arial(12dp) (16dp) (8dp)</p> | <p>Still images: loginback.jpg</p> |

Table 37: prototype2= Login page

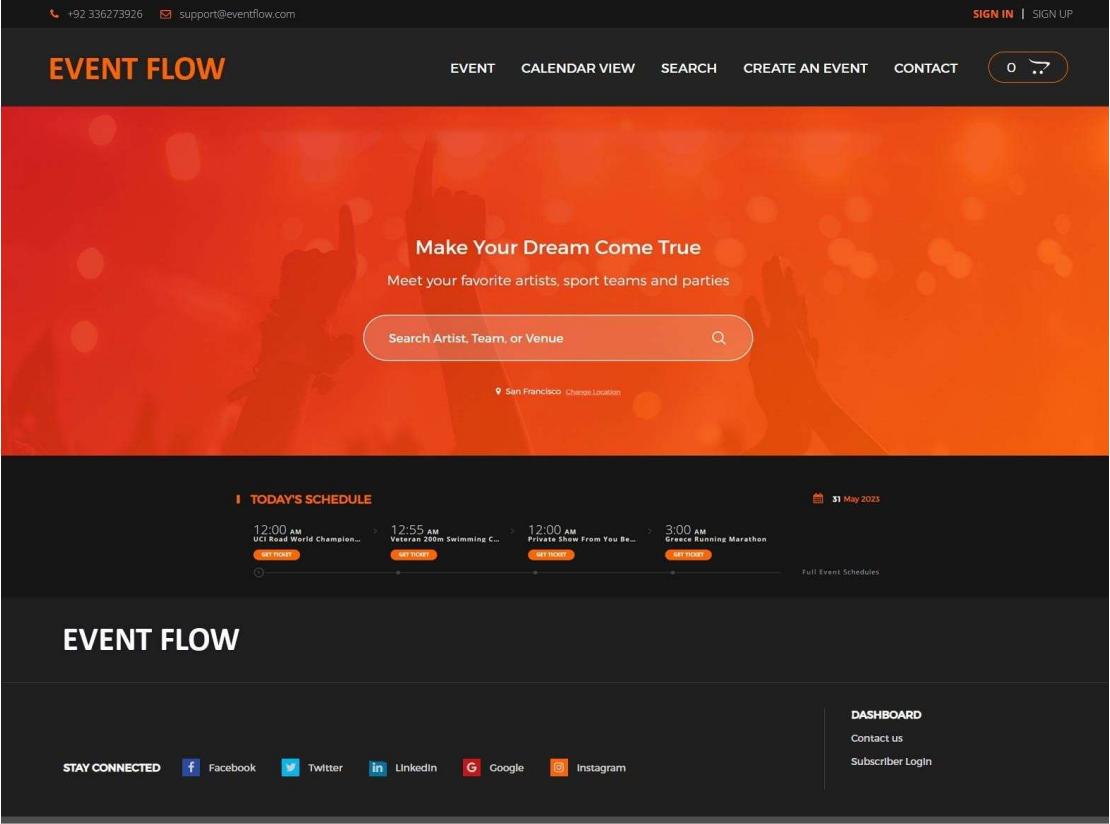
| | |
|--|--|
| <p>Project Title: Event Flow Web Application Date: 18/06/2023 Screen Name: Application Main page or Event Search Page</p> | |
| <p>Screen: < 3 of 6 > Link from screen: Login Window Link to screen: Event, Calendar View, Account Search , Create a new event Windows</p> | <p>Screen Description: Event, Calendar View, Search, Create an Event, Contact, Account, Search bar</p> |
| <p>Functionality/Interactivity: Acts as a dashboard where user can choose Event List, Calendar View See contact info and go to account page. If user is organizer he can also choose create an event</p> | |
| <p>Screen Design:</p>  | |
| <p>Background: white</p> | <p>Audio: none</p> |
| <p>Color scheme: white, dark blue, black, red, green, pink, purple, black, yellow, orange, red and brown</p> | <p>Video: Sideshow at Banner</p> |
| <p>Text attributes: Aclonica(24sp)Arial(12dp) (16dp) (8dp)</p> | <p>Still images: cardfour.jpg, cardthree.jpg, cardone.jpg</p> |

Table 38: prototype3= Main Window

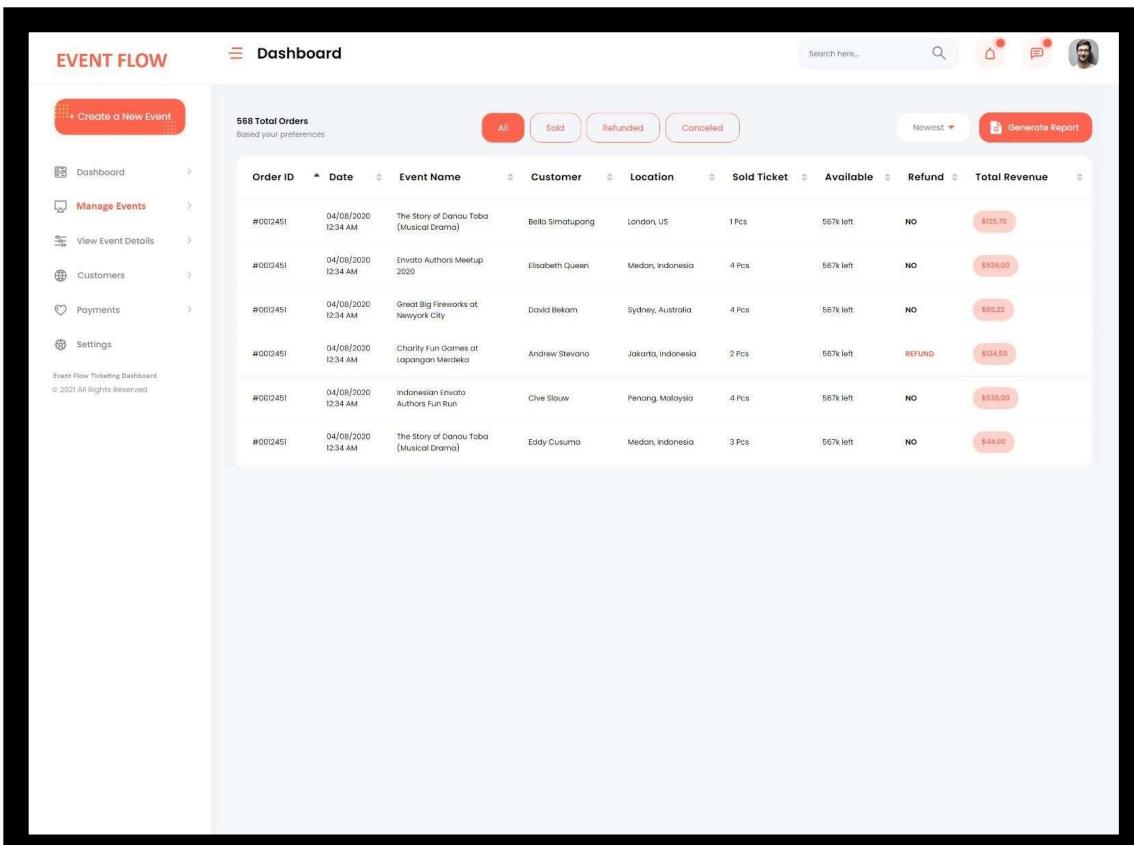
| <p>Project Title: Event Flow Web Application</p> <p>Date: 18/06/2023</p> <p>Screen Name: Organizer dashboard for manage events</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|---|------------------|--------------------|-------------|-------------|-----------|---------------|---------------|----------|------------------------|---|------------------|------------|-------|-----------|----|----------|----------|------------------------|----------------------------|-----------------|------------------|-------|-----------|----|----------|----------|------------------------|--------------------------------------|-------------|-------------------|-------|-----------|----|---------|----------|------------------------|---------------------------------------|----------------|--------------------|-------|-----------|--------|----------|----------|------------------------|-----------------------------------|------------|------------------|-------|-----------|----|----------|----------|------------------------|---|-------------|------------------|-------|-----------|----|---------|
| <p>Screen: < 4 of 6 ></p> <p>Link from screen: Dashboard Window</p> <p>Link to screen: Dashboard, View Event details, Contact, Account, Search bar</p> <p>Customers, Payments, Settings Windows</p> | <p>Screen Description:</p> <p>Event, Calendar View, Search, Create an Event, Contact, Account, Search bar</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Functionality/Interactivity:</p> <p>Acts as a dashboard where Organizer can manage Events, view Event details, view Attendees list, View Payments, view sold tickets</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Screen Design:</p>  <table border="1"> <thead> <tr> <th>Order ID</th> <th>Date</th> <th>Event Name</th> <th>Customer</th> <th>Location</th> <th>Sold Ticket</th> <th>Available</th> <th>Refund</th> <th>Total Revenue</th> </tr> </thead> <tbody> <tr> <td>#0012451</td> <td>04/08/2020 12:34 AM</td> <td>The Story of Danau Toba (Musical Drama)</td> <td>Bella Simatupang</td> <td>London, US</td> <td>1 Pcs</td> <td>567k left</td> <td>NO</td> <td>\$125.70</td> </tr> <tr> <td>#0012451</td> <td>04/08/2020 12:34 AM</td> <td>Envato Authors Meetup 2020</td> <td>Elisabeth Queen</td> <td>Medan, Indonesia</td> <td>4 Pcs</td> <td>567k left</td> <td>NO</td> <td>\$530.00</td> </tr> <tr> <td>#0012451</td> <td>04/08/2020 12:34 AM</td> <td>Great Big Fireworks at New York City</td> <td>David Tekam</td> <td>Sydney, Australia</td> <td>4 Pcs</td> <td>567k left</td> <td>NO</td> <td>\$89.23</td> </tr> <tr> <td>#0012451</td> <td>04/08/2020 12:34 AM</td> <td>Charity Fun Games at Lapangan Merdeka</td> <td>Andrew Stevano</td> <td>Jakarta, Indonesia</td> <td>2 Pcs</td> <td>567k left</td> <td>REFUND</td> <td>\$104.55</td> </tr> <tr> <td>#0012451</td> <td>04/08/2020 12:34 AM</td> <td>Indonesian Envato Authors Fun Run</td> <td>Cive Staww</td> <td>Penang, Malaysia</td> <td>4 Pcs</td> <td>567k left</td> <td>NO</td> <td>\$530.00</td> </tr> <tr> <td>#0012451</td> <td>04/08/2020 12:34 AM</td> <td>The Story of Danau Toba (Musical Drama)</td> <td>Eddy Cusumo</td> <td>Medan, Indonesia</td> <td>3 Pcs</td> <td>567k left</td> <td>NO</td> <td>\$44.00</td> </tr> </tbody> </table> | Order ID | Date | Event Name | Customer | Location | Sold Ticket | Available | Refund | Total Revenue | #0012451 | 04/08/2020 12:34 AM | The Story of Danau Toba (Musical Drama) | Bella Simatupang | London, US | 1 Pcs | 567k left | NO | \$125.70 | #0012451 | 04/08/2020 12:34 AM | Envato Authors Meetup 2020 | Elisabeth Queen | Medan, Indonesia | 4 Pcs | 567k left | NO | \$530.00 | #0012451 | 04/08/2020 12:34 AM | Great Big Fireworks at New York City | David Tekam | Sydney, Australia | 4 Pcs | 567k left | NO | \$89.23 | #0012451 | 04/08/2020 12:34 AM | Charity Fun Games at Lapangan Merdeka | Andrew Stevano | Jakarta, Indonesia | 2 Pcs | 567k left | REFUND | \$104.55 | #0012451 | 04/08/2020 12:34 AM | Indonesian Envato Authors Fun Run | Cive Staww | Penang, Malaysia | 4 Pcs | 567k left | NO | \$530.00 | #0012451 | 04/08/2020 12:34 AM | The Story of Danau Toba (Musical Drama) | Eddy Cusumo | Medan, Indonesia | 3 Pcs | 567k left | NO | \$44.00 |
| Order ID | Date | Event Name | Customer | Location | Sold Ticket | Available | Refund | Total Revenue | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| #0012451 | 04/08/2020 12:34 AM | The Story of Danau Toba (Musical Drama) | Bella Simatupang | London, US | 1 Pcs | 567k left | NO | \$125.70 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| #0012451 | 04/08/2020 12:34 AM | Envato Authors Meetup 2020 | Elisabeth Queen | Medan, Indonesia | 4 Pcs | 567k left | NO | \$530.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| #0012451 | 04/08/2020 12:34 AM | Great Big Fireworks at New York City | David Tekam | Sydney, Australia | 4 Pcs | 567k left | NO | \$89.23 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| #0012451 | 04/08/2020 12:34 AM | Charity Fun Games at Lapangan Merdeka | Andrew Stevano | Jakarta, Indonesia | 2 Pcs | 567k left | REFUND | \$104.55 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| #0012451 | 04/08/2020 12:34 AM | Indonesian Envato Authors Fun Run | Cive Staww | Penang, Malaysia | 4 Pcs | 567k left | NO | \$530.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| #0012451 | 04/08/2020 12:34 AM | The Story of Danau Toba (Musical Drama) | Eddy Cusumo | Medan, Indonesia | 3 Pcs | 567k left | NO | \$44.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Background: white</p> <p>Color scheme: white, dark blue, black, black, orange and brown</p> <p>Text attributes: Aclonica(24sp) Arial(12dp) (16dp) (8dp)</p> | <p>Audio: none</p> <p>Video: none</p> <p>Still images: Logo Jpg</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 39: prototype4= Manage Event Window

Project Title: Event Flow Web Application

Date: 18/06/2023

Screen Name: Organizer dashboard for View sold ticket and customer details

Screen: < 5 of 6 >

Link from screen: Dashboard Window

Link to screen: Dashboard, Manage Events,

View Event details, Payments, Settings Windows

Screen Description:

Event, Calendar View, Search, Create an Event, Contact, Account, Search bar

Functionality/Interactivity:

Acts as a dashboard where Organizer can manage Events, view Event details, view Attendees list, View Payments, view sold tickets

Screen Design:

| Cust. ID | Date Join | Customer Name | Ticket Ordered | Location | Last Order | Total Spent |
|----------|------------|------------------|--|------------------------|---------------------|-------------|
| #0012451 | 21/11/2017 | Bella Simatupang | The Powerfull Concert Festival London 2020 | Sydney, Australia | 04/08/2020 12:34 AM | \$623.55 |
| #0012451 | 21/11/2017 | Olivia Brownlee | The Powerfull Concert Festival London 2020 | Medan, Indonesia | 04/08/2020 12:34 AM | \$240.45 |
| #0012451 | 21/11/2017 | Clive Stauw | The Story of Danau Toba (Musical drama) | London, United Kingdom | 04/08/2020 12:34 AM | \$1300 |
| #0012451 | 21/11/2017 | Andrew Stevano | The Story of Danau Toba (Musical drama) | Sydney, Australia | 04/08/2020 12:34 AM | \$623.55 |
| #0012451 | 21/11/2017 | James Roberto | The Powerfull Concert Festival London 2020 | Sydney, Australia | 04/08/2020 12:34 AM | \$1300 |
| #0012451 | 21/11/2017 | Kevin Hurt | The Powerfull Concert Festival London 2020 | Medan, Indonesia | 04/08/2020 12:34 AM | \$623.65 |
| #0012451 | 21/11/2017 | Kevin Hurt | The Powerfull Concert Festival London 2020 | Medan, Indonesia | 04/08/2020 12:34 AM | \$623.55 |
| #0012451 | 21/11/2017 | Sonuel Jakson | The Story of Danau Toba (Musical drama) | Jakarta, Indonesia | 04/08/2020 12:34 AM | \$56.025 |
| #0012451 | 21/11/2017 | Bella Simatupang | The Powerfull Concert Festival London 2020 | Sydney, Australia | 04/08/2020 12:34 AM | \$623.55 |
| #0012451 | 21/11/2017 | Bella Simatupang | The Powerfull Concert Festival London 2020 | Sydney, Australia | 04/08/2020 12:34 AM | \$623.55 |

Background: white

Audio: none

Color scheme: white, dark blue, black, black, orange and brown

Video: none

Text attributes: Aclonica(24sp) Arial(12dp)
(16dp) (8dp)

Still images: Logo Jpg

Table 40: prototype5= Customer Window

| | |
|---|-------------------------------|
| <p>Project Title: Event Flow Web Application</p> <p>Date: 18/06/2023</p> <p>Screen Name: Organizer dashboard for View Analytics Report</p> | |
| Screen: < 5 of 6 > | Screen Description: |
| <p>Link from screen: Manage event Window</p> <p>Link to screen: Dashboard, Manage Events, View Event details, Payments, Settings Windows</p> | |
| <p>Functionality/Interactivity: Acts as a dashboard where Organizer can manage Events, view Event details, view Attendees list, View Payments, view sold tickets</p> | |
| <p>Screen Design:</p> | |
| Background: white | Audio: none |
| Color scheme: white, dark blue, black, black, orange, green and brown | Video: none |
| Text attributes: Aclonica(24sp) Arial(12dp) (16dp) (8dp) | Still images: Logo Jpg |

Table 41: prototype6= Analytics Report Window

5.2 .Database Queries:

Figure6: Signup and Login page:

```
admin > includes > db.php > ...
1  <?php
2  session_start();
3  include("config.php");
4
5 //For sign up page
6 if (isset($_POST["signup"])) {
7     // Fetch data from the form
8     $username = $_POST['username'];
9     $email = $_POST['email'];
10    $password = $_POST['password'];
11    $role = $_POST['role'];
12    $targetDirectory = "../assets/images/avatar/";
13    $avatar = $_FILES["avatar"]["name"];
14    $targetFile = $targetDirectory . basename($_FILES["avatar"]["name"]);
15    $avatarFileType = strtolower(pathinfo($targetFile, PATHINFO_EXTENSION));
16    $allowedTypes = array('jpg', 'jpeg', 'png', 'gif');
17    if (in_array($avatarFileType, $allowedTypes)) {
18        if (move_uploaded_file($_FILES["avatar"]["tmp_name"], $targetFile)) {
19            echo "The file " . htmlspecialchars(basename($_FILES["avatar"]["name"])) . " has been uploaded.";
20        } else {
21            echo "Sorry, there was an error uploading your file.";
22            exit();
23        }
24    } else {
25        echo "Invalid file type. Please upload only JPG, JPEG, PNG, or GIF files.";
26    }
27
28    $sql = mysqli_query($con, "INSERT INTO `users` (`username`, `email`, `password`, `role`, `avatar`) VALUES
29    ('$username', '$email', '$password', '$role', '$avatar')");
30    if ($sql) {
31        header("location:../login.php");
32        exit;
33    }
34 }
35
36 //login
37 if (isset($_POST['login'])) {
38     $email = $_POST['email'];
39     $password = $_POST['password'];
40     $query = "SELECT * FROM `users` WHERE `email` = '$email' AND `password` = '$password'";
41     $result = mysqli_query($con, $query);
42     if ($result) {
43         if (mysqli_num_rows($result) == 1) {
44             foreach ($result as $user) {
45                 $role = $user['role'];
46                 $_SESSION['uid'] = $user['id'];
47                 $_SESSION[$role] = $user;
48                 $_SESSION['amount'] = $user['id'];
49                 $_SESSION['email'] = $user['email'];
50                 $_SESSION['username'] = $user['name'];
51                 header("location: ../../index.php");
52                 exit();
53             } else {
54                 echo "Invalid credentials. Please check your email and password.";
55             }
56         } else {
57             echo "Error in query execution: " . mysqli_error($con);
58         }
59     }
60 }
```

Figure7: Create, Edit and Delete organizer:

```
admin > includes > db.php > ...
1 //Create organizer
2 if (isset($_POST['create_organizer'])) {
3     $name = $_POST['username'];
4     $email = $_POST['email'];
5     $password = $_POST['password'];
6     $targetDirectory = "../images/avatar/";
7     $image = $_FILES["image"]["name"];
8     $targetFile = $targetDirectory . basename($_FILES["image"]["name"]);
9     $imageFileType = strtolower(pathinfo($targetFile, PATHINFO_EXTENSION));
10    $allowedTypes = array('jpg', 'jpeg', 'png', 'gif');
11    if (in_array($imageFileType, $allowedTypes)) {
12        if (move_uploaded_file($_FILES["image"]["tmp_name"], $targetFile)) {
13            echo "The file " . htmlspecialchars(basename($_FILES["image"]["name"])) . " has been uploaded.";
14        } else {
15            echo "Sorry, there was an error uploading your file.";
16        }
17    } else {
18        echo "Invalid file type. Please upload only JPG, JPEG, PNG, or GIF files.";
19    }
20    $insert = mysqli_query($con, "INSERT INTO `users`(`username`, `email`, `password`, `avatar`, `role`) VALUES ('$name', '$email', '$password', '$image', 'Organizer')");
21    if ($insert) {
22        header("location:../users.php");
23        exit();
24    } else {
25        echo "User not inserted";
26    }
27}
28 //delete organizer
29 if (isset($_POST["deleteOrganizer"])) {
30     $id = $_POST["id"];
31     $delete_user = mysqli_query($con, "UPDATE `users` SET `is_deleted` = 1 WHERE `id` = '$id'");
32     if ($delete_user) {
33         header("location: ../users.php");
34         exit();
35     }
36 }
37
38 admin > includes > db.php > ...
39
40 //edit organizer
41 if (isset($_POST["editOrganizer"])) {
42     $id = $_POST["id"];
43     $username = $_POST["organizerName"];
44     $email = $_POST["email"];
45     $password = $_POST["password"];
46     $address = $_POST["address"];
47     $city = $_POST["city"];
48     $state = $_POST["state"];
49     $zip_code = $_POST["zip_code"];
50     $location = $_POST["location"];
51     $phone = $_POST["phone"];
52     $edit_user = mysqli_query($con, "UPDATE `users` SET `username`='$username',
53     `email`='$email', `password`='$password', `address`='$address', `city`='$city',
54     `state`='$state', `zip_code`='$zip_code', `location`='$location', `phone`='$phone'
55     |WHERE `id` = '$id' ");
56     if ($edit_user) {
57         header("location: ../users.php");
58         exit();
59     } else {
60         echo "Error: " . mysqli_error($con); // Display error, if any
61     }
62 }
```

Figure8: User profile page:

```
admin > includes > db.php > ...
125
126 //edit userprofile
127 if (isset($_POST["editProfile"])) {
128     $id = $_POST["id"];
129     $username = $_POST["name"];
130     $email = $_POST["email"];
131     $password = $_POST["password"];
132     $address = $_POST["address"];
133     $city = $_POST["city"];
134     $state = $_POST["state"];
135     $zip_code = $_POST["zip_code"];
136     $location = $_POST["location"];
137     $phone = $_POST["phone"];
138     $old_avatar = $_POST["old_avatar"];
139     if (!empty($_FILES['new_avatar']['name'])) {
140         $avatar = $_FILES['new_avatar']['name'];
141         $image = '../assets/images/avatar/' . basename($_FILES['new_avatar']['name']);
142         move_uploaded_file($_FILES['new_avatar']['tmp_name'], $image);
143     } else {
144         // Use the old image name if no new image is provided
145         $avatar = $old_avatar;
146     }
147     $edit_user = mysqli_query($con, "UPDATE `users` SET `username`='$username', `email`='$email',
148                             `password`='$password', `address`='$address', `city`='$city', `state`='$state',
149                             `zip_code`='$zip_code', `location`='$location', `phone`='$phone' , `avatar`='$avatar'
150                             WHERE `id`='$id' ");
151
152     if ($edit_user) {
153         header("location: ../profile.php");
154         exit;
155     } else {
156         echo "Error: " . mysqli_error($con); // Display error, if any
157     }
158 }
```

