Department of Computer Science and Technology,

NMAM Institute of Technology, Nitte, Karnataka, India

PROJECT REPORT

ON

Event Ticket Management System

Submitted to

NMAM Institute of Technology, Nitte

(Off-Campus Centre, Nitte Deemed to be University, Nitte – 576110, Karnataka, India)

On partial fulfilment of the requirements for the award of the

Degree of Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

By

Akshatha B Shenoy

NNM23CS019

Ankita Kamath K

NNM23CS029

Under the guidance of

Ms. Jayapadmini Kanchan

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the **mini project** entitled **"Event Ticket Management Report"** submitted by **Akshatha B Shenoy** (NNM23CS019) and **Ankita Kamath K**(NNM23CS029) of II year B-Tech, a Bonafide students of NMAM Institute of Technology, Nitte, has done mini project during January to April 2025 fulfilling the partial requirement for the award of degree of bachelor of technology in computer science and engineering at NMAM Institute of Technology, Nitte.

Ms. Jayapadmini Kanchan	**Dr. Jyothi Shetty**
Assistant Professor	Professor & Head
Dept. of CSE, NMAMIT	Dept. of CSE, NMAMIT

ABSTRACT

The **Event Ticket Management System** is a digital solution designed to streamline the process of organizing events by automating ticket booking, and verification. The system aims to eliminate the inefficiencies of manual ticketing methods and provide a user-friendly platform. This project includes features such as online ticket registration & real-time availability tracking. **EventTix** is a PHP-based web application integrated with a MySQL database. It allows users to register, log in, and easily browse available events. Attendees can view ticket details such as price and type, select tickets, and proceed to a secure payment gateway to confirm their bookings. The platform supports real-time ticket availability tracking, ensuring that users can purchase tickets based on current event status. This project serves as a practical application of software development principles, integrating frontend and backend technologies to address real-world challenges in event coordination and ticketing. With its focus on transparency, convenience, and ease of use, **EventTix** aims to transform the event ticketing experience, making it more organized, accessible, and scalable for various types of events, from conferences to workshops and cultural fests.

ACKNOWLEDGMENT

We dedicate this page to acknowledging and thanking those responsible for shaping the project. Without their guidance and help, the experience while constructing the dissertation would not have been so smooth and efficient.

I take this opportunity to express my deep sense of gratitude to our project guide **Ms. Jayapadmini Kanchan**, Assistant Professor Gd-II, Department of Computer Science and Engineering, NMAMIT, Nitte for her whole-hearted support and valuable guidance she has given us during the Project.

Our sincere thanks to our Principal, **Dr. Niranjan N Chiplunkar** and Vice Principal **Dr. I. R. Mithanthaya** for their inspiration and support while carrying out the project work.

Lastly, we acknowledge our families and friends for their understanding and support during this journey.

We extend our gratitude to **NMAM INSTITUTE OF TECHNOLOGY, NITTE** for giving us this opportunity. This project would not have been possible without the collective efforts of everyone involved.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Purpose

The **Event Ticket Management System** is designed to streamline the process of discovering, booking, and managing event tickets for both attendees and organizers. Its purpose is to provide a user-friendly platform where attendees can easily find and purchase tickets, while event organizers can efficiently manage event details, venues, ticket categories, and payments. With scalability in mind, the system is suitable for a wide range of events, from small gatherings to large festivals. By incorporating secure payment gateways and modern technologies, the system ensures reliable, secure transactions and an overall smooth booking experience.

1.2 Scope

The **Event Ticket Management System** aims to provide a streamlined, digital platform for event discovery, ticket booking, and management.

1. **User Features**:
    - Secure registration, login, and ticket booking.
    - Event discovery through dates, and venues.
    - View and manage booking history.
2. **Organizer Features**:
    - Event creation, updates, and management.
    - Real-time ticket availability and category management.
    - Payment tracking and reporting.

3. **Scalability & Flexibility**:

    - Suitable for events of all sizes, from small workshops to large festivals.
    - Easy to scale and integrate additional features.

The system is designed to simplify event ticketing, enhance user experience, and provide event organizers with a robust management platform.

1.3 Brief Overview

In the era of online ticket booking, it is difficult to keep track of all the different ticket booking websites. Purchasing tickets for respective events can be time-consuming and confusing, especially when there are multiple events happening at the same time. Many of these events may have their own platform for selling tickets. For instance, sports fans often must buy tickets from different platforms for different sporting events such as football, and cricket. The Event Ticket Management System is developed using PHP for backend development, with MySQL for database management. The user interface is designed with HTML, CSS, and JavaScript to ensure a responsive and user-friendly experience.

Upon logging in, users are directed to a personalized dashboard where they can:

- View available events and their details
- Book tickets for upcoming events
- Track their bookings and past events

Users can browse through different categories of events, select tickets based on availability, and proceed with secure payment options. Once a ticket is successfully booked, the user receives a confirmation notification, and the booking is recorded in the system.

The project shows a practical application of web development skills, focusing on secure payment integration, database design, session management, and intuitive UI/UX development. It demonstrates how digital platforms can streamline event booking and management, providing convenience, transparency, and efficiency for both attendees and event organizers.

Chapter 2

Requirements Specification

2.1 Hardware Specification

To ensure smooth development, testing, and execution of the website, the following minimum hardware requirements are recommended:

- Device: Desktop / Laptop
- Processor: Quad-core 2.0 GHz or higher
- RAM: 4 GB or higher (8 GB recommended for smooth admin operations)
- Storage: Minimum 100 GB (for storing attendee data, logs, and reports)
- Internet: High-speed internet connection (5 Mbps or higher)

2.2 Software Specification

- OS: Windows/Linux
- Web Server: Apache
- Programming Language: PHP 7.4+
- Database: MySQL 5.6+
- Frontend: HTML, CSS, JavaScript, Bootstrap
- Development Tools: IDE (VS Code), Database Management Tools (MySQL Workbench)
- Browser: Chrome/Firefox

Chapter 3

System Design

3.1 ER Diagram

The **ER diagram** models a ticket booking system with five main entities: Venue, Event, Ticket, Attendee, and Payment, and their relationships. A Venue holds multiple Events (1:N), and an Event sells multiple Tickets (1:N). Attendees buy multiple Tickets, and Tickets can be bought by multiple Attendees (M:N). Attendees make multiple Payments (1:N), with each Payment linked to one Attendee and one Ticket (PAYS: 1:N). This system manages event organization, ticket sales, attendee registration, and payment tracking.