Figure9: Creating new events:

```

admin > includes > db.php > ...
162 //Create new Event
163 if (isset($_POST['create_new'])) {
164     $parent_id = $_POST['parent_id'];
165     $org_id = $_SESSION['uid'];
166     $name = $_POST['name'];
167     $description = $_POST['description'];
168     $status = $_POST['status'];
169     $start_date = $_POST['start_date'];
170     $end_date = $_POST['end_date'];
171     $start_time = $_POST['start_time'];
172     $end_time = $_POST['end_time'];
173     $location = $_POST['location'];
174     $capacity = $_POST['capacity'];
175     $category_id = $_POST['category_id'];
176     $targetDirectory = "../../assets/images/events/";
177     $image = $_FILES["image"]["name"];
178     $targetFile = $targetDirectory . basename($_FILES["image"]["name"]);
179     $imageFileType = strtolower(pathinfo($targetFile, PATHINFO_EXTENSION));
180     $allowedTypes = array('jpg', 'jpeg', 'png', 'gif');
181     if (in_array($imageFileType, $allowedTypes)) {
182         if (move_uploaded_file($_FILES["image"]["tmp_name"], $targetFile)) {
183             echo "The file " . htmlspecialchars(basename($_FILES["image"]["name"])) . " has been uploaded.";
184         } else {
185             echo "Sorry, there was an error uploading your file.";
186             exit();
187         }
188     } else {
189         echo "Invalid file type. Please upload only JPG, JPEG, PNG, or GIF files.";
190     }
191     if ($_POST['event_type'] == "main") {
192         $insert = mysqli_query($con, "INSERT INTO `events`(`event_id`, `category_id`, `name`, `start_date`, `end_date`, `image`, `description`, `organizer_id`) VALUES ('$id', '$category_id', '$name', '$start_date', '$end_date', '$image', '$description', '$org_id')");
193         if ($insert) {
194             header("location:../main_events.php");
195             exit();
196         }
197     }
}

admin > includes > db.php > ...
198     } else {
199         echo "User not inserted";
200     }
201 } else {
202     $q = mysqli_query($con, "(select category_id from events where event_id = $parent_id)");
203     $req = mysqli_fetch_assoc($q);
204     $cat = $req['category_id'];
205     $insert = mysqli_query($con, "INSERT INTO `events`(`parent_id`, `category_id`, `event_id`, `name`, `description`, `status`, `start_date`, `end_date`, `start_time`, `end_time`, `location`, `capacity`, `image`, `organizer_id`) VALUES ('$parent_id', '$cat', '$id', '$name', '$description', '$status', '$start_date', '$end_date', '$start_time', '$end_time', '$location', '$capacity', '$image', '$org_id')");
206     if ($insert) {
207         header("location:../events.php");
208         exit();
209     } else {
210         echo "User not inserted";
211     }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
```

Figure10: Editing and Deleting Main Events:

```
admin > includes > db.php > ...
269 //Edit Main event
270 if (isset($_POST["editMainEvent"])) {
271     $id = $_POST['event_id'];
272     $name = $_POST['eventname'];
273     $description = $_POST['description'];
274     $start_date = $_POST['start_date'];
275     $end_date = $_POST['end_date'];
276     $oldImage = $_POST['old_image'];
277     $newImage = $_FILES['new_image']['name'];
278     if (!empty($newImage)) {
279         $targetDirectory = "../assets/images/events/";
280         $targetFile = $targetDirectory . basename($_FILES["new_image"]["name"]);
281         $image = $_FILES['new_image']['name'];
282         if (move_uploaded_file($_FILES["new_image"]["tmp_name"], $targetFile)) {
283         }
284     } else {
285         $image = $oldImage;
286     }
287     $edit_main_event = mysqli_query($con, "UPDATE `events` SET `event_id`='$id',
288     `name`='$name', `image` = '$image' , `start_date` = '$start_date',
289     `end_date` = '$end_date' , `description` = '$description' WHERE `event_id` = '$id' ");
290
291     if ($edit_main_event) {
292         header("location: ../main_events.php");
293         exit;
294     } else {
295         echo "Error: " . mysqli_error($con); // Display error, if any
296     }
297 }
298

admin > includes > db.php > ...
299 //Delete Main event
300 if (isset($_POST["deleteMainEvent"])) {
301     $id = $_POST["id"];
302     $delete_event = mysqli_query($con, "UPDATE `events` SET `is_deleted` = 1 WHERE `event_id` = '$id'");
303     if ($delete_event) {
304         $delete_event = mysqli_query($con, "UPDATE `events` SET `is_deleted` = 1 WHERE `parent_id` = '$id'");
305         header("location: ../main_events.php");
306         exit;
307     }
308 }
```

Figure11: Editing and Deleting Sub Events:

```
admin > includes > db.php > ...
222 //Edit Sub event
223 if (isset($_POST["editEvent"])) {
224     $id = $_POST['event_id'];
225     $name = $_POST['eventname'];
226     $description = $_POST['description'];
227     $start_date = $_POST['start_date'];
228     $end_date = $_POST['end_date'];
229     $start_time = $_POST['start_time'];
230     $end_time = $_POST['end_time'];
231     $seats = $_POST['seats'];
232     $location = $_POST['location'];
233     $capacity = $_POST['capacity'];
234     $oldImage = $_POST['old_image'];
235     $newImage = $_FILES['new_image'][name];
236     if (!empty($newImage)) [
237         $targetDirectory = "../../assets/images/events/";//Change the directory as per your configuration
238         $targetFile = $targetDirectory . basename($_FILES["new_image"]["name"]);
239         $image = $_FILES['new_image'][name];
240         if (move_uploaded_file($_FILES["new_image"]["tmp_name"], $targetFile)) {
241             }
242         } else {
243             $image = $oldImage;
244         }
245         $edit_event = mysqli_query($con, "UPDATE `events` SET `event_id`='$id',
246         `name`='$name',`description`='$description',`start_date`='$start_date',
247         `end_date`='$end_date',`start_time`='$start_time',`end_time`='$end_time',
248         `seats`='$seats',`location`='$location',`capacity`='$capacity',`image`='$image'
249         WHERE `event_id` = '$id' ");
250         if ($edit_event) {
251             header("location: ../../events.php");
252             exit;
253         } else {
254             echo "Error: " . mysqli_error($con); // Display error, if any
255         }
256     }
257     //Delete Sub event
258     if (isset($_POST["deleteEvent"])) {
259         $id = $_POST["id"];
260         $delete_event = mysqli_query($con, "UPDATE `events` SET `is_deleted`= 1 WHERE `event_id` = '$id'");
261         if ($delete_event) {
262             header("location: ../../events.php");
263             exit;
264         }
265     }
}
```

Figure12: Creating and deleting blogs:

```
admin > includes > db.php > ...
311 // Create Blogs
312 if (isset($_POST["create_blog"])) {
313     $blog_id = $_POST['blog_id'];
314     $id = $_POST['event_id'];
315     $text = $_POST['text'];
316     $insert = mysqli_query($con, "INSERT INTO `blog`(`blog_id`, `event_id`, `text`) VALUES ('$blog_id', '$id','$text')");
317     if ($insert) {
318         header("location:../Blogs.php");
319         exit();
320     } else {
321         echo "Blog not inserted";
322     }
323 }
324
325 //Delete Blogs
326 if (isset($_POST["deleteBlog"])) {
327     $Blog_id = $_POST["blog_id"];
328     $delete_blog = mysqli_query($con, "UPDATE `blog` SET `is_deleted`= 1 WHERE `blog_id` = '$Blog_id'");
329     if ($delete_blog) {
330         header("location: ../blogs.php");
331         exit;
332     }
333 }
334 }
```

Figure13: Creating and deleting Event Categories:

```
admin > includes > db.php > ...
529 //Add event categories
530 if (isset($_POST['create_category'])) {
531     $category_id = $_POST["category_id"];
532     $name = $_POST['category_name'];
533     $insert = mysqli_query($con, "INSERT INTO `event_categories`(`category_id`, `name`) VALUES ('$category_id', '$name')");
534     if ($insert) {
535         header("location:../categories.php");
536         exit();
537     } else {
538         echo "category not inserted";
539     }
540 }
541 // delete event categories
542 if (isset($_POST["delete_Category"])) {
543     $category_id = $_POST["category_id"];
544     $DeleteAddon = mysqli_query($con, "UPDATE `event_categories` SET `is_deleted`= 1 WHERE `category_id` = '$category_id'");
545     if ($DeleteAddon) {
546         header("location: ../categories.php");
547         exit();
548     }
549 }
550 ?>
```

Figure14: Creating Editing and Deleting Event Tickets:

```

admin > includes > db.php > ...
338 // Create Ticket
339 if (isset($_POST['create_ticket'])) {
340     $event_id = $_POST['event_id'];
341     $ticket_type_id = $_POST['ticket_type_id'];
342     $discount = $_POST['discount'];
343     // Fetch event capacity
344     $fetch_event = mysqli_query($con, "SELECT `capacity` FROM `events` WHERE `event_id` = '$event_id'");
345     $events = mysqli_fetch_assoc($fetch_event);
346     $capacity = $events['capacity'];
347     // Get the number of tickets already created
348     $fetch_created_tickets = mysqli_query($con, "SELECT COUNT(*) as total_tickets FROM ticket WHERE `event_id` = '$event_id'");
349     $created_tickets = mysqli_fetch_assoc($fetch_created_tickets);
350     $remaining_capacity = $capacity - $created_tickets['total_tickets'];
351     $quantity = isset($_POST['quantity']) ? intval($_POST['quantity']) : 0;
352     if ($quantity > 0 && $quantity <= $remaining_capacity) {
353         for ($i = 0; $i < $quantity; $i++) {
354             // Insert ticket
355             $insert_ticket_query = "INSERT INTO ticket (event_id, ticket_type_id, discount, Qrcode)
356                                     VALUES ('$event_id', '$ticket_type_id', '$discount', UUID())";
357             $result_ticket = mysqli_query($con, $insert_ticket_query);
358             if ($result_ticket) {
359                 $ticket_id = mysqli_insert_id($con);
360                 // Insert selected addons for each ticket
361                 if (isset($_POST['addons']) && is_array($_POST['addons'])) {
362                     foreach ($_POST['addons'] as $addon_id) {
363                         $insertAddonQuery = "INSERT INTO ticket_addons (ticket_id, addon_id) VALUES ('$ticket_id', '$addon_id')";
364                         mysqli_query($con, $insertAddonQuery);
365                     }
366                 }
367             }
368         }
369         header("Location: ../ticket.php");
370         exit();
371     } else {
372         echo "Error: Invalid quantity or capacity exceeded!";
373         exit();
374     }
375 }

admin > includes > db.php > ...
376
377 //delete ticket
378 if (isset($_POST["deleteticket"])) {
379     $tt_id = $_POST["tt_id"];
380     $delete_ticket = mysqli_query($con, "UPDATE `ticket` SET `is_deleted` = 1 WHERE `ticket_type_id` = '$tt_id'");
381     if ($delete_ticket) {
382         header("location: ../ticket.php?tt_id=$tt_id");
383         exit();
384     }
385 }

386 //edit ticket
387 if (isset($_POST["editticket"])) {
388     $ticket_Id = $_POST["ticket_id"];
389     $ticket_type_id = $_POST["ticket_type_id"];
390     $ticket_name = $_POST["ticket_type_name"];
391     $price = $_POST["price"];
392     $discount = $_POST["discount"];
393     $addon_id = $_POST["addon_id"];
394     $capacity = $_POST["capacity"];
395     $edit_ticket = mysqli_query($con, "UPDATE `ticket` SET `ticket_type_id`='$ticket_type_id',
396                                `name`='$ticket_name', `price`='$price', `discount`='$discount', `addon_id`='$addon_id',
397                                `capacity`='$capacity' WHERE `ticket_id` = '$ticket_Id'");
398     if ($edit_ticket) {
399         header("location: ../ticket.php");
400         exit();
401     } else {
402         echo "Error: " . mysqli_error($con); // Display error, if any
403     }
404 }

405
406

```

Figure15: Creating Editing and Deleting Event Ticket Types:

```
admin > includes > db.php > ...
408 //Create ticket type
409 if (isset($_POST['create_ticket_type'])) {
410
411     $name = $_POST['name'];
412     $price = $_POST['price'];
413     $event_id = $_POST["event_id"];
414     $temp = $_POST["temp_id"];
415     $insert = mysqli_query($con, "INSERT INTO `ticket_type`(`name`, `price`, `event_id`, `temp_id`)
416     VALUES ('$name','$price','$event_id','$temp')");
417     if ($insert) {
418         header("location:../ticket_types.php");
419         exit();
420     } else {
421         echo "User not inserted";
422     }
423 }
424
425 //delete ticket type
426 if (isset($_POST["deleteTicket_type"])) {
427     $ticket_type_id = $_POST["ticket_type_id"];
428     $ticket_type = mysqli_query($con, "UPDATE `ticket_type` SET `is_deleted` = 1
429     WHERE `ticket_type_id` = '$ticket_type_id'");
430     if ($ticket_type) {
431         header("location: ../ticket_types.php");
432         exit();
433     }
434 }
435
admin > includes > db.php > ...
435 //edit ticket type
436 if (isset($_POST["edit_ticket_type"])) {
437     $ticket_type_id = $_POST["ticket_type_id"];
438     $event_id = $_POST["event_id"];
439     $name = $_POST["name"];
440     $price = $_POST["price"];
441     $ticket_type = mysqli_query($con, "UPDATE `ticket_type`
442     SET `name`='$name', `price`='$price' ,`event_id` = '$event_id'
443     WHERE `ticket_type_id` = '$ticket_type_id' ");
444     if ($ticket_type) {
445         header("location: ../ticket_types.php");
446         exit;
447     } else {
448         echo "Error: " . mysqli_error($con); // Display error, if any
449     }
450 }
```

Figure16: Creating Editing and Deleting Ticket Add-on:

```
admin > includes > db.php > ...
454 //create addon
455 if (isset($_POST['createAddon'])) {
456
457     $name = $_POST['name'];
458     $price = $_POST['price'];
459     $organizer = $_SESSION['Organizer']['id'];
460     $insert = mysqli_query($con, "INSERT INTO `addon`(`organizer_id`, `name`, `price`) VALUES ('$Organizer', '$name', '$price')");
461     if ($insert) {
462         header("location:../Addons.php");
463         exit();
464     } else {
465         echo "User not inserted";
466     }
467 }
468
469 // delete addon
470 if (isset($_POST["deleteAddon"])) {
471     $addon_id = $_POST["addon_id"];
472     $DeleteAddon = mysqli_query($con, "UPDATE `addon` SET `is_deleted` = 1 WHERE `Addon_id` = '$addon_id'");
473     if ($DeleteAddon) {
474         header("location: ../Addons.php");
475         exit();
476     }
477 }
478 // edit addon
479 if (isset($_POST["editAddon"])) {
480     $addon_id = $_POST["addon_id"];
481     $name = $_POST["name"];
482     $price = $_POST["price"];
483     $addon = mysqli_query($con, "UPDATE `addon` SET `name`='$name', `price`='$price' WHERE `Addon_id` = '$addon_id' ");
484     if ($addon) {
485         header("location: ../Addons.php");
486         exit();
487     } else {
488         echo "Error: " . mysqli_error($con); // Display error, if any
489     }
490 }
```

Figure17: Creating Question Form:

```
admin > includes > db.php > ...
495 // create question form
496 if (isset($_POST["add_form"])) {
497     $form_id = $_POST["form_id"];
498     $event_id = $_POST["event_id"];
499     $organizer = $_SESSION['Organizer']['id'];
500     $insertform = mysqli_query($con, "INSERT INTO `registration_form`"
501     ("form_id", `event_id`, `organizer_id`) VALUES ('$form_id', '$event_id', '$organizer')");
502     if ($insertform) {
503         header("location: ../question_form.php");
504         exit();
505     } else {
506         echo "form not inserted";
507     }
508 }
509
```

Figure18: Creating and Deleting Registration Questions:

```
admin > includes > db.php > ...
512 // Add registration question
513 if (isset($_POST["add_questions"])) {
514
515     $question_id = $_POST["question_id"];
516     $form_id = $_POST["form_id"];
517     $questionsArray = $_POST["Question"];
518     foreach ($questionsArray as $Question) {
519         $Question = mysqli_real_escape_string($con, $Question);
520         $insertRegistration = mysqli_query($con, "INSERT INTO `registrationquestions`"
521         ("question_id", `form_id`, `Question`) VALUES ('$question_id', '$form_id', '$Question')");
522     }
523     header("location: ../question_form.php");
524     exit();
525 }
526
527 // delete registration question
528 if (isset($_POST["delete_question"])) {
529     $question_id = $_POST["question_id"];
530     $Deletequestion = mysqli_query($con, "UPDATE `registrationquestions` SET `is_deleted` = 1"
531     WHERE `question_id` = '$question_id'");
532     if ($Deletequestion) {
533         header("location: ../view_questions.php");
534         exit();
535     }
536 }
```

5.3. External Libraries:

5.3.1. Bootstrap:

Import Code:

html

```
<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
```

Brief Information:

Bootstrap is used for responsive and mobile-friendly UI design, providing pre-built CSS components.

5.3.2. PHP Mailer:

Import Code:

php

```
require_once 'path/to/PHPMailer/PHPMailerAutoload.php';
```

Brief Information:

PHPMailer is utilized for sending emails from PHP scripts, facilitating communication with users.

5.3.3. jQuery:

Import Code:

html

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

Brief Information:

jQuery simplifies DOM manipulation and event handling, enhancing the interactivity of your web app.

5.3.4. PayPal API:

Import Code:

html

```
<script src="https://www.paypal.com/sdk/js?client-id=YOUR_CLIENT_ID"></script>
```

Brief Information:

The PayPal API integration enables secure online payment processing for ticket purchases.

5.3.5. Full Calendar:

Import Code:

html

```
<link href="https://cdn.jsdelivr.net/npm/fullcalendar@5.10.1/main.min.css" rel="stylesheet">
```

```
<script src="https://cdn.jsdelivr.net/npm/fullcalendar@5.10.1/main.min.js"></script>
```

Brief Information:

Full Calendar is used for event scheduling and display, providing an interactive calendar view.

5.4. Screenshots:

Attendee look of website:

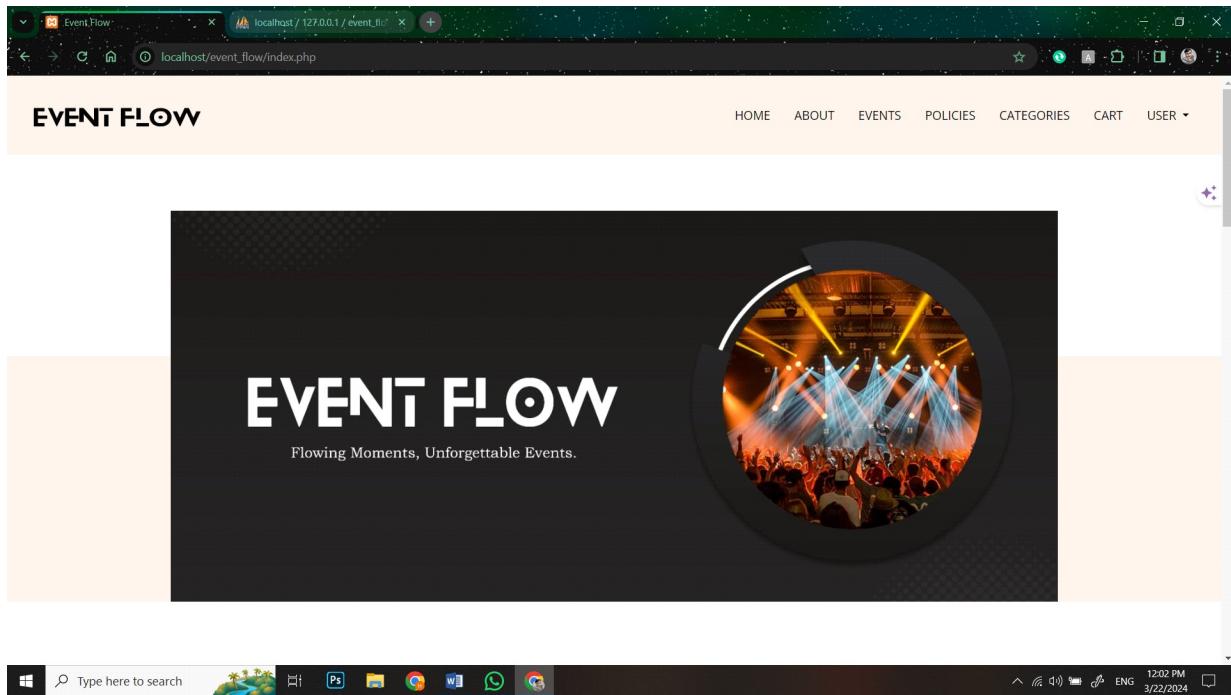


Figure19: Screen Shot 1: Main page (Header and Banner)

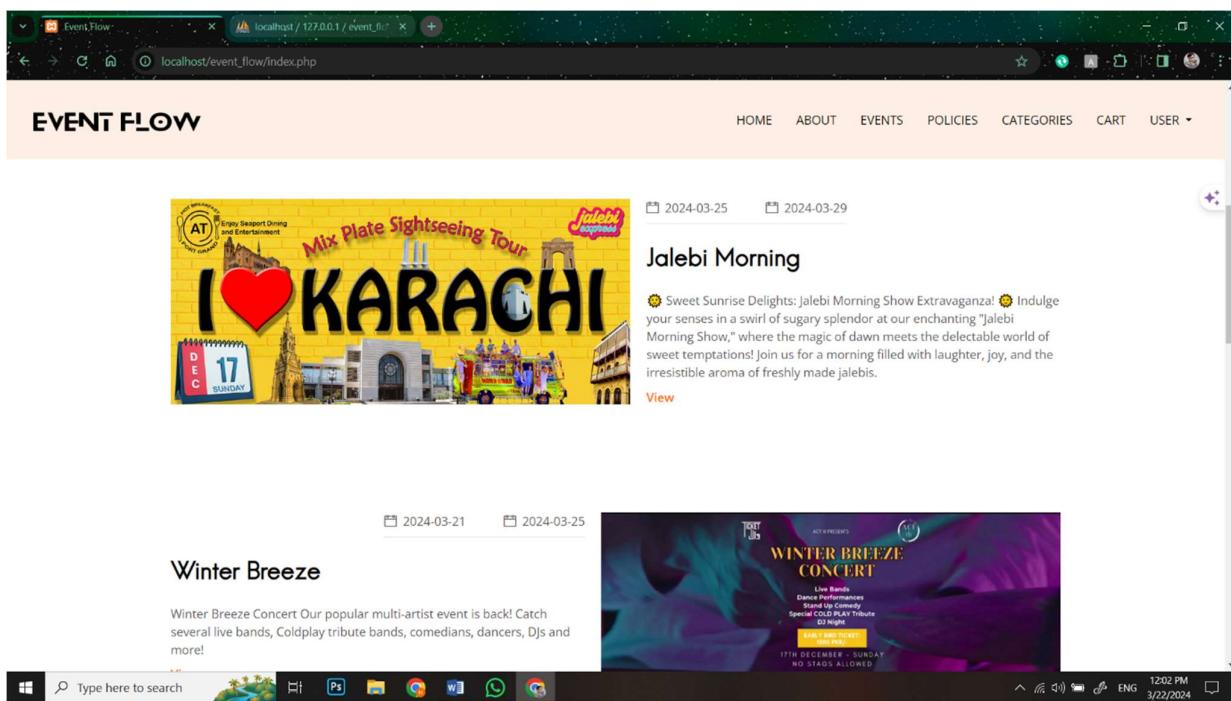


Figure20: Screen Shot 2: Main Page (Featured Events)

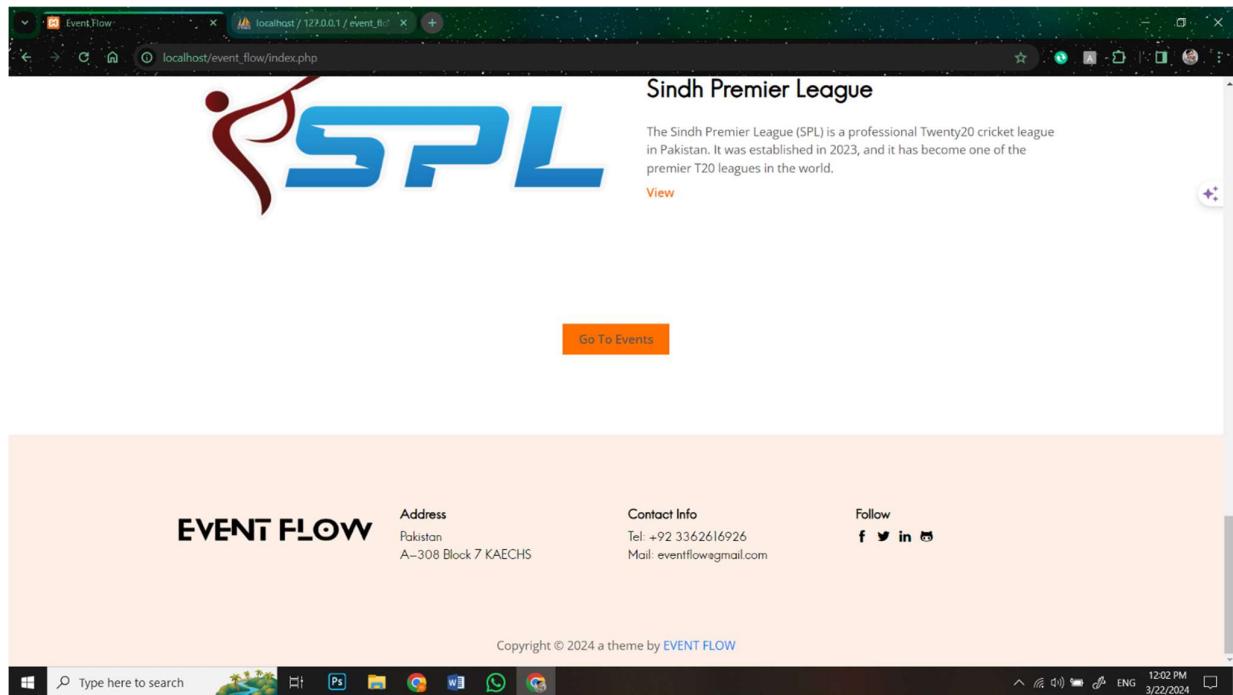


Figure21: Screen Shot 3: Main Page(Footer)

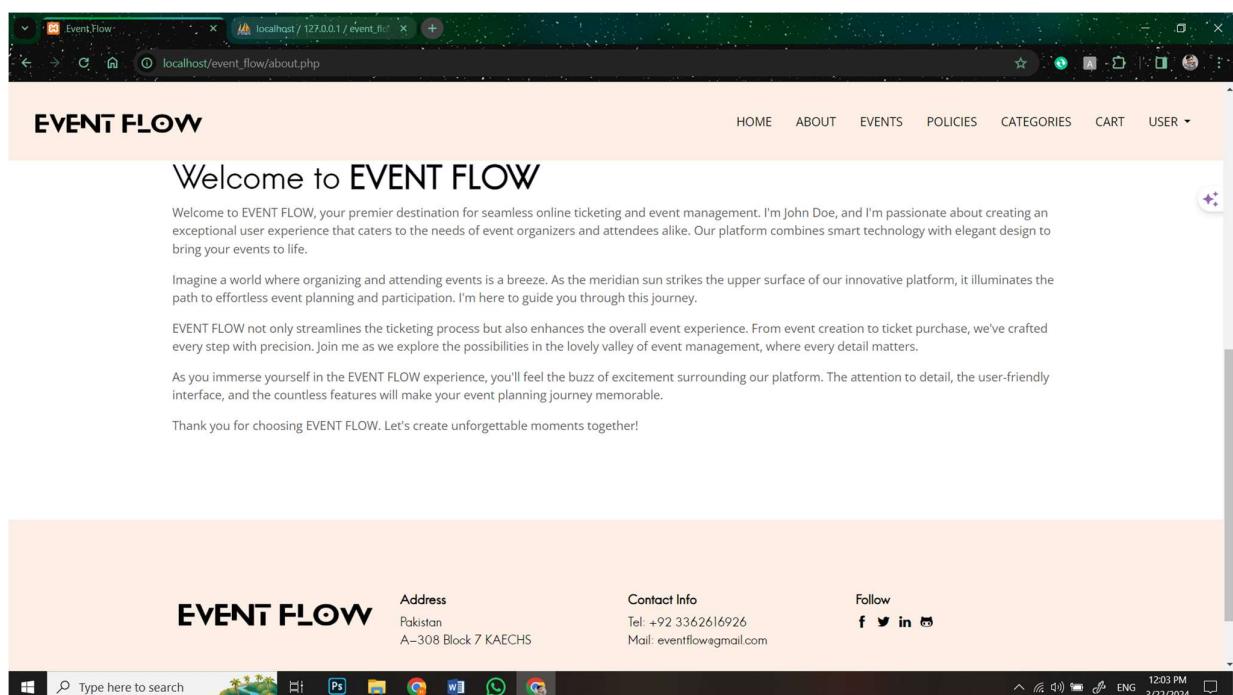


Figure22: Screen Shot 4: About Page

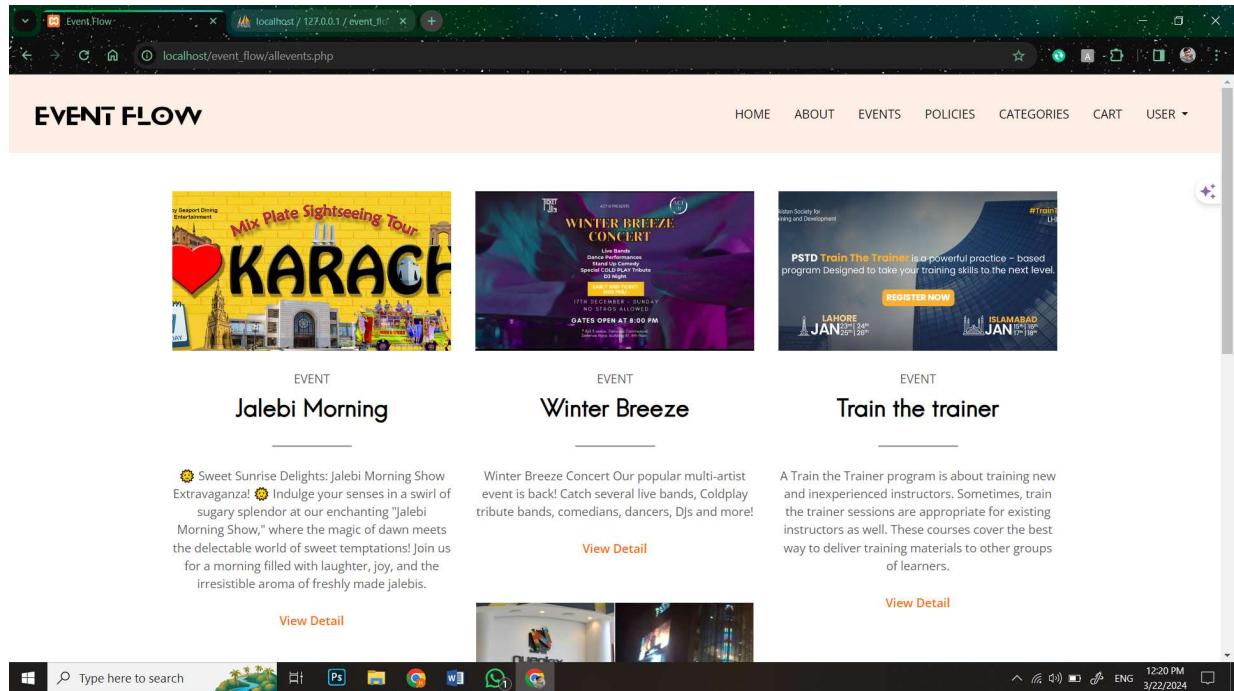


Figure23: Screen Shot 5: Events Page

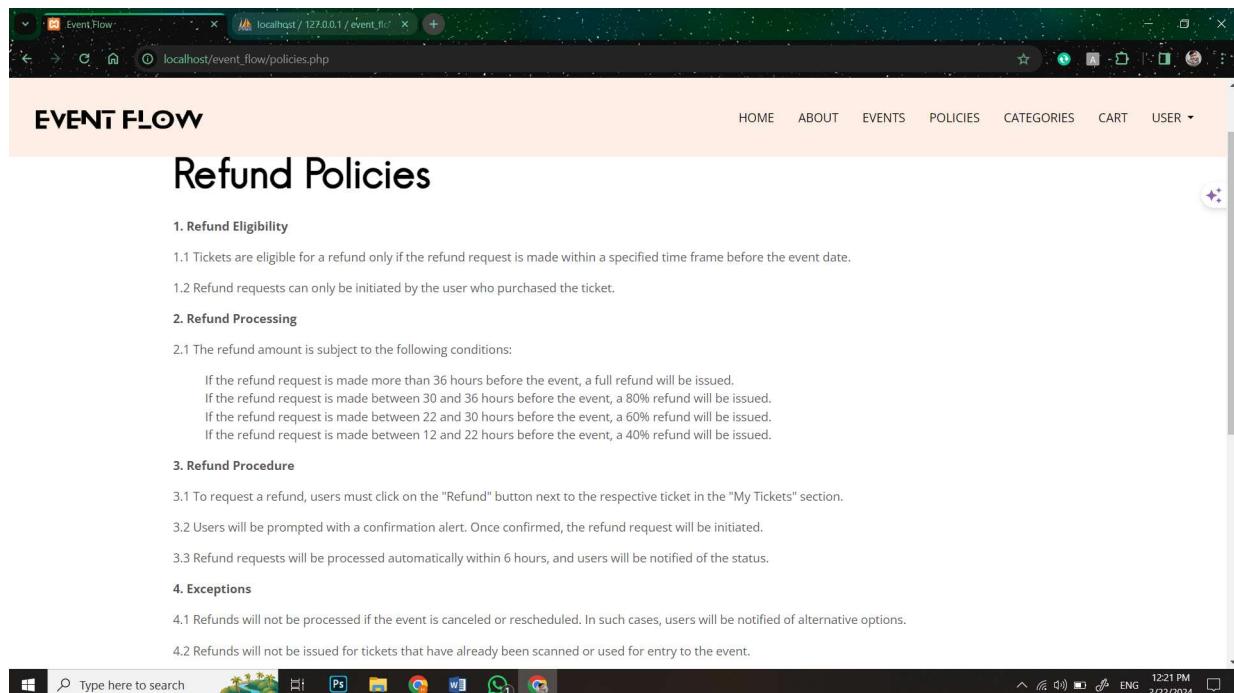


Figure34: Screen Shot 6: Policy Page

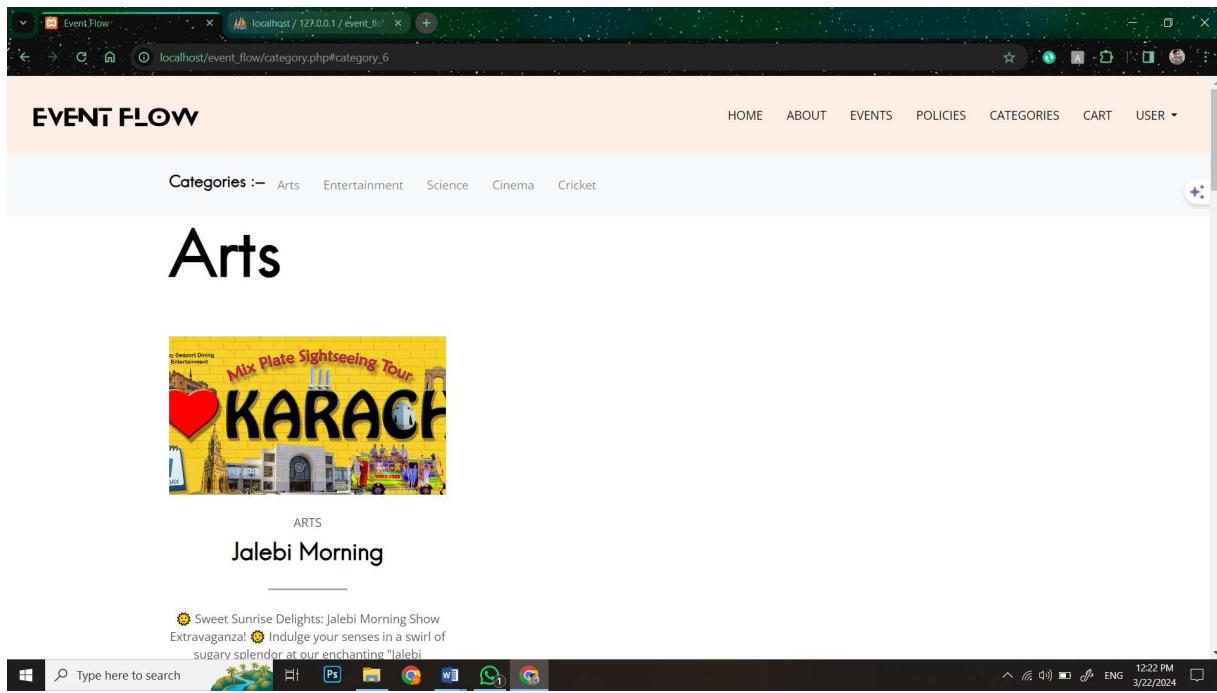


Figure35: Screen Shot 7: Event Categories Page



Figure36: Screen Shot 8:Sub Events Page

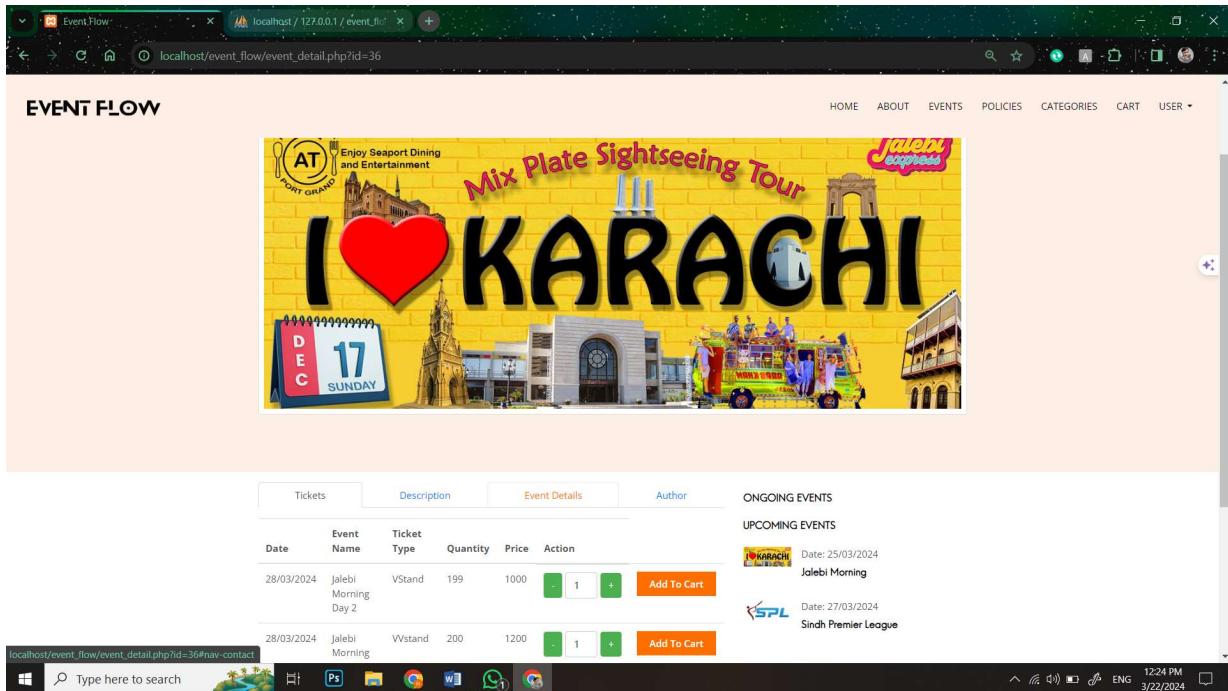


Figure37: Screen Shot 9: Event Ticket Page

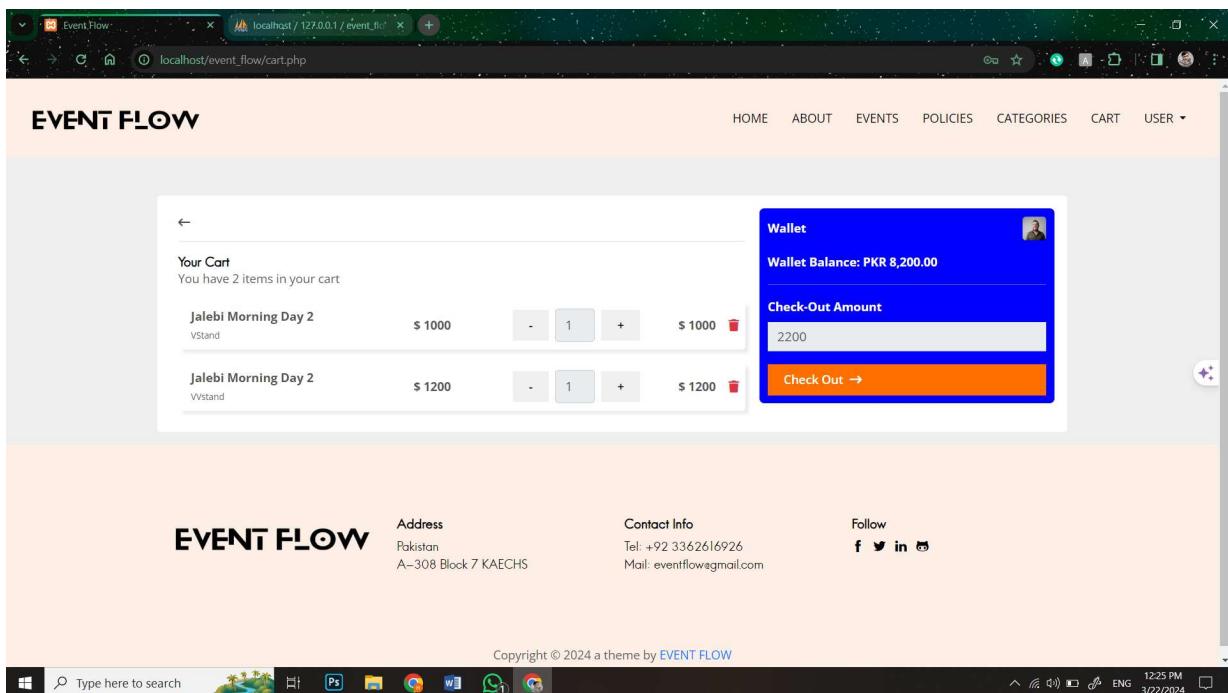


Figure38: Screen Shot 10: Cart Page

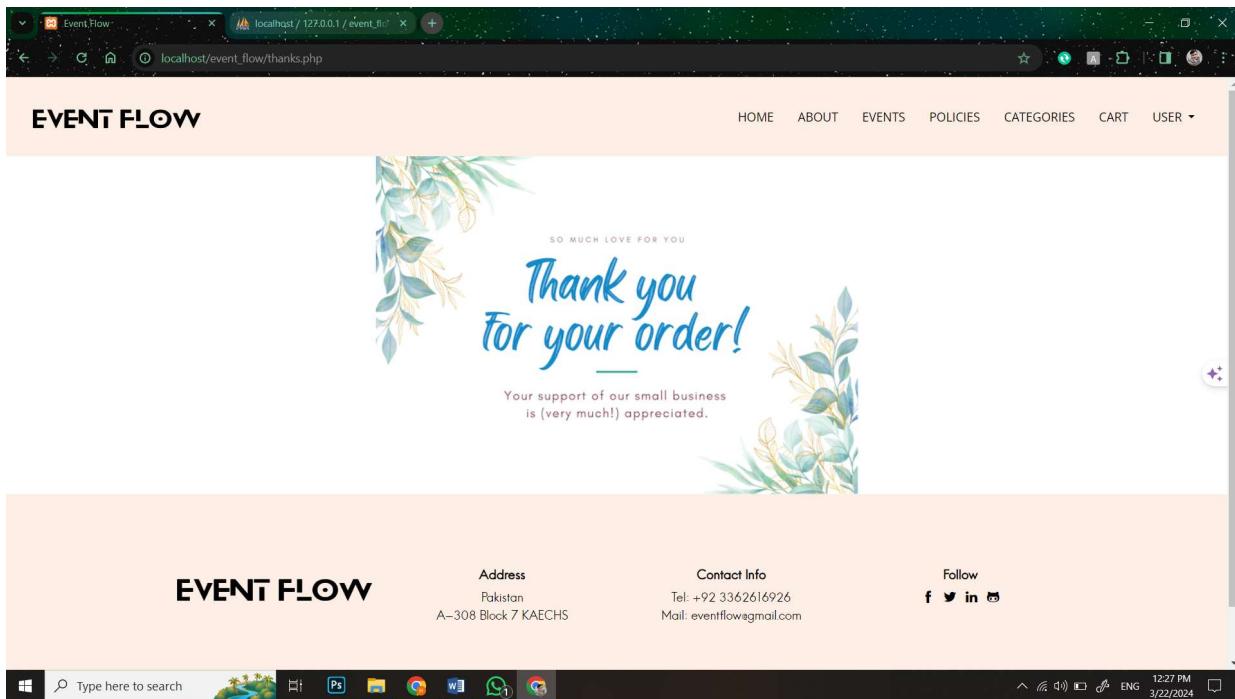


Figure39: Screen Shot 11: After checkout Page

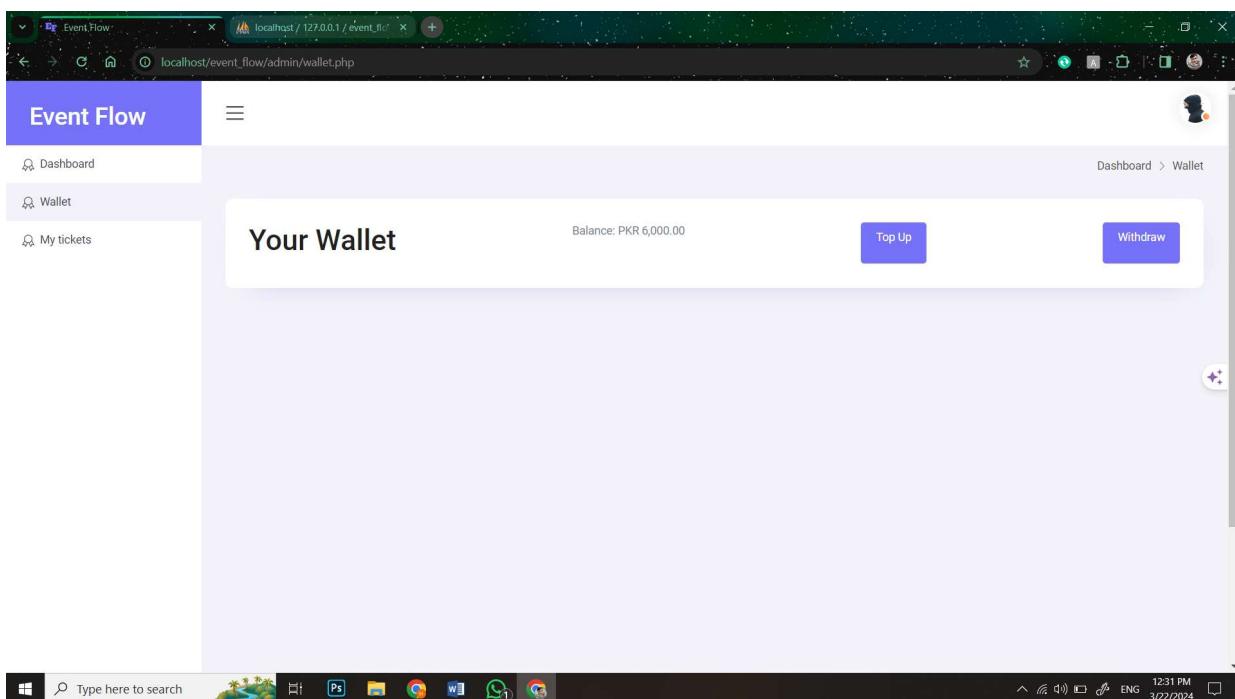


Figure40: Screen Shot 12: Wallet Page

The screenshot shows a web application interface titled "Event Flow". The left sidebar has a purple header "Event Flow" and three menu items: "Dashboard", "Wallet", and "My tickets" (which is currently selected). The main content area is titled "My tickets" and displays a table of purchased tickets. The table columns are "Event name", "ticket type", "Price", "Count", and "Action". There are two entries:

| Event name | ticket type | Price | Count | Action |
|----------------------|-------------|-------|-------|--------------------------------|
| Jalebi Morning Day 2 | VStand | 1000 | 1 | View mytickets |
| Jalebi Morning Day 2 | VVstand | 1200 | 1 | View mytickets |

Below the table, it says "Showing 1 to 2 of 2 entries". The bottom status bar shows the Windows taskbar with various icons and the date/time "3/22/2024 12:33 PM".

Figure41: Screen Shot 13: Attendee Bought tickets Page

This screenshot shows the same "Event Flow" application. The left sidebar is identical. The main content area is titled "View My tickets" and displays a table with one entry. The table columns are "Event name", "tickettype", "Price", "Event Start Time left", and "Action". There is one entry:

| Event name | tickettype | Price | Event Start Time left | Action |
|----------------------|------------|-------|------------------------|--|
| Jalebi Morning Day 2 | VStand | 1000 | 136 hours & 14 minutes | View Ticket Refund |

Below the table, it says "Showing 1 to 1 of 1 entries". The bottom status bar shows the Windows taskbar with various icons and the date/time "3/22/2024 12:32 PM".

Figure42: Screen Shot 14: Attendee Bought ticket status Page

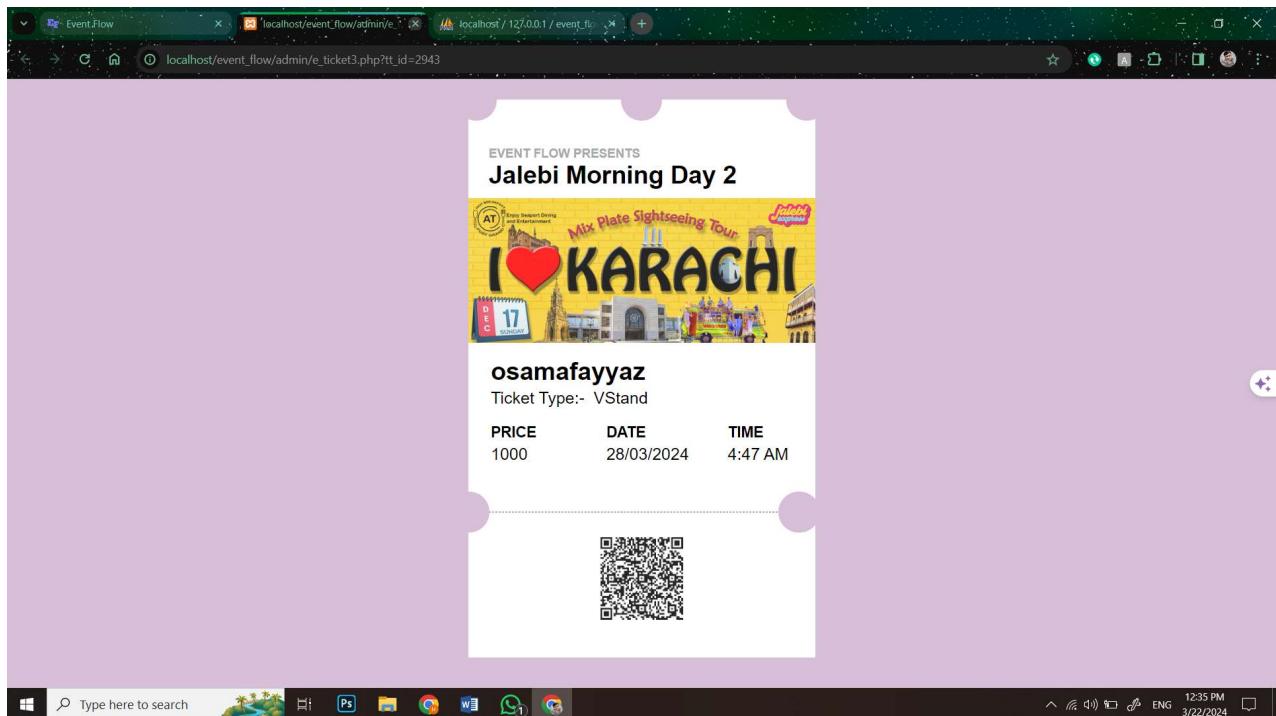


Figure43: Screen Shot 15: shown ticket Page

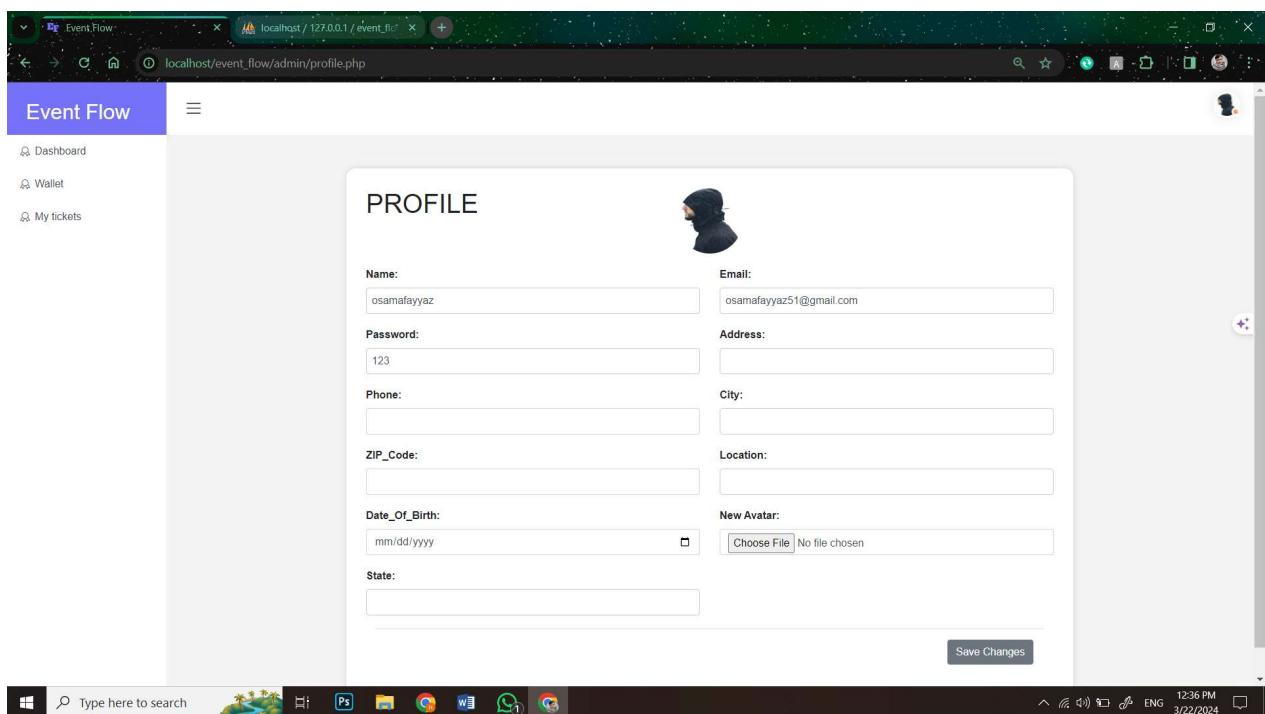


Figure44: Screen Shot 16: Attendee profile Page

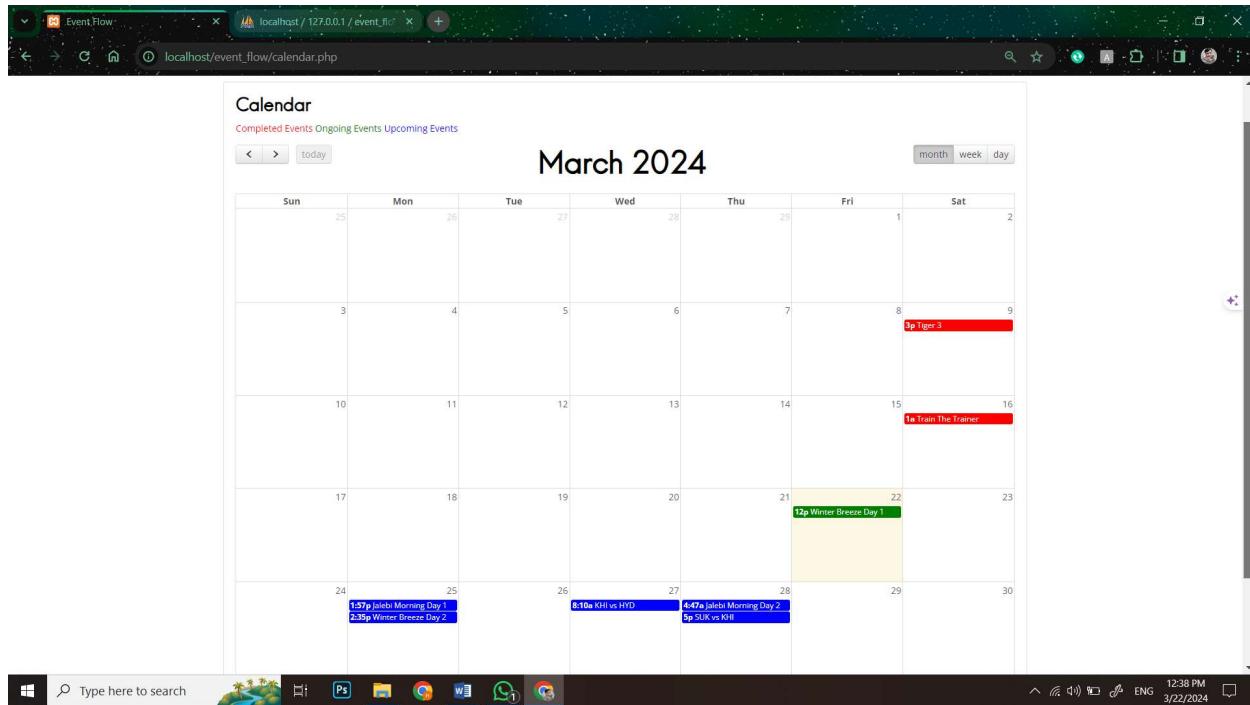


Figure45: Screen Shot 17: Calendar View Page

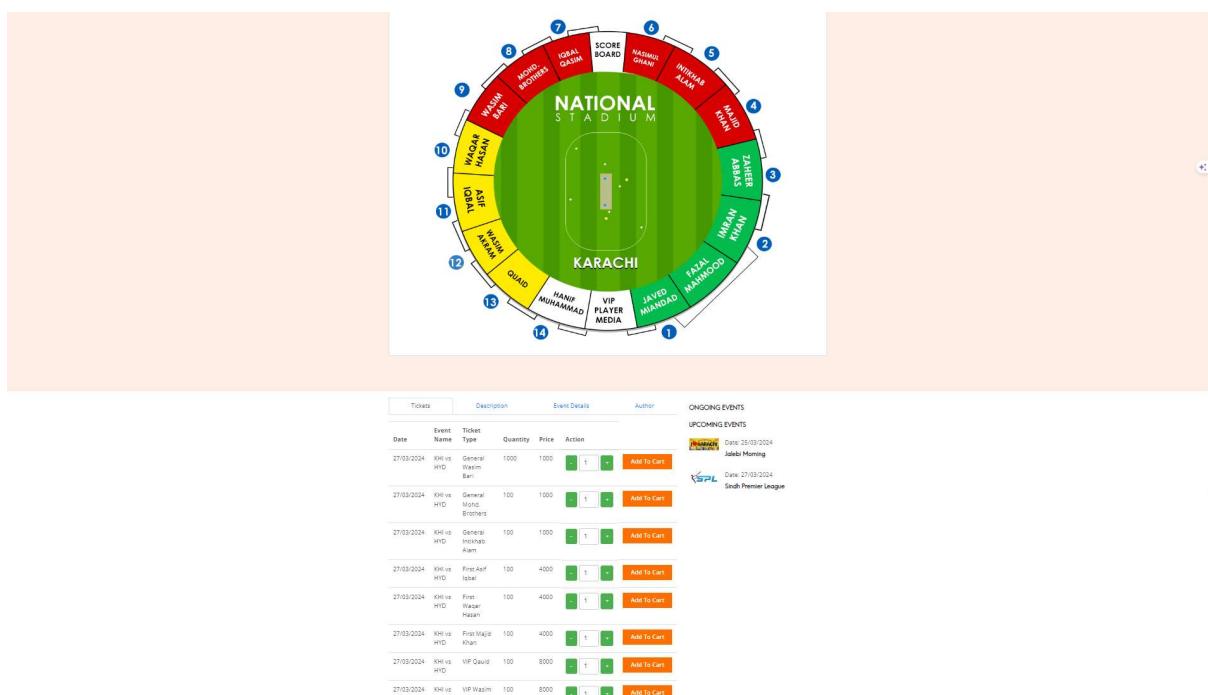


Figure46: Screen Shot 18: Matches Ticket Page

Organizer look of website:

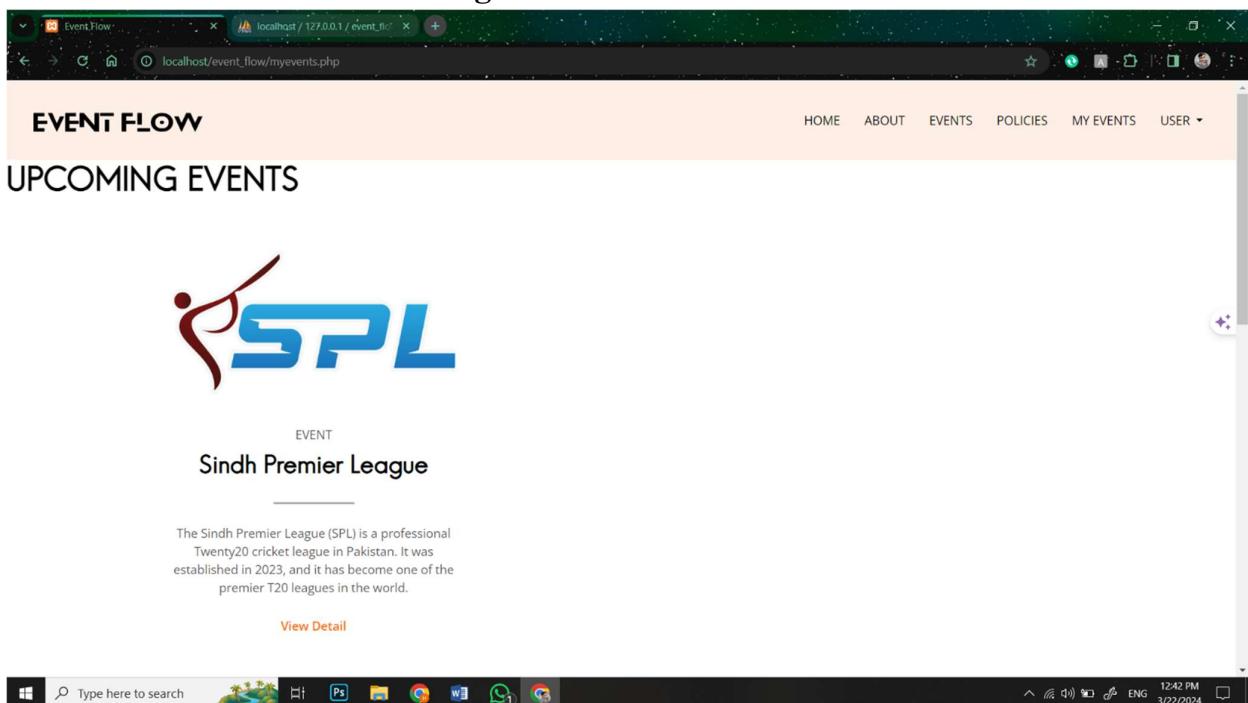


Figure47: Screen Shot 19: Organizer My Event Page1

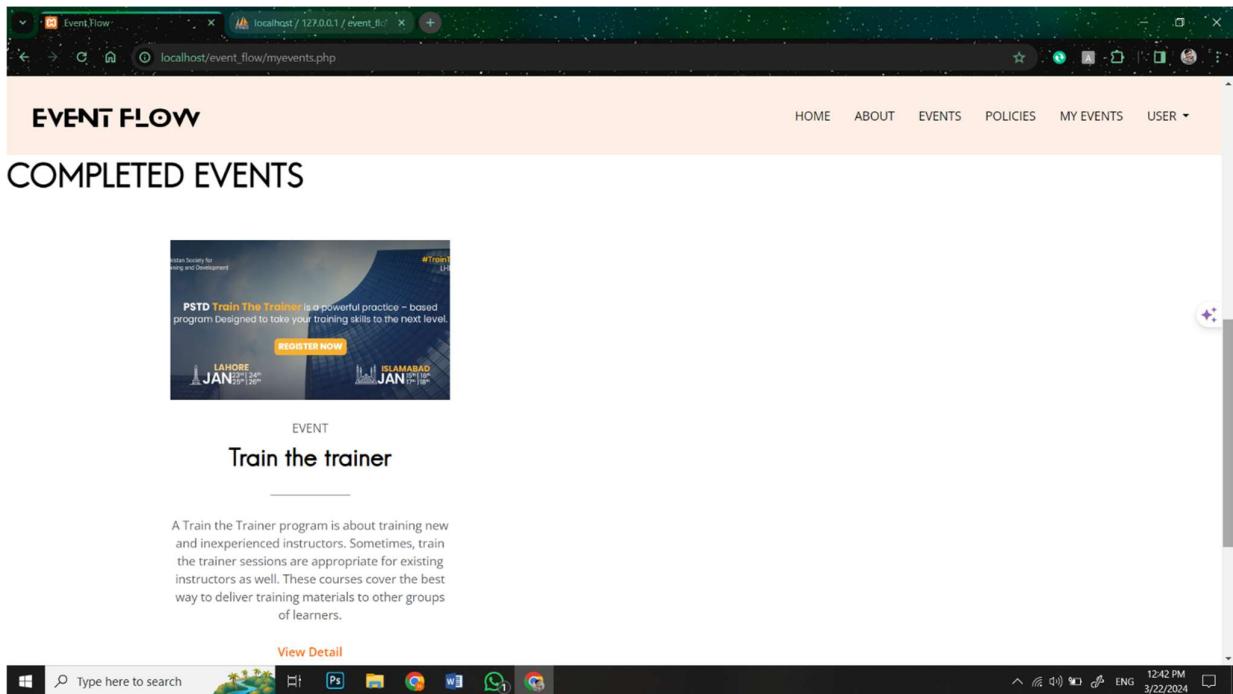


Figure48: Screen Shot 20: Organizer My Event Page2

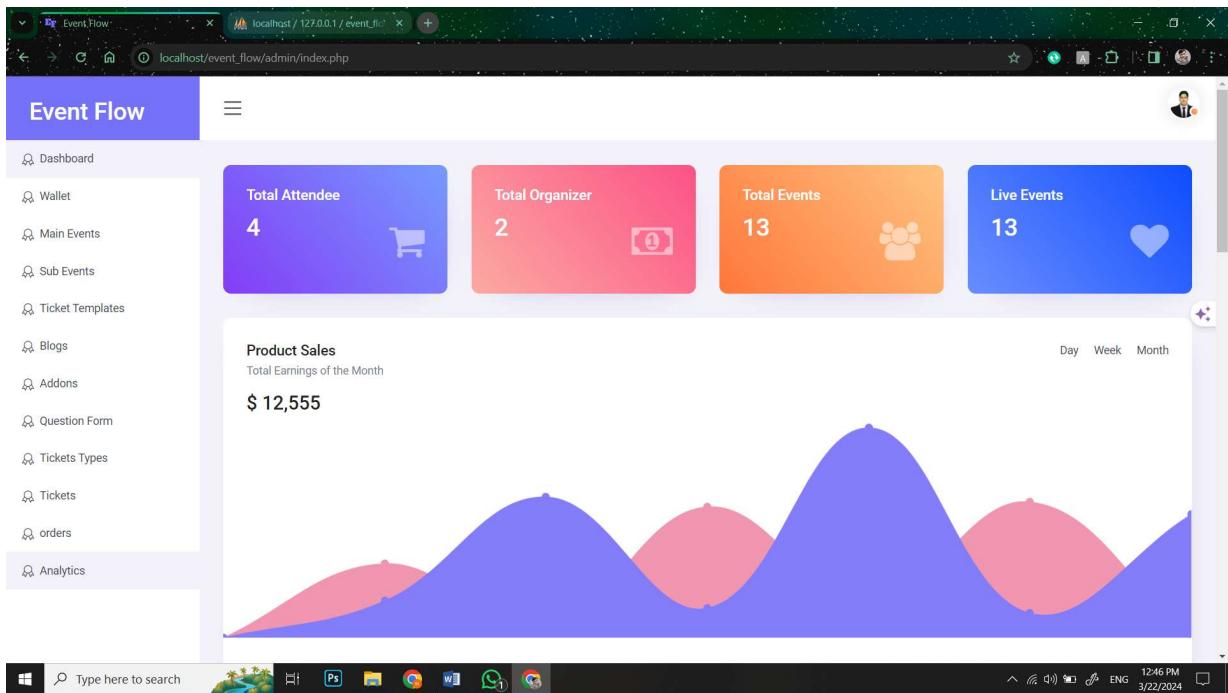


Figure49: Screen Shot 21: Organizer Dashboard

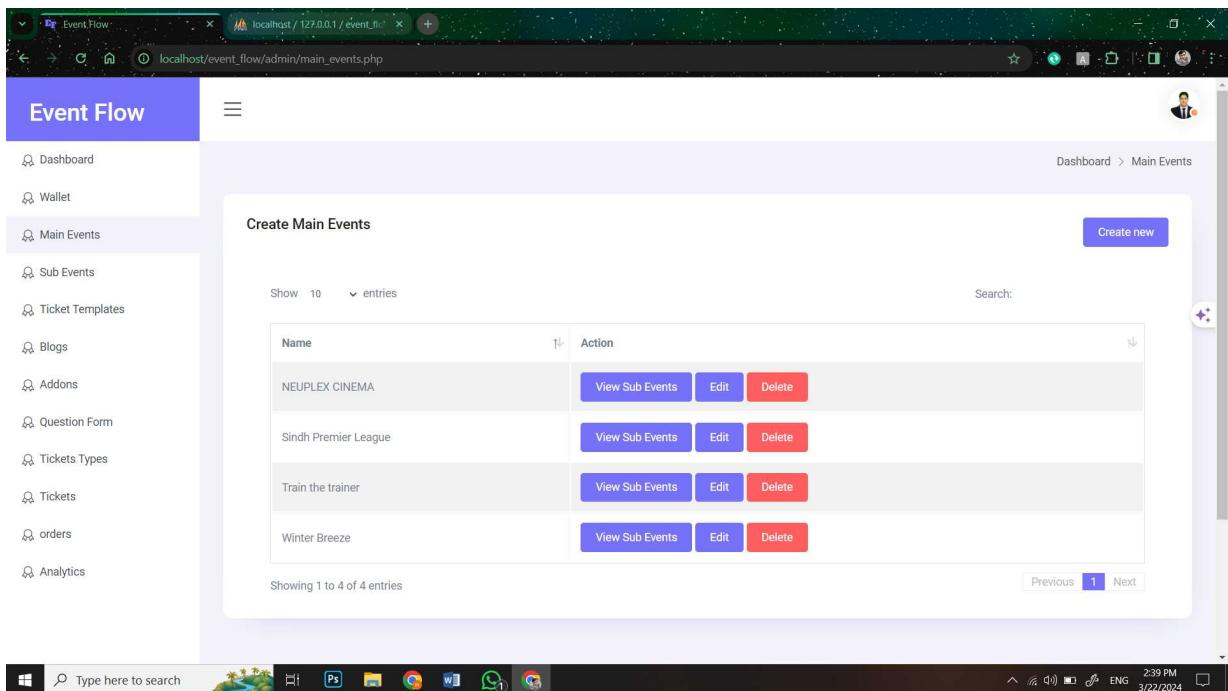


Figure50: Screen Shot 22: View Main Events

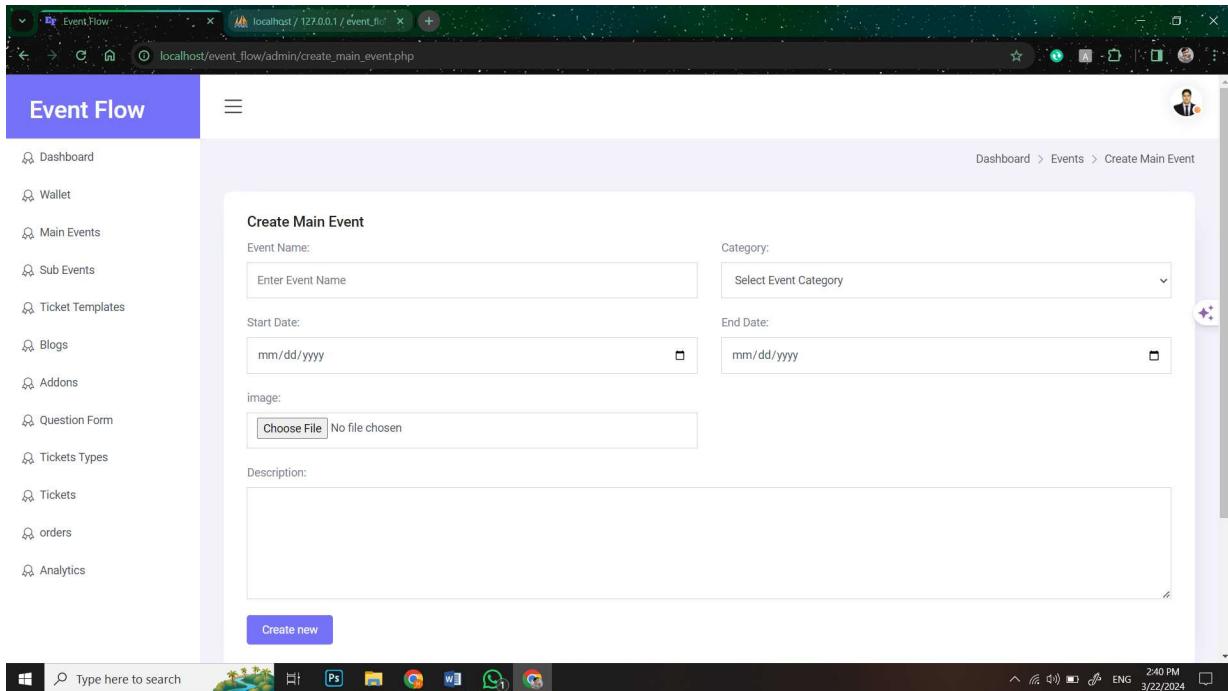


Figure51: Screen Shot 23: Create Main event

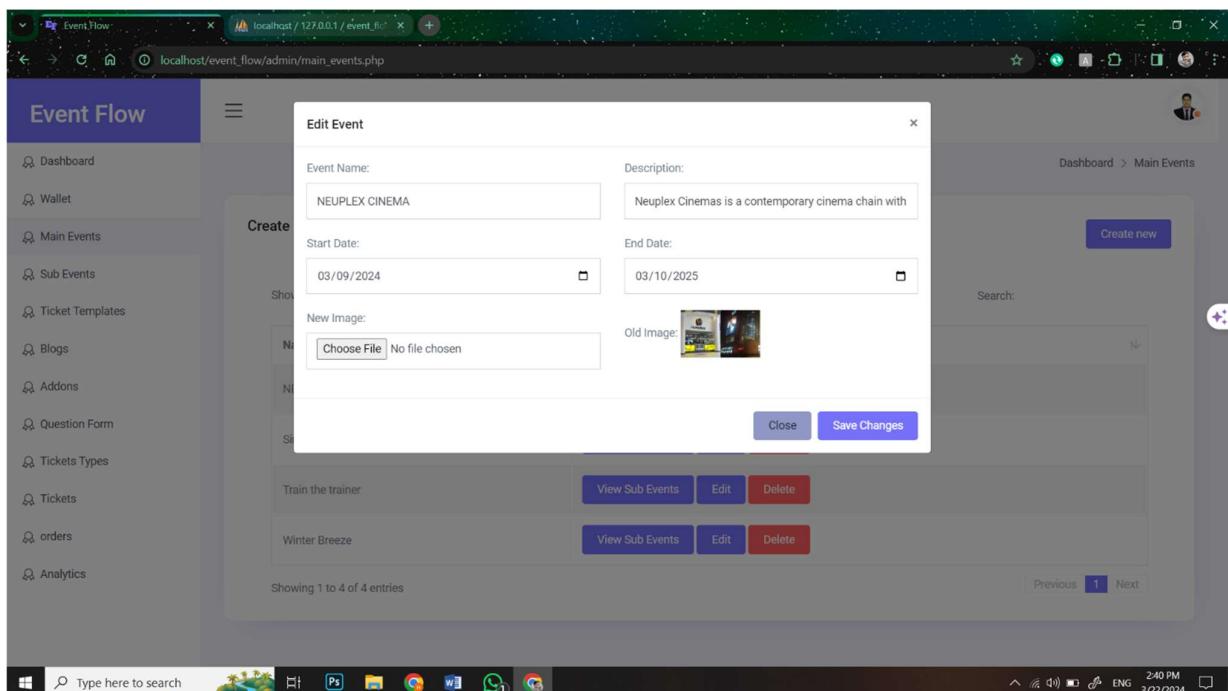


Figure52: Screen Shot 24: Edit Main event

Event Flow

Dashboard > Sub events

Sub Events

Show 10 entries Search:

| Main Event | Name | Start_date | End_date | Action |
|----------------------|---------------------|------------|------------|---|
| NEUPLEX CINEMA | Tiger 3 | 09-03-2024 | 09-03-2024 | <button>Generate Ticket</button> <button>Create Blog</button> <button>View</button> <button>Edit</button> <button>Delete</button> |
| Sindh Premier League | KHI vs HYD | 27-03-2024 | 27-03-2024 | <button>Generate Ticket</button> <button>Create Blog</button> <button>View</button> <button>Edit</button> <button>Delete</button> |
| Sindh Premier League | SUK vs KHI | 28-03-2024 | 28-03-2024 | <button>Generate Ticket</button> <button>Create Blog</button> <button>View</button> <button>Edit</button> <button>Delete</button> |
| Train the trainer | Train The Trainer | 16-03-2024 | 16-03-2024 | <button>Generate Ticket</button> <button>Create Blog</button> <button>View</button> <button>Edit</button> <button>Delete</button> |
| Winter Breeze | Winter Breeze Day 1 | 22-03-2024 | 22-03-2024 | <button>Generate Ticket</button> <button>Create Blog</button> <button>View</button> <button>Edit</button> <button>Delete</button> |
| Winter Breeze | Winter Breeze Day 2 | 25-03-2024 | 25-03-2024 | <button>Generate Ticket</button> <button>Create Blog</button> <button>View</button> <button>Edit</button> <button>Delete</button> |

localhost/event_flow/admin/create_blog.php?eid=56

Type here to search

Figure53: Screen Shot 25: View Sub event

Event Flow

Dashboard > Sub Events > Create Sub Event

Create New Sub Event

Main Event: Select Main Event Name

Event Name: Your Event Name

start_date: mm/dd/yyyy

start_time: --:-- --

end_date: mm/dd/yyyy

end_time: --:-- --

location:

capacity:

image: Choose File No file chosen

Status: Choose

Description:

Figure54: Screen Shot 26: Create Sub event

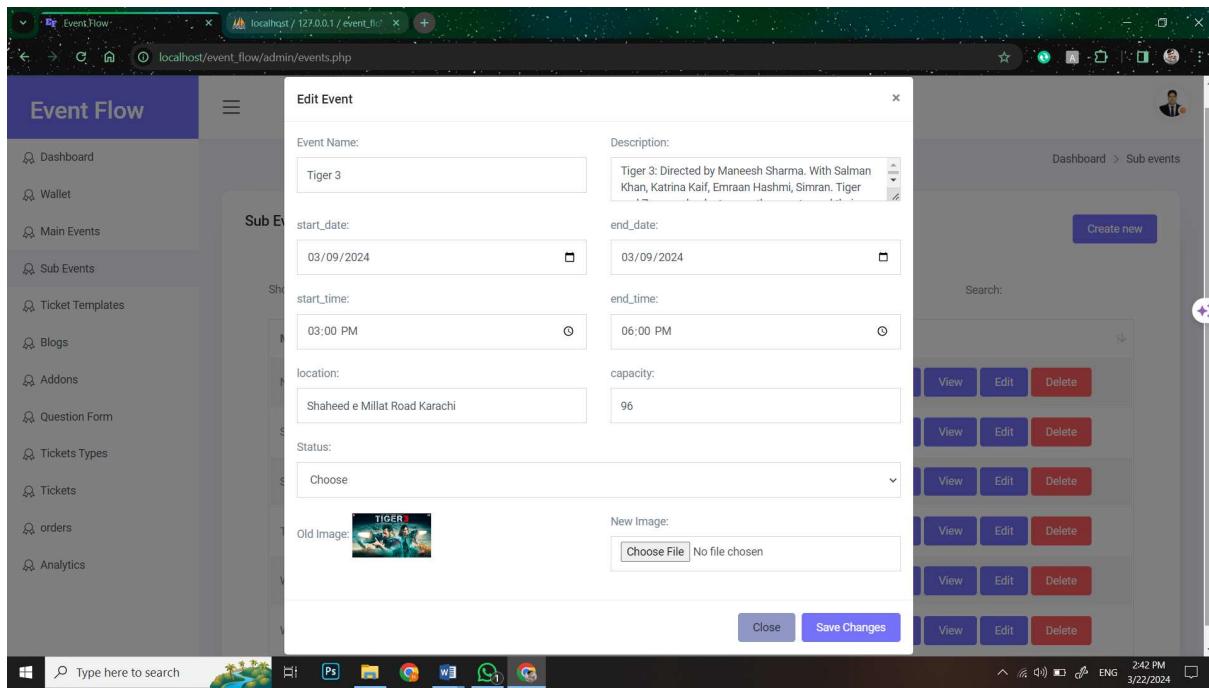


Figure55: Screen Shot 27: Edit Sub event

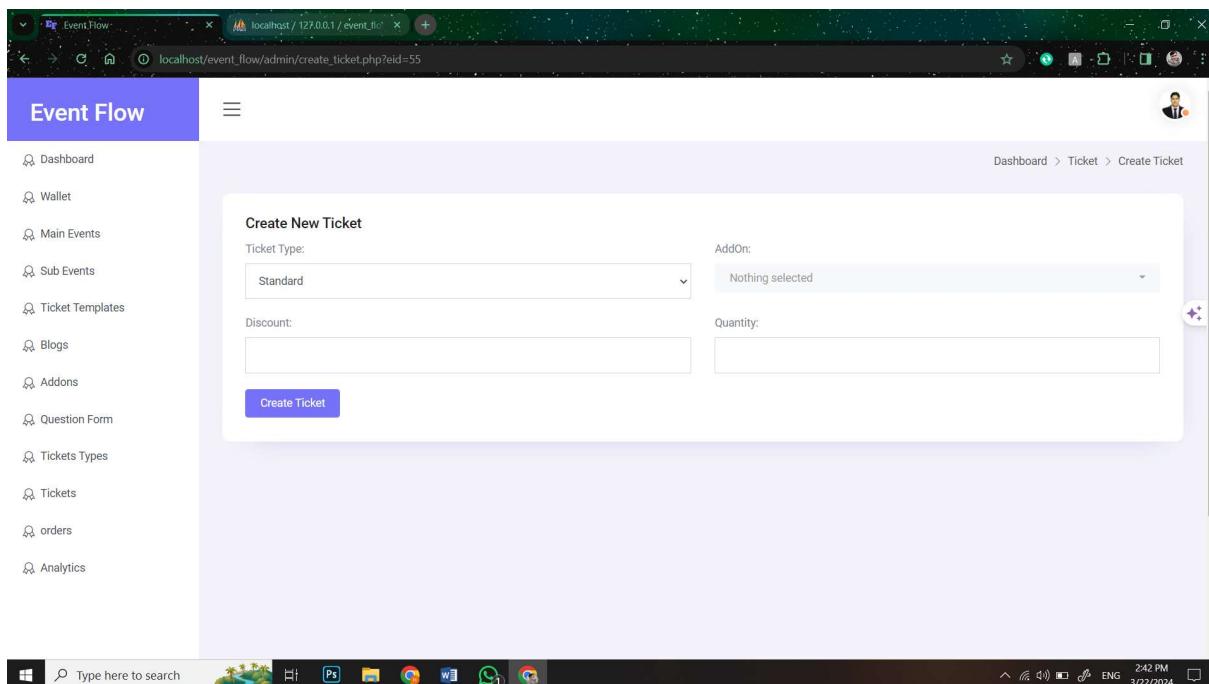


Figure56: Screen Shot 28: Create new ticket

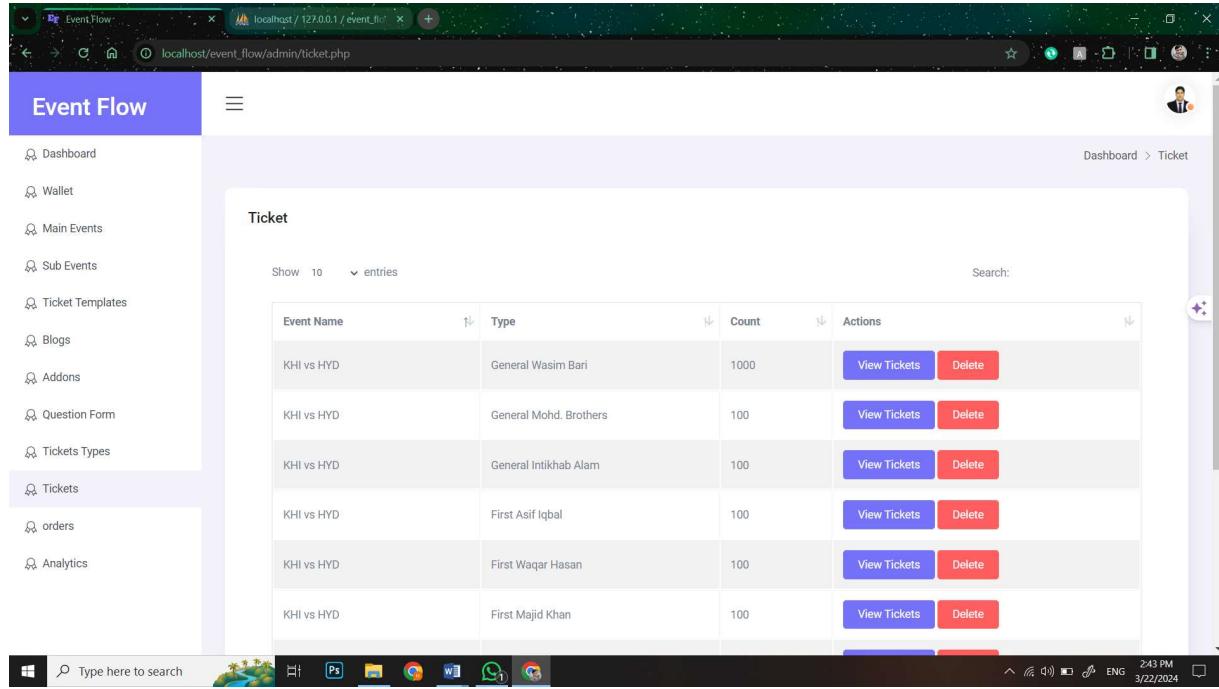


Figure57: Screen Shot 29: View different tickets

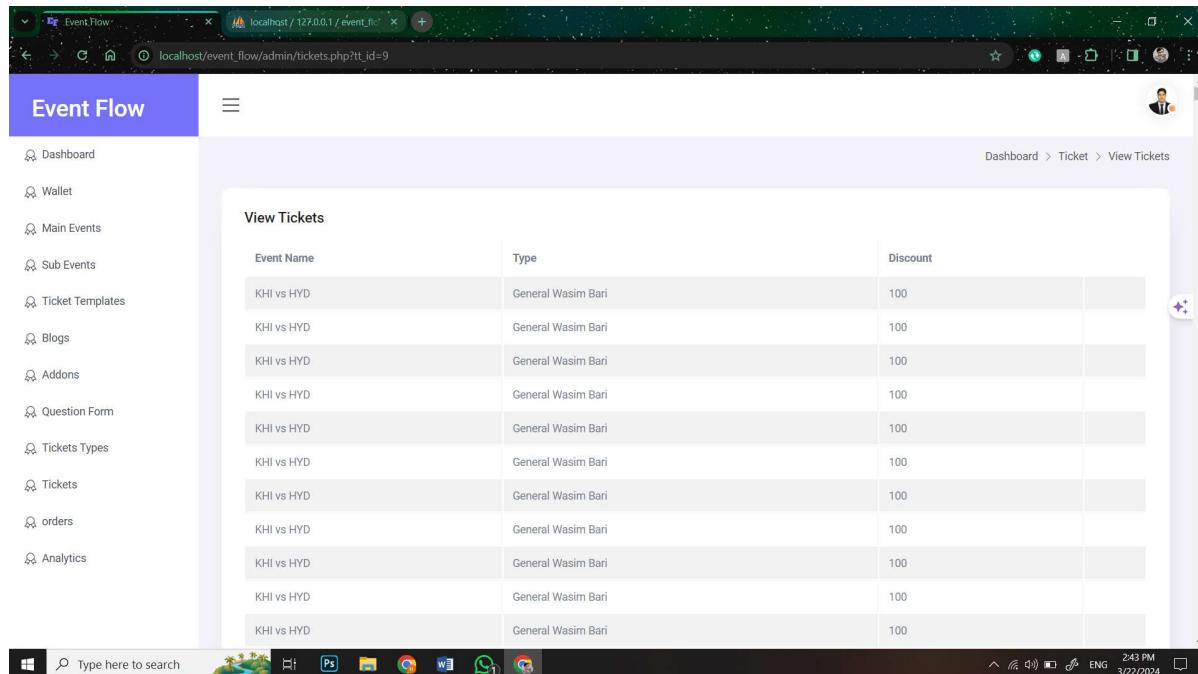


Figure58: Screen Shot 30: View quantity of each ticket

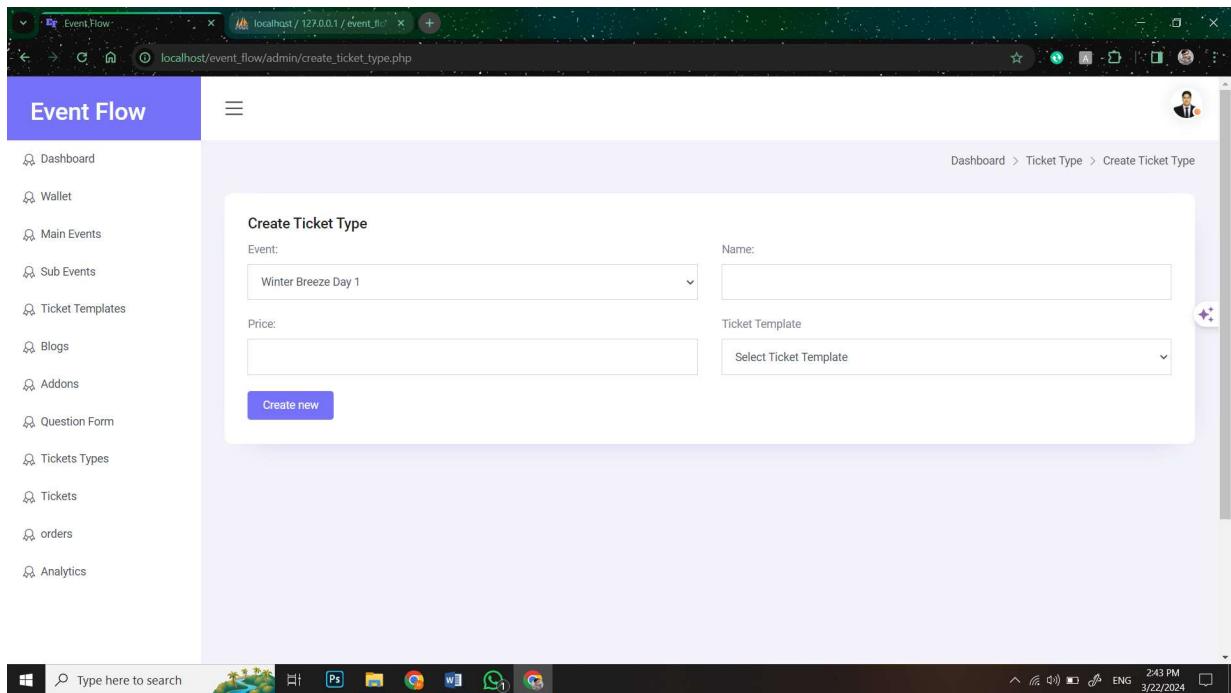


Figure59: Screen Shot 31: Create Ticket Type

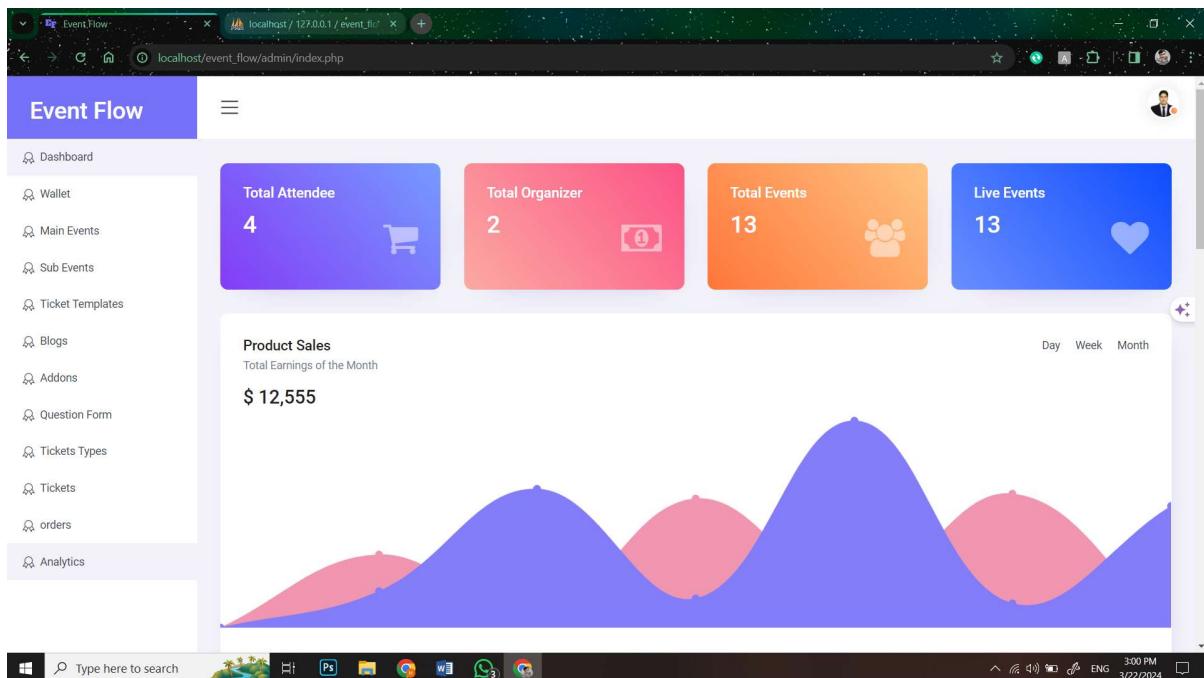


Figure60: Screen Shot 32: View Analytics and reporting

The screenshot shows the 'Event Flow' application interface. On the left, there is a sidebar with various menu items: Dashboard, Wallet, Main Events, Sub Events, Ticket Templates, Blogs, Addons (which is currently selected and highlighted in blue), Question Form, Tickets Types, Tickets, orders, and Analytics. The main content area is titled 'Addons' and displays a table of existing add-ons. The table has columns for Name, Price, and Action (with 'Edit' and 'Delete' buttons). The data in the table is as follows:

| Name | Price | Action |
|-----------|-------|-------------|
| capachino | 450 | Edit Delete |
| Coffee | 200 | Edit Delete |
| Tea | 80 | Edit Delete |
| Wine | 600 | Edit Delete |

At the bottom of the table, it says 'Showing 1 to 4 of 4 entries'. The top right corner of the main content area has a 'Create new' button. The top navigation bar shows the URL 'localhost/127.0.0.1/event_flow/admin/Addons.php' and the title 'Event Flow'. The system tray at the bottom right shows the date and time as '3/22/2024 2:44 PM'.

Figure61: Screen Shot 33: View Add-Ons

The screenshot shows the 'Event Flow' application interface. The sidebar is identical to Figure 33, with 'Addons' selected. The main content area is titled 'Create Addon'. It contains two input fields: 'Name:' and 'Price:', each with an associated text input box. Below these fields is a 'Create new' button. The top navigation bar shows the URL 'localhost/127.0.0.1/event_flow/admin/create_addon.php' and the title 'Event Flow'. The system tray at the bottom right shows the date and time as '3/22/2024 2:44 PM'.

Figure62: Screen Shot 34: Create Add-Ons

The screenshot shows a web browser window titled "Event Flow" with the URL "localhost/127.0.0.1/event_flow". The left sidebar contains a navigation menu with items like Dashboard, Wallet, Main Events, Sub Events, Ticket Templates, Blogs (which is selected and highlighted in blue), Addons, Question Form, Tickets Types, Tickets, orders, and Analytics. The main content area is titled "Blogs" and displays a table of "Sub Event name" entries. The table has columns for "Sub Event name" and "Action". The entries are:

| Sub Event name | Action |
|---------------------|---|
| KHI vs HYD | <button>View</button> <button>Delete</button> |
| SUK vs KHI | <button>View</button> <button>Delete</button> |
| Tiger 3 | <button>View</button> <button>Delete</button> |
| Train The Trainer | <button>View</button> <button>Delete</button> |
| Winter Breeze Day 1 | <button>View</button> <button>Delete</button> |
| Winter Breeze Day 2 | <button>View</button> <button>Delete</button> |

At the bottom, there is a message "Showing 1 to 6 of 6 entries" and a navigation bar with "Previous" and "Next" buttons.

Figure63: Screen Shot 35: View Blogs

The screenshot shows a web browser window titled "Event Flow" with the URL "localhost/127.0.0.1/event_flow/admin/create_blog.php?eid=55". The left sidebar is identical to Figure 35. The main content area is titled "Create New Blog" and features a rich text editor toolbar at the top. Below the toolbar is a large text input field. At the bottom right of the form is a blue "Create New" button.

Figure64: Screen Shot 36: Create Blogs

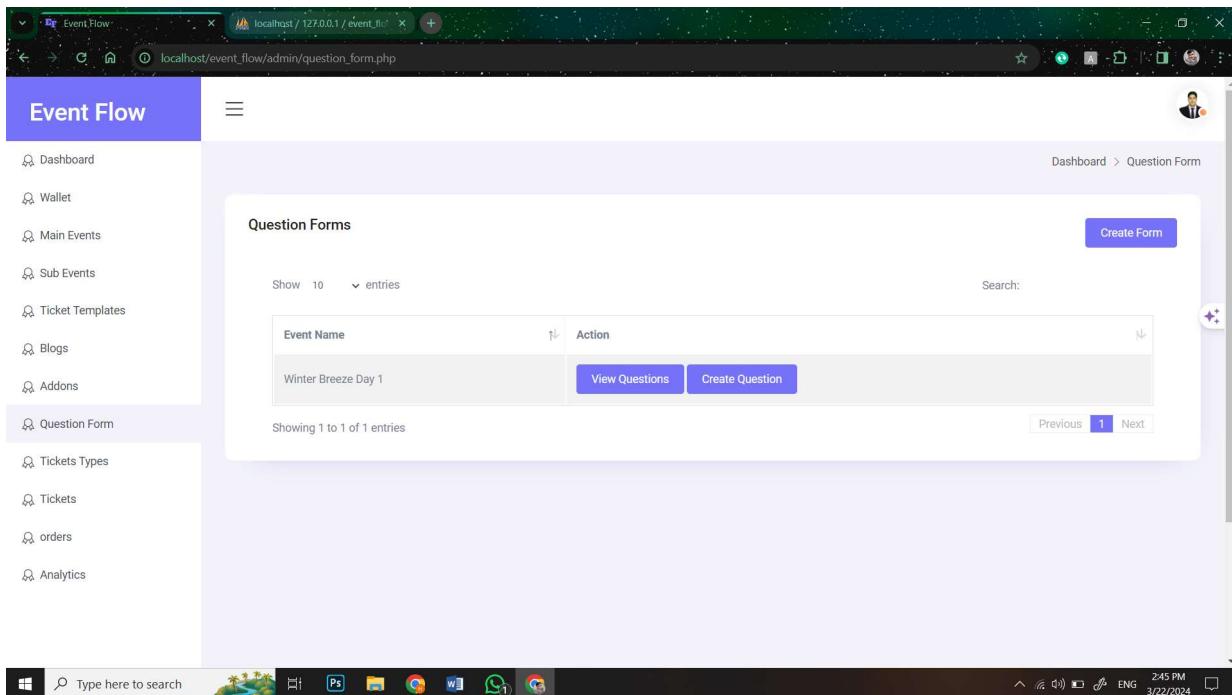


Figure65: Screen Shot 37: View Question form

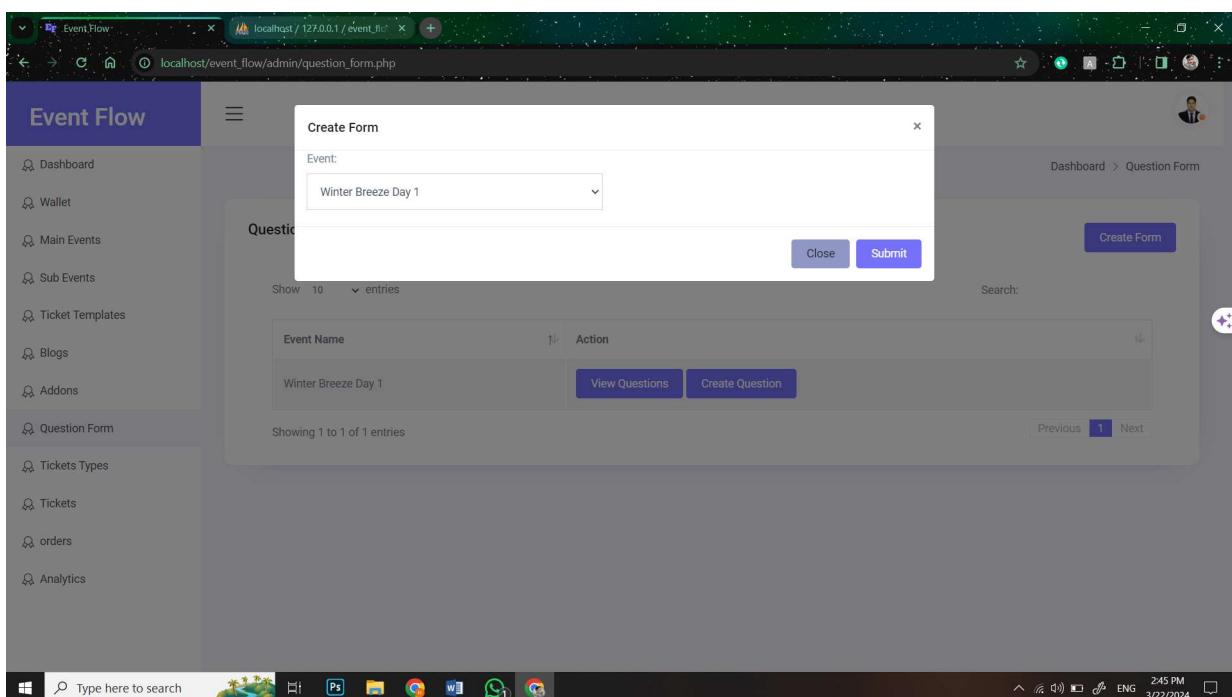


Figure66: Screen Shot 38: Create Question Form

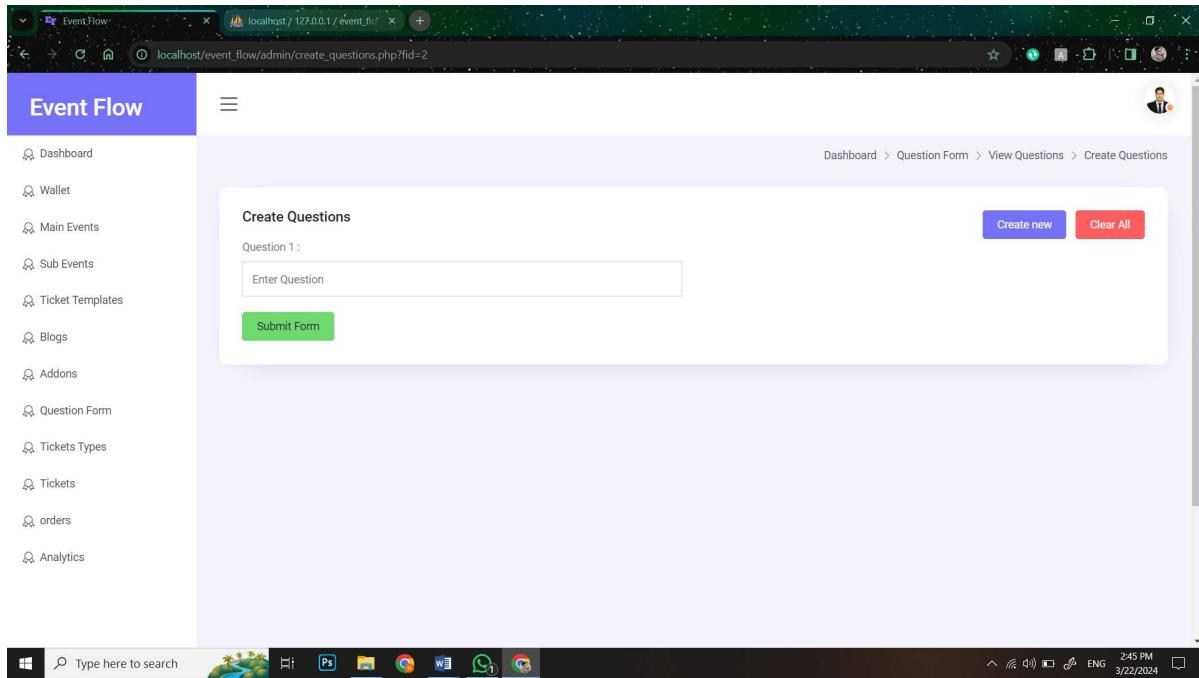


Figure67: Screen Shot 39: Create Form Questions

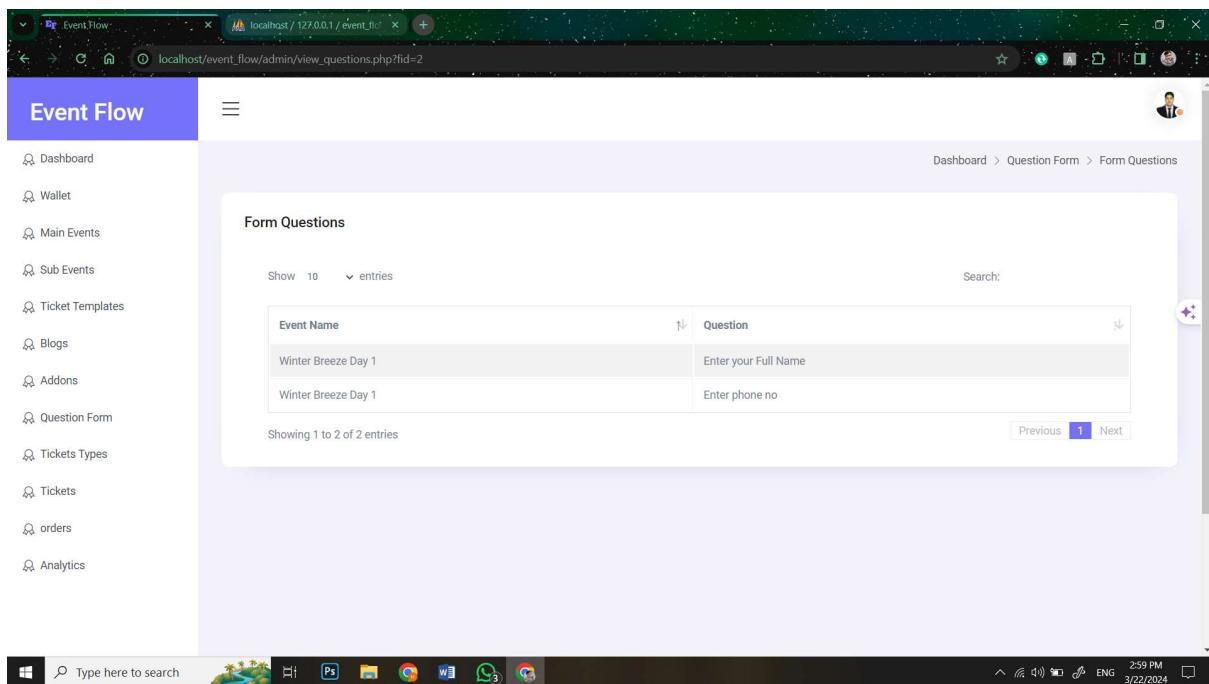


Figure68: Screen Shot 40: View Form Question

The screenshot shows a web browser window titled "Event Flow" with the URL "localhost/127.0.0.1/event_flow/admin/orders.php". The left sidebar has a purple header "Event Flow" and a list of navigation items: Dashboard, Wallet, Main Events, Sub Events, Ticket Templates (which is selected), Blogs, Addons, Question Form, Tickets Types, Tickets, orders (which is highlighted in blue), and Analytics. The main content area is titled "My tickets" and displays a table with two entries:

| Event name | ticket type | Ticket Quantity | Ticket Sold | Ticket Sold Price | Ticket Left |
|----------------------|-------------|-----------------|-------------|-------------------|-------------|
| Jalebi Morning Day 2 | VStand | 600 | 2 | 2000 | 598 |
| Jalebi Morning Day 2 | VVstand | 600 | 1 | 1200 | 599 |

Below the table, it says "Showing 1 to 2 of 2 entries". At the bottom right of the content area are buttons for "Previous", "1", and "Next". The top right corner of the main content area shows "Dashboard > My tickets". The bottom of the screen shows the Windows taskbar with various pinned icons.

Figure69: Screen Shot 41: View Orders

The screenshot shows a web browser window titled "Event Flow" with the URL "localhost/127.0.0.1/event_flow/admin/template.php". The left sidebar has a purple header "Event Flow" and a list of navigation items: Dashboard, Wallet, Main Events (which is selected), Sub Events, Ticket Templates (which is selected), Blogs, Addons, Question Form, Tickets Types, Tickets, orders, and Analytics. The main content area is titled "Create Main Events" and displays a table with five entries:

| Template No | Action |
|-------------|---------------------------------|
| 1 | <button>View Tempelate</button> |
| 2 | <button>View Tempelate</button> |
| 3 | <button>View Tempelate</button> |
| 4 | <button>View Tempelate</button> |
| 5 | <button>View Tempelate</button> |

Below the table, it says "Showing 1 to 5 of 5 entries". At the bottom right of the content area are buttons for "Previous", "1", and "Next". In the top right corner of the main content area, there is a "Create new" button. The top right corner of the main content area shows "Dashboard > Main Events". The bottom of the screen shows the Windows taskbar with various pinned icons.

Figure70: Screen Shot 42: View Ticket Templates



Figure71: Screen Shot 43: View Ticket Templates 1

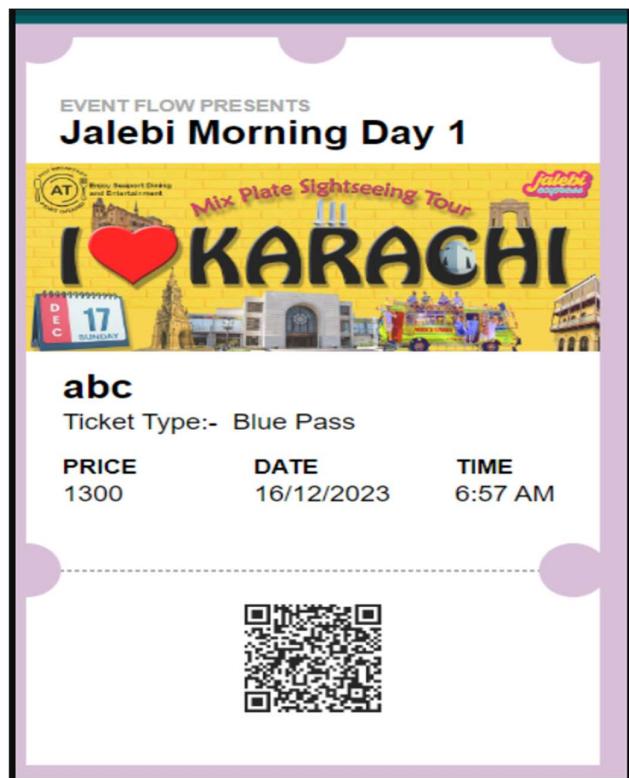


Figure72: Screen Shot 44: View Ticket Templates 2



Figure73: Screen Shot 45: View Ticket Templates 3

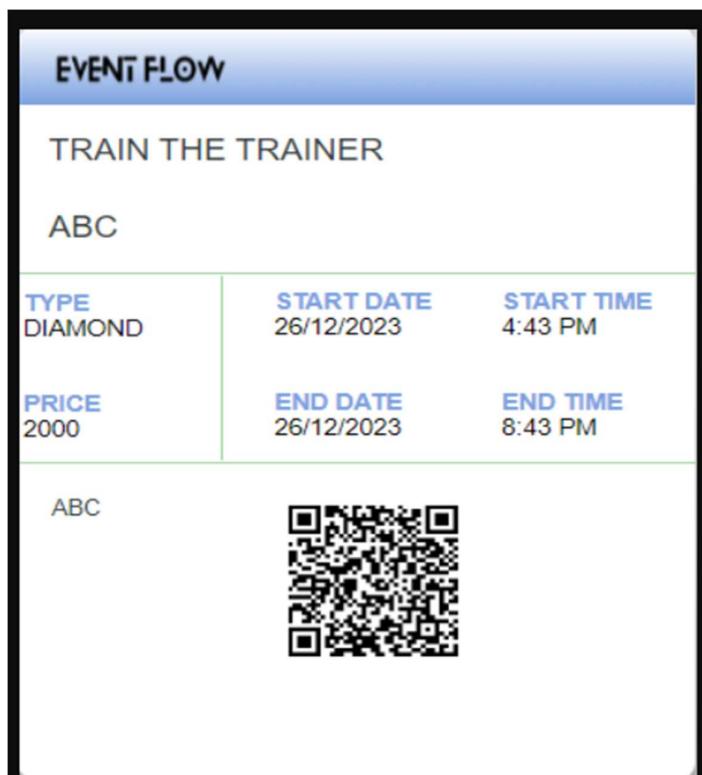


Figure74: Screen Shot 46: View Ticket Templates 4

Admin look of website:

The screenshot shows a Windows desktop environment with a browser window open to the 'Event Flow' admin dashboard at `localhost/event_flow/admin/users.php`. The left sidebar menu includes options like Dashboard, Wallet, Users, Categories, Main Events, Sub Events, and Analytics. The main content area is titled 'Organizer' and displays a table of users with columns for Username, Email, and Actions (Edit, View, Delete). The users listed are Ather Khaliq, Imran Khan, maliha, Osama, osamafayyaz, and Sheikh Osama. A 'Create new' button is visible in the top right corner of the table area.

Figure75: Screen Shot 47: View Users

The screenshot shows a Windows desktop environment with a browser window open to the 'Event Flow' admin dashboard at `localhost/event_flow/admin/users.php`. A modal dialog box titled 'Create organizer' is displayed in the center. It contains fields for 'Username' (with a user icon), 'Password' (with a lock icon), 'Email' (with an envelope icon), and 'Avatar' (with a camera icon and a 'Choose File' button). Below the dialog is a table of existing users: Ather Khaliq, Imran Khan, maliha, Osama, osamafayyaz, and Sheikh Osama. The table has columns for Username, Email, and Actions (Edit, View, Delete). A 'Close' and 'Add' button are located at the bottom right of the dialog.

Figure76: Screen Shot 48: Create Users

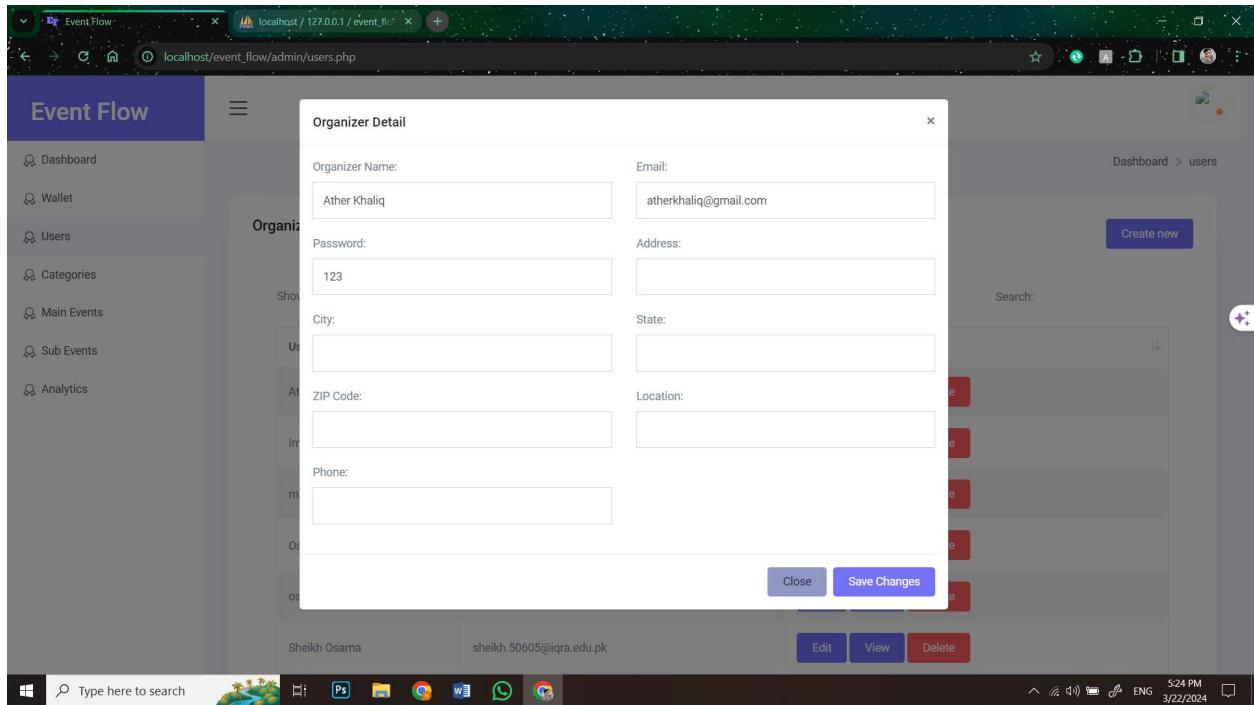


Figure77: Screen Shot 49: Edit Users

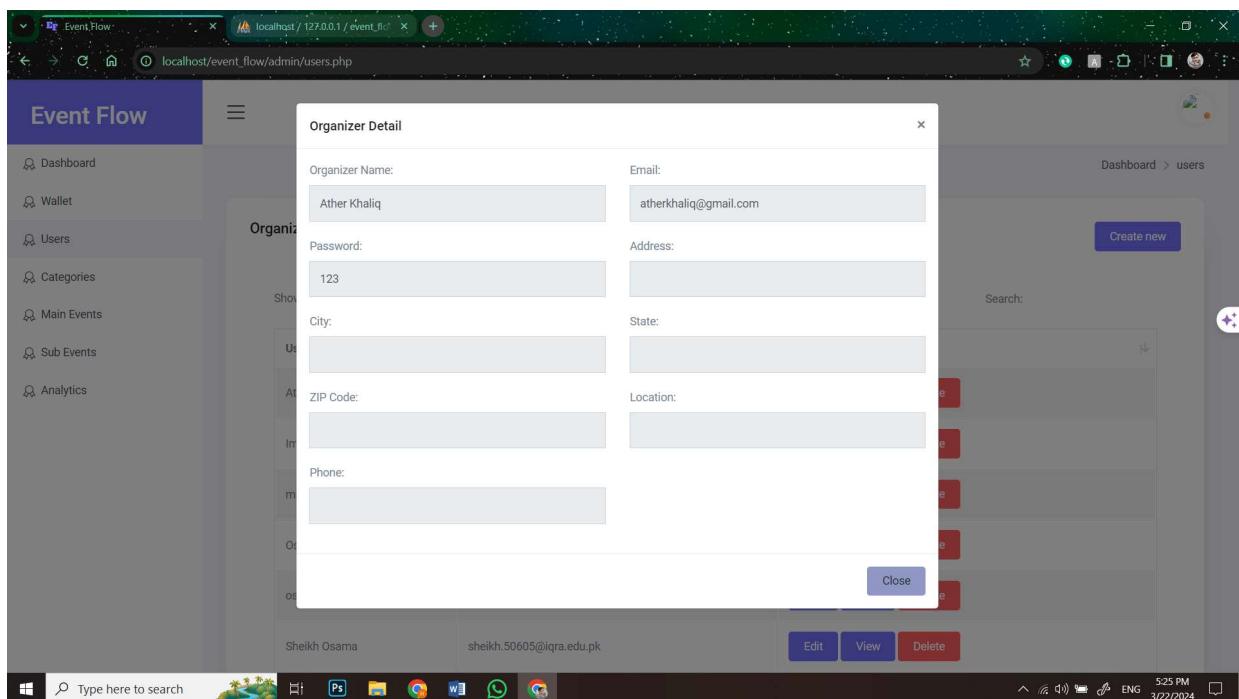


Figure78: Screen Shot 50: View Users

The screenshot shows the 'Event Flow' application interface. On the left is a sidebar with icons for Dashboard, Wallet, Users, Categories (which is selected), Main Events, Sub Events, and Analytics. The main content area is titled 'Event Categories' and shows a table of categories. The table has columns for 'Event Category Name' and 'Action'. The categories listed are Arts, Cinema, Cricket, Entertainment, Music, and Science. Each category row has a 'Delete' button in the 'Action' column. A 'Create Category' button is located at the top right of the table area. The status bar at the bottom shows the date and time as 3/22/2024 5:18 PM.

| Event Category Name | Action |
|---------------------|--------|
| Arts | Delete |
| Cinema | Delete |
| Cricket | Delete |
| Entertainment | Delete |
| Music | Delete |
| Science | Delete |

Figure79: Screen Shot 51: Edit Users

This screenshot shows the same 'Event Flow' application interface as Figure 79, but with a modal dialog box overlaid. The dialog is titled 'Add Category' and contains a single input field labeled 'Category Name:' with a placeholder 'Category Name...'. Below the input field are two buttons: 'Close' and 'Create New'. The background of the application is dimmed, and the 'Event Categories' table is visible but inactive. The status bar at the bottom shows the date and time as 3/22/2024 5:27 PM.

Figure80: Screen Shot 52: Edit Users

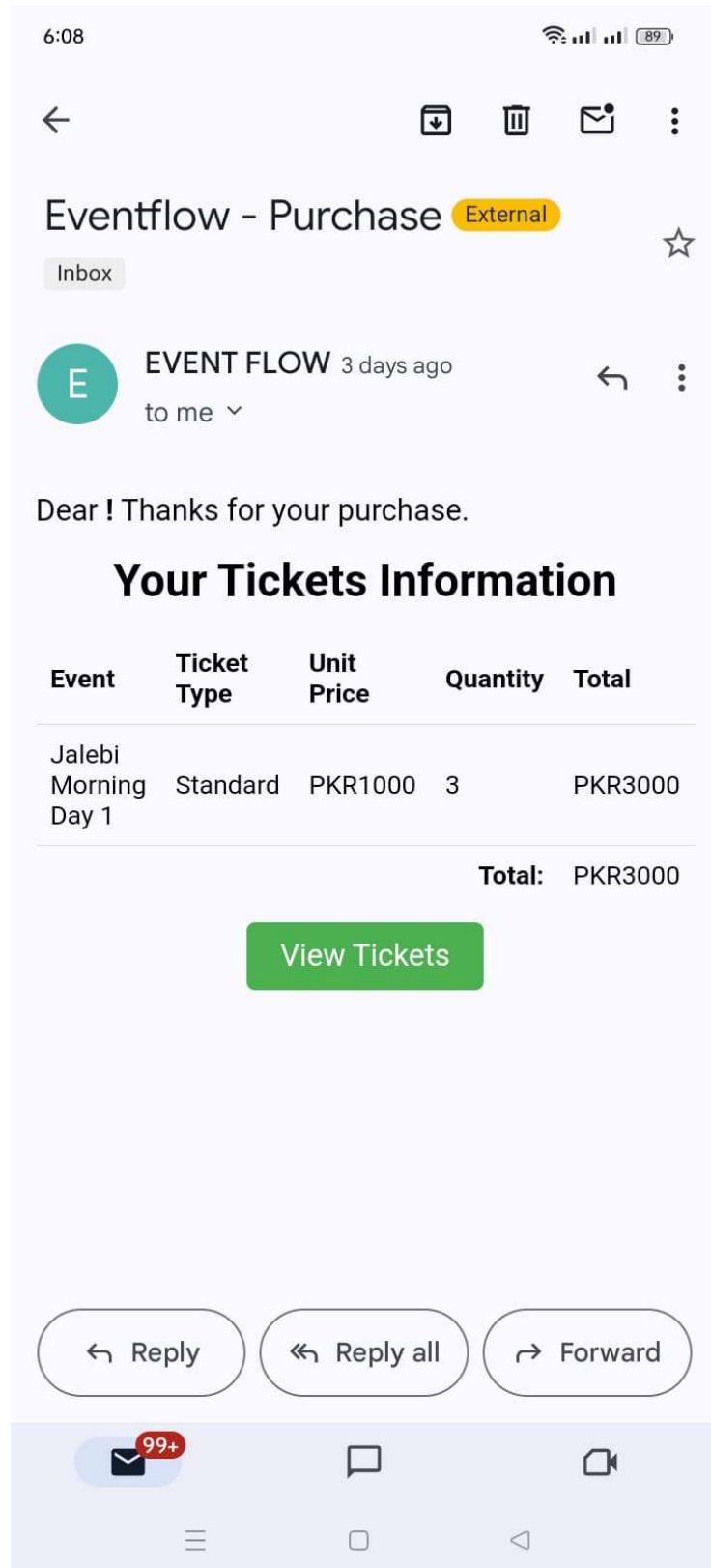


Figure81: Screen Shot 53: Confirmation Email

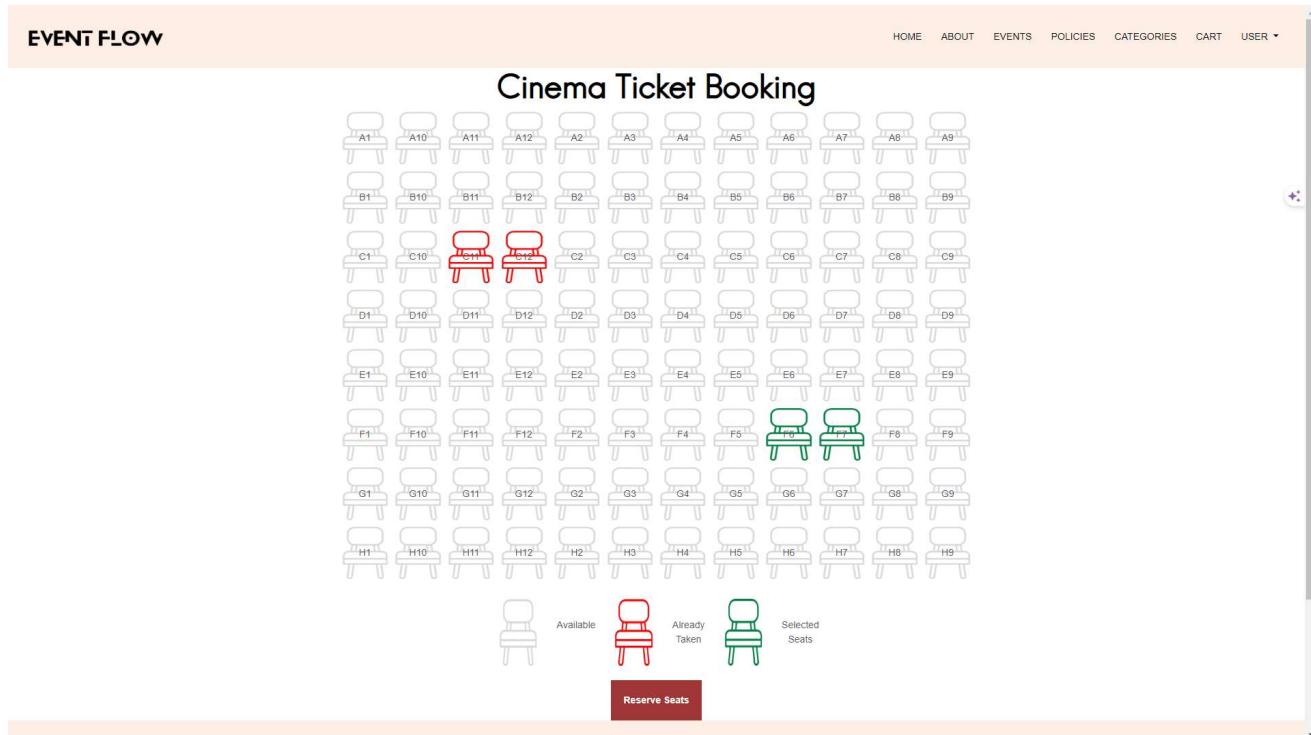


Figure82: Screen Shot 54: Seat Selection

5.5. Summary:

This Chapter consist of system prototype and development in which we explained about the prototype, frontend and backend design some database queries and some screenshot of our application. In this chapter we briefly give some explanation about the type of error happen on the system, source code of validation and etc. So from the next part some information flows are given to know how to system assignation works. The prototype design provides the clear view of working screens one by one. The prototype table consist of project title, observation date, screen name, screen number, and link of the screen. The table also informs about the overall description and functionality of the screen, while providing a specific description trough background, color scheme, video/audio connections and still images. The database queries are also provided to show how the system retrieve the data while functioning.

CHAPTER – 6

6.0. Introduction:

In this chapter, we have discussed about the test cases to determine whether the software is working the way it should and producing the expected results. We also test cases and usability test cases to test our software. This will help the readers to know about all the minor as well as the major working options of the software. After the completion of the implementation phase testing plays a vital role for making sure that the system works properly. Once the software was developed, the testing phase started. Each and every screen and button were tested according to the requirements and functionalities. Even though the Web application is complex it does use I treated functionality as one window is being used multiple times for different kinds of functionalities. Due to the re usability of our code the total test cases of the web application turned out to be 37. Each test case has the requirement reference, project name, web application name while providing all the details of the test case. In detail attributes of the test case, such as test case ID, Test case description, test steps, expected result, pass or fail status, preparation date, running date and end date at which it was tested.

6.1. Project Test Cases:

Attendee Test Cases:

| Requirement reference | ATT01 | Project Name | Event Flow |
|------------------------------|---|---------------------|------------|
| Test Case Id | ATT01_TC01 | Test Type | Functional |
| Test Case Description | Creating an Account | | |
| Test Steps | 1. Navigate to the signup page. 2. Enter valid details such as email, and password and click on attendee. 3. Click on the sign-up button. | | |
| Expected Results | User account is successfully created. | | |
| Actual Result | User account is successfully created. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 42: Attendee Test case 1

| Requirement reference | ATT01 | Project Name | Event Flow |
|------------------------------|---|---------------------|------------|
| Test Case Id | ATT01_TC02 | Test Type | Functional |
| Test Case Description | Signing In | | |
| Test Steps | 1. Navigate to the sign-in page. 2. Enter valid email and password. 3. Click on the sign-in button. | | |
| Expected Results | User sign-in successfully. | | |
| Actual Result | User sign-in successfully. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 43: Attendee Test case 2

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC03 | Test Type | Functional |
| Test Case Description | Viewing Events | | |
| Test Steps | 1. Sign in to the account. 2. Navigate to the events page. | | |
| Expected Results | Main Events are displayed with relevant details such as name, date. | | |
| Actual Result | Main Events are displayed with relevant details such as name, date. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 44: Attendee Test case 3

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC04 | Test Type | Functional |
| Test Case Description | Buying Tickets | | |
| Test Steps | 1. Log in to the attendee account. 2. Select an event. 3. Choose a sub-event. 4. Select ticket type and quantity. 5. Add tickets to the cart. 6. Proceed to checkout. | | |
| Expected Results | Tickets will be successfully added to the cart and purchased. Confirmation email is then received. | | |
| Actual Result | Tickets are successfully added to the cart and purchased. Confirmation email is received. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 21/3/2024 | | |
| Date Run | 21/3/2024 | | |
| Prepared By | Huzaifa Ahmed & Muhammad Sameer | | |
| Tested By | Huzaifa Ahmed & Muhammad Sameer | | |

Table 45: Attendee Test case 4

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC05 | Test Type | Functional |
| Test Case Description | Managing Tickets | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on the "My Tickets" page. | | |
| Expected Results | Purchased tickets are displayed. | | |
| Actual Result | Purchased tickets are displayed. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 46: Attendee Test case 5

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC06 | Test Type | Functional |
| Test Case Description | Refunding Tickets | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on the "My Tickets" page. 4. Click on view mytickets. 5. Select the ticket to refund. 6. Make Confirmation to refund. 7. Initiate the refund process. | | |
| Expected Results | Ticket will be successfully refunded, and funds are returned to the wallet. | | |
| Actual Result | Ticket are successfully refunded, and funds are returned to the wallet. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 47: Attendee Test case 6

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC07 | Test Type | Functional |
| Test Case Description | Top-up Wallet | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on the "Wallet" button. 4. Click on the "Topup" button. 5. Add funds using PayPal or credit/debit card enter amount and click on the "Topup" button. | | |
| Expected Results | Funds will be successfully added to the wallet. | | |
| Actual Result | Funds are successfully added to the wallet. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 21/3/2024 | | |
| Date Run | 21/3/2024 | | |
| Prepared By | Huzaifa Ahmed & Muhammad Sameer | | |
| Tested By | Huzaifa Ahmed & Muhammad Sameer | | |

Table 48: Attendee Test case 7

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC08 | Test Type | Functional |
| Test Case Description | Withdraw Amount | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on the "Wallet" button. 4. Click on the "Withdraw" button. 5. Withdraw funds to PayPal account enter amount and click on the "Withdraw button" button. | | |
| Expected Results | Funds will be successfully withdrawn from the wallet. | | |
| Actual Result | Funds are successfully withdrawn from the wallet. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama & Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 49: Attendee Test case 8

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC09 | Test Type | Functional |
| Test Case Description | Event Categories | | |
| Test Steps | 1. Log in to the attendee account. 2. Navigate to the categories page. | | |
| Expected Results | Events are categorized based on types such as music, sports, or conferences. | | |
| Actual Result | Events are categorized. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 50: Attendee Test case 9

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC010 | Test Type | Functional |
| Test Case Description | View Event Calendar | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on the "Event Calendar" button. | | |
| Expected Results | Events are displayed on a calendar view, allowing users to see events scheduled for specific dates. | | |
| Actual Result | Events are displayed on a calendar view. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama & Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 51: Attendee Test case 10

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC011 | Test Type | Functional |
| Test Case Description | Editing Account Information | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on the "User Profile" button. 4. Edit account information such as name, email, or password. 5. Click on save Changes button | | |
| Expected Results | Account information will be successfully updated. | | |
| Actual Result | Account information is successfully updated. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama & Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 52: Attendee Test case 11

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC012 | Test Type | Functional |
| Test Case Description | Sharing Events | | |
| Test Steps | 1. Log in to the attendee account. 2. Select a Main event. 3. Select its Sub event. 4. Use the social media sharing buttons to share the event on social platforms. | | |
| Expected Results | Event will be successfully shared on selected social media platforms | | |
| Actual Result | Event is not successfully shared on selected social media platforms | | |
| Pass/Fail | Fail | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Huzaifa Ahmed | | |
| Tested By | Huzaifa Ahmed | | |

Table 53: Attendee Test case 12

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC013 | Test Type | Functional |
| Test Case Description | View Tickets | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on the "My Tickets" page. 4. Click on view mytickets. 5. Click on View Tickets | | |
| Expected Results | Tickets will be viewed. | | |
| Actual Result | Tickets are viewed. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 54: Attendee Test case 13

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC014 | Test Type | UI/UX |
| Test Case Description | Navigation Ease | | |
| Test Steps | 1. Log in to the attendee account. 2. Navigate through different pages using menu or navigation links. | | |
| Expected Results | Navigation is intuitive and easy to understand, with clear labels and consistent placement. | | |
| Actual Result | Navigation is intuitive and easy . | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Huzaifa Ahmed | | |
| Tested By | Huzaifa Ahmed | | |

Table 55: Attendee Test case 14

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC015 | Test Type | UI/UX |
| Test Case Description | Event Discovery | | |
| Test Steps | 1. Log in to the attendee account. 2. Browse through events and categories. | | |
| Expected Results | Events will be presented in an attractive and organized manner, making it easy for users to discover relevant events. | | |
| Actual Result | Events are presented in an attractive and organized manner, making it easy for users to discover relevant events. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 22/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama & Huzaifa Ahmed | | |
| Tested By | Sheikh Mohammad Osama | | |

Table 56: Attendee Test case 15

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC016 | Test Type | UI/UX |
| Test Case Description | Confirmation Email | | |
| Test Steps | 1. Log in to the attendee account. 2. Complete the ticket purchase process. | | |
| Expected Results | Confirmation email will be received after purchase contains all necessary details and is well-formatted. | | |
| Actual Result | Confirmation email is received after purchase contains all necessary details and is well-formatted. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 22/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama & Huzaifa Ahmed | | |
| Tested By | Sheikh Mohammad Osama | | |

Table 57: Attendee Test case 16

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ATT01 | Project Name | Event Flow |
| Test Case Id | ATT01_TC017 | Test Type | UI/UX |
| Test Case Description | Ticket Viewing | | |
| Test Steps | 1. Log in to the attendee account. 2. View purchased tickets in the "My Tickets" section. | | |
| Expected Results | Tickets shall displayed clearly with all relevant information, such as event name, date, and QR code for easy check-in. | | |
| Actual Result | Tickets are displayed clearly with all relevant information, such as event name, date, and QR code. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 21/3/2024 | | |
| Date Run | 21/3/2024 | | |
| Prepared By | Huzaifa Ahmed & Muhammad Sameer | | |
| Tested By | Huzaifa Ahmed & Muhammad Sameer | | |

Table 58: Attendee Test case 17

Organizer Test Cases:

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC01 | Test Type | Functional |
| Test Case Description | Creating an Account | | |
| Test Steps | 1. Navigate to the signup page. 2. Enter valid details such as email, and password and click on organizer. 3. Click on the sign-up button. | | |
| Expected Results | User account is successfully created. | | |
| Actual Result | User account is successfully created. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 59: Organizer Test case 1

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC02 | Test Type | Functional |
| Test Case Description | Signing In | | |
| Test Steps | 1. Navigate to the sign-in page. 2. Enter valid email and password. 3. Click on the sign-in button. | | |
| Expected Results | User is successfully logged in to their account. | | |
| Actual Result | User is successfully logged in to their account. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz & Huzaifa Ahmed | | |
| Tested By | Muhammad Sameer | | |

Table 60: Organizer Test case 2

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC03 | Test Type | Functional |
| Test Case Description | Creating Main Event | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Main Events tab. 5. Click Create new. 6. Fill in event details. 7. Click create new. | | |
| Expected Results | Main event will be successfully created with provided details. | | |
| Actual Result | Main event is successfully created with provided details. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Huzaifa Ahmed | | |
| Tested By | Huzaifa Ahmed | | |

Table 61: Organizer Test case 3

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC04 | Test Type | Functional |
| Test Case Description | Creating Sub-Events | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Sub Events tab. 5. Click Create new. 6. Fill in event details. 7. Click create new. | | |
| Expected Results | Sub event will be successfully created with provided details. | | |
| Actual Result | Sub event is successfully created with provided details. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Huzaifa Ahmed | | |
| Tested By | Huzaifa Ahmed | | |

Table 62: Organizer Test case 4

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC05 | Test Type | Functional |
| Test Case Description | Generating Ticket Types | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Ticket Types tab. 5. Click Create new. 6. Fill in event details. 7. Click create new. | | |
| Expected Results | Tickets Types will be successfully generated for each sub event. | | |
| Actual Result | Tickets Types are successfully generated for each sub event. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 63: Organizer Test case 5

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC06 | Test Type | Functional |
| Test Case Description | Generating Tickets | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Sub events tab. 5. Click on Generate Tickets new. 6. Fill in event details. 7. Click generate ticket. | | |
| Expected Results | Tickets will be successfully generated for each sub event. | | |
| Actual Result | Tickets are successfully generated for each sub event. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 64: Organizer Test case 6

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC07 | Test Type | Functional |
| Test Case Description | Creating Blogs | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Sub events tab. 5. Click on Create Block of that sub event. 6. Give blog details. 7. Click create new. | | |
| Expected Results | Blog post will be successfully created and published. | | |
| Actual Result | Blog post is successfully created and published. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 65: Organizer Test case 7

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC08 | Test Type | Functional |
| Test Case Description | Adding Add-ons | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Sub events tab. 5. Click on generate Ticket of that sub event. 6. Add Addons. | | |
| Expected Results | Add-ons will be successfully added for purchase along with tickets. | | |
| Actual Result | Add-ons are successfully added for purchase along with tickets. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 66: Organizer Test case 8

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC09 | Test Type | Functional |
| Test Case Description | Creating/ Editing/Deleting Add-ons | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Addons 5. Click on Create new than fill information then click create new to add addon. 6. Click delete Addons to delete. 6. Click edit Addons to edit. | | |
| Expected Results | Add-ons will be successfully created/edited & deleted and available for purchase along with tickets. | | |
| Actual Result | Add-ons are successfully created/edited & deleted and available for purchase along with tickets. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Huzaifa Ahmed | | |
| Tested By | Huzaifa Ahmed | | |

Table 67: Organizer Test case 9

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC10 | Test Type | Functional |
| Test Case Description | Viewing Orders | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Orders | | |
| Expected Results | Orders will be displayed showing the number of tickets sold and remaining. | | |
| Actual Result | Orders are displayed showing the number of tickets sold and remaining. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Huzaifa Ahmed & Muhammad Sameer | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 68: Organizer Test case 10

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC11 | Test Type | Functional |
| Test Case Description | Editing/Deleting Main Event | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Main Events tab. 5. Click Edit/Delete main event. | | |
| Expected Results | Main event will be successfully edited or deleted. | | |
| Actual Result | Main event is successfully edited and deleted | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Huzaifa Ahmed | | |

Table 69: Organizer Test case 11

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC12 | Test Type | Functional |
| Test Case Description | Editing/Deleting Sub Event | | |
| Test Steps | 1. Log in to the organizer account. 2. Go to the "User". 3. Click on "Dashboard" button. 4. Click on Sub Events tab. 5. Click Edit/Delete main event. | | |
| Expected Results | Sub event will be successfully edited or deleted. | | |
| Actual Result | Sub event is successfully edited and deleted | | |
| Pass/Fail | Pass | | |
| Date Prepared | 20/3/2024 | | |
| Date Run | 22/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama Fayyaz | | |
| Tested By | Sheikh Mohammad Osama Fayyaz | | |

Table 70: Organizer Test case 12

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC013 | Test Type | Functional |
| Test Case Description | Sharing Events | | |
| Test Steps | 1. Log in to the attendee account. 2. Select a Main event. 3. Select its Sub event. 4. Use the social media sharing buttons to share the event on social platforms. | | |
| Expected Results | Event will be successfully shared on selected social media platforms | | |
| Actual Result | Event is not successfully shared on selected social media platforms | | |
| Pass/Fail | Fail | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Huzaifa Ahmed | | |
| Tested By | Huzaifa Ahmed | | |

Table 71: Organizer Test case 13

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC014 | Test Type | Functional |
| Test Case Description | Editing Account Information | | |
| Test Steps | 1. Log in to the organizer account. 2. Click on Users on Header Tab 3. Click on the "Dashboard" button. 4. Click on your picture Icon the Click profile. 4. Edit account information such as name, email, or password. 5. Click on save Changes button | | |
| Expected Results | Account information will be successfully updated. | | |
| Actual Result | Account information is successfully updated. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 72: Organizer Test case 14

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC15 | Test Type | Functional |
| Test Case Description | Top-up Wallet | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on Dashboard button. 4. Click on the "Wallet" button. 5. Click on the "Withdraw" button. 6. Add funds using PayPal or credit/debit card enter amount and click on the "Topup" button. | | |
| Expected Results | Funds will be successfully added to the wallet. | | |
| Actual Result | Funds are successfully added to the wallet. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 21/3/2024 | | |
| Date Run | 21/3/2024 | | |
| Prepared By | Huzaifa Ahmed & Muhammad Sameer | | |
| Tested By | Huzaifa Ahmed & Muhammad Sameer | | |

Table 73: Organizer Test case 15

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC16 | Test Type | Functional |
| Test Case Description | Withdraw Amount | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on Dashboard button. 4. Click on the "Wallet" button. 5. Click on the "Withdraw" button. 6. Withdraw funds to PayPal account enter amount and click on the "Withdraw button" button. | | |
| Expected Results | Funds will be successfully withdrawn from the wallet. | | |
| Actual Result | Funds are successfully withdrawn from the wallet. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama & Huzaifa Ahmed | | |
| Tested By | Huzaifa Ahmed | | |
| Requirement reference | ORG01 | Project Name | Event Flow |

Table 74: Organizer Test case 16

| | | | |
|------------------------------|---|------------------|-------|
| Test Case Id | ORG01_TC17 | Test Type | UI/UX |
| Test Case Description | Event Creation | | |
| Test Steps | 1. Log in to the attendee account. 2. Click on Users on Header Tab 3. Click on Dashboard button. 4. Create a new main event. | | |
| Expected Results | Event creation form will be well-designed with fields clearly labeled and relevant options provided. | | |
| Actual Result | Event creation form is well-designed with fields clearly labeled and relevant options provided. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Huzaifa Ahmed & Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 75: Organizer Test case 17

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC18 | Test Type | UI/UX |
| Test Case Description | Ticket Type Generation | | |
| Test Steps | 1. Log in to the organizer account. 2. Define ticket types for an event. | | |
| Expected Results | Ticket type creation interface will be intuitive, allowing organizers to set prices, capacities, and discounts easily. | | |
| Actual Result | Ticket type creation interface is intuitive, allowing organizers to set prices, capacities, and discounts easily. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama & Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 76: Organizer Test case 18

| | | | |
|------------------------------|--|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC19 | Test Type | UI/UX |
| Test Case Description | Event Management | | |
| Test Steps | 1. Log in to the organizer account. 2. Manage Event in Main Events & Sub Events | | |
| Expected Results | Ticket type creation interface will be intuitive, allowing organizers to set prices, capacities, and discounts easily. | | |
| Actual Result | Organized event management interface with options to edit, delete, or view event details clearly displayed | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Sheikh Mohammad Osama & Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 77: Organizer Test case 19

| | | | |
|------------------------------|---|---------------------|------------|
| Requirement reference | ORG01 | Project Name | Event Flow |
| Test Case Id | ORG01_TC20 | Test Type | UI/UX |
| Test Case Description | Blog Creation | | |
| Test Steps | 1. Log in to the organizer account. 2. Create a new blog post for an event | | |
| Expected Results | Blog creation interface is user-friendly, with options to add text, images, and formatting easily accessible. | | |
| Actual Result | Blog creation interface is user-friendly, with options to add text, images, and formatting easily accessible. | | |
| Pass/Fail | Pass | | |
| Date Prepared | 23/3/2024 | | |
| Date Run | 23/3/2024 | | |
| Prepared By | Muhammad Sameer | | |
| Tested By | Muhammad Sameer | | |

Table 78: Organizer Test case 20

6.2. Summary:

We test our software to get our expected results of our software or whether a system under test satisfies requirements or works correctly. After test cases, we get satisfied results the usability test case.

CHAPTER – 7

7.0. Introduction:

This chapter is like a summary of all the important stuff we did for our Online Ticketing Event Management System. We will talk about what we did, what problems we faced, and what cool stuff we could do in the future.

Our Online Ticketing Event Management System is like a super helpful tool for organizing events and selling tickets online. It makes things easier for both the people throwing the event and the people going to it. However, you know, making something like this is not always easy. We had to work hard and face some challenges along the way.

But hey, it's not all bad news! There are also some exciting things we can do to make our system even better in the future. So, in this chapter, we're going to talk about all of that – the good stuff, the tricky stuff, and the things we can do to make our system even cooler.

7.1. System Limitations and Challenges:

Okay, so let us get real for a moment. Every project has its limits and problems. Our Online Ticketing Event Management System is no different. Here are some things we found that might make it a bit tricky for people to use:

7.1.1. Internet Connection required*: You need the internet to use our system. That means if you do not have internet, you cannot use it.

7.1.2. Tech Know-How*: Not everyone is a tech genius, right? Some people might find it a bit tricky to figure out how to use our system if they are not used to using computers or smartphones.

7.1.3. Parental Controls*: We do not have any built-in rules for parents to control what their kids can do on the system. Therefore, if parents want to keep an eye on what their kids are up to, they might find it a bit tricky.

7.1.4. Camera Challenges*: One big challenge we faced was getting the system to recognize things using the camera. It was tough, but we managed to figure it out!

7.2. Future Work:

All right, so we have talked about the problems. However, what about the good stuff? Well, there is plenty of that too! In the future, we have some cool ideas for making our Online Ticketing Event Management System even better:

7.2.1. More Features: We want to add more cool features to our system. Maybe we could make it even easier to buy tickets or give organizers more options for customizing their events.

7.2.2. Better User Experience: We are always looking for ways to make our system easier and more fun to use. Therefore, in the future, we might make some changes to make things even smoother for everyone.

7.2.3. Improved Security: We take security seriously. In the future, we will keep working to make sure our system is as safe and secure as possible for everyone to use.

7.2.4. Recurring Billing: Adding recurring billing options could be super helpful for organizers who host events regularly. It would make it easier for them to manage payments and subscriptions.

7.2.5. QR Code Scanner Machine: Introducing a QR code scanner machine could streamline the check-in process at events. Attendees could simply scan their tickets upon arrival, making entry quicker and more efficient.

7.2.6. Seat Mapping Tools: Providing seat-mapping tools would allow organizers to plan seating arrangements more effectively. Attendees could select their seats during the ticket purchase process, enhancing their experience.

7.3. Conclusion:

To sum it all up, our Online Ticketing Event Management System is a handy tool for organizing events and selling tickets. Sure, we had some challenges along the way, but we managed to figure things out. Moreover, the best part? There is still so much we can do to make our system even better in the future. We are excited to keep working on it and see where it takes us!

REFERENCES

1. W3Schools. (n.d.). W3Schools Online Web Tutorials. Retrieved from <https://www.w3schools.com/>
2. mith, Emily. (2020). "The Rise of Online Ticketing Systems: A Comprehensive Analysis." Event Planning Journal, 15(2), 45-60.
3. Free Code Camp. (n.d.). Learn to Code — For Free. Retrieved from <https://www.freecodecamp.org/>
4. CSS-Tricks. (n.d.). CSS-Tricks. Retrieved from <https://css-tricks.com/>
5. PHP Programming for Beginners: Key Programming Concepts" by Joseph Joyner.
6. Ticket Tailor. (n.d.). Flexible Ticket Pricing and Capacity Management. Retrieved from <https://www.tickettailor.com/>
7. Constant Contact. (n.d.). Customized Email Invitations for Events. Retrieved from <https://www.constantcontact.com/>
8. Attendify. (n.d.). Personalized Name Badges with QR Code Integration. Retrieved from <https://www.attendify.com/>
9. RSVPify. (n.d.). Seat Assignment Management for Events of Any Size. Retrieved from <https://rsvpify.com/>
10. Eventzilla. (n.d.). Interactive Calendar View for Event Scheduling and Ticket Booking. Retrieved from <https://www.eventzilla.net/>
11. Eventbrite. (n.d.). Purchase Protection & Refunds Handling. Retrieved from <https://www.eventbrite.com/features/refunds/>

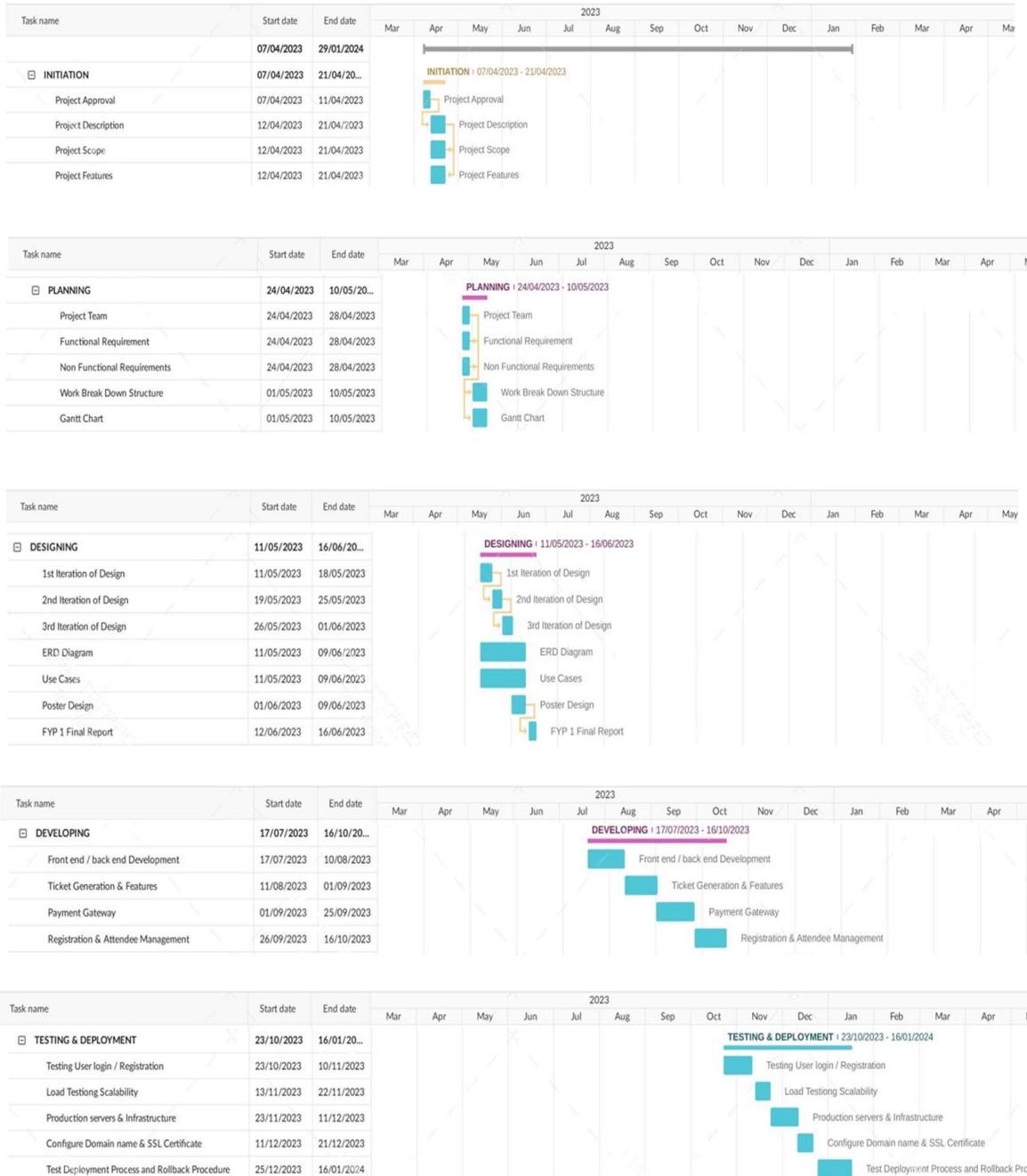
APPENDIX

Business Canvas:

BUSINESS MODEL CANVAS

| | | | | |
|---|---|--|--|--|
| Designed For: EVENT FLOW | Designed By: EVENT FLOW TEAM | Date: 3/22/2024 | Version: 1.0 | |
| PROBLEMS | SOLUTIONS | UNIQUE VALUE PROPOSITION | UNFAIR ADVANTAGE | |
| <p>Manual Ticket Sales Traditional event management processes involve manual ticket sales, leading to inefficiencies and errors.</p> <p>Limited Promotional Reach Organizers struggle to reach a wider audience for their events, limiting promotional efforts.</p> <p>Difficulty in Managing Attendee Information Organizers find it challenging to manage attendee information efficiently, leading to confusion and delays.</p> | <p>Streamlined Ticketing Process Event Flow offers an automated ticketing system, simplifying the ticket purchasing process for both organizers and attendees.</p> <p>Efficient Attendee Management Event Flow facilitates easy attendee registration, confirmation, and management, ensuring a smooth event experience for both organizers and attendees.</p> <p>Streamlined ticket customization Effortlessly customize ticket types, prices, and capacities to suit your event needs.</p> | <p>Event Flow offers a seamless online ticketing and event management solution, simplifying the entire event lifecycle for organizers and attendees alike, ensuring a memorable and hassle-free experience.</p> | <p>Our proprietary algorithms for personalized event promotion and attendee management cannot be easily replicated by competitors, giving Event Flow a distinct advantage in the market.</p> | |
| COST STRUCTURE | | REVENUE STREAMS | | |
| Development and Maintenance Operations Infrastructure Costs Time Cost of Student Electricity Workspace Cost | | Ticket Sales Commissions or Transaction Fees Subscription Fees for Premium Features (in future) Advertising Revenue Licensing or Partnership Fees | | |
| CUSTOMER SEGMENTS <ul style="list-style-type: none"> Event organizers ranging from small businesses to large corporations Attendees seeking convenient and secure ticket purchasing experiences for various events Event planning professionals looking for efficient event management solutions | | | | |

Detailed Gantt chart:



Software Manual:

Introduction

The software manual provides guidance on using the Event Flow Online Ticketing Event Management System. This manual is divided into two parts: the Attendee User Manual and the Organizer User Manual. The Attendee User Manual helps attendees navigate the system to purchase tickets and manage their event experience, while the Organizer User Manual assists organizers in setting up and managing events effectively.

- Both the users should have High Speed Internet Connectivity.
- Open Chrome/Google Browser to access the Website.
- Search Event Flow

Attendee User Manual:

1. Creating an Account:

To use the Event Flow platform, you need to create an account. Visit the signup page and provide necessary information such as your name, email address, and password.

2. Signing In:

-Once you have created an account, you can sign in using your email address and password on the sign-in page.

3. Viewing Events:

After signing in, you can browse through the list of events available on the platform. You can see details like event name, date.

4. Viewing Calendars:

If you prefer viewing events in a calendar format, you can access the calendar feature to see events on specific dates.

5. Buying Tickets:

To purchase tickets for an event, click on the event you are interested in. Then, select the sub-event you want to attend, choose the ticket type and quantity, and add them to your cart.

6. Checkout:

Once you have added all desired tickets to your cart, proceed to the checkout page. Here, you can review your order and provide payment information.

7. Ticket Confirmation:

After successful checkout, a confirmation email containing your tickets information will be sent to your registered email address.

8. Managing Tickets:

You can view all purchased tickets on the "My Tickets" page. From here, you can also initiate a refund for any ticket if needed.

9. Refund Policies:

Detailed refund policies are available on the policies page. Make sure to review them before requesting a refund.

10. Wallet Management:

If you wish to add money to your wallet for future purchases, you can do so using PayPal, credit, or debit cards. You can also withdraw funds from your wallet to your PayPal account

11. Event Categories:

To explore events based on categories such as music, sports, or conferences, you can visit the categories page.

Organizer User Manual:

1. Creating an Account:

- Organizers need to sign up for an account to use the Event Flow platform. Visit the signup page and provide necessary information.

2. Signing In:

After creating an account, organizers can sign in using their credentials on the sign-in page.

3. Creating Main Event:

To start organizing an event, go to the Dashboard "Main events" page and create a new main event. Fill in all the required details such as event name, date, and description.

4. Creating Sub-Events:

For complex events with multiple parts, you can create sub-events under the main event. Provide details for each sub-event, such as location and schedule.

5. Generating Ticket Types:

Define different ticket types for your event on the ticket types page. Set prices, capacities, and any discounts available..

6. Generating Tickets:

After defining ticket types, you can generate tickets for each type. This will create unique tickets for attendees to purchase.

7. Creating Blogs:

Share updates and information about your event by creating blog posts from the sub event page.

8. Adding Add-ons:

Offer additional services or products related to your event by creating add-ons. These can be purchased along with tickets.

9. Ticket Templates:

Customize the design of your event tickets using templates available on the ticket templates page.

10. Wallet Management:

Manage funds in your wallet by adding or withdrawing money using PayPal, credit, or debit cards.

11. Viewing Orders:

Keep track of ticket sales and remaining tickets by viewing orders on the orders page.

12. Event Management:

Organizers can edit or delete main events or sub-events as needed. All events organized by the user can be viewed on the "My Events" page.

This manual provides step-by-step instructions for both attendees and organizers on how to use the Event Flow platform efficiently. If you encounter any issues or have further questions, please refer to the support section for assistance or email us at eventflow786@gmail.com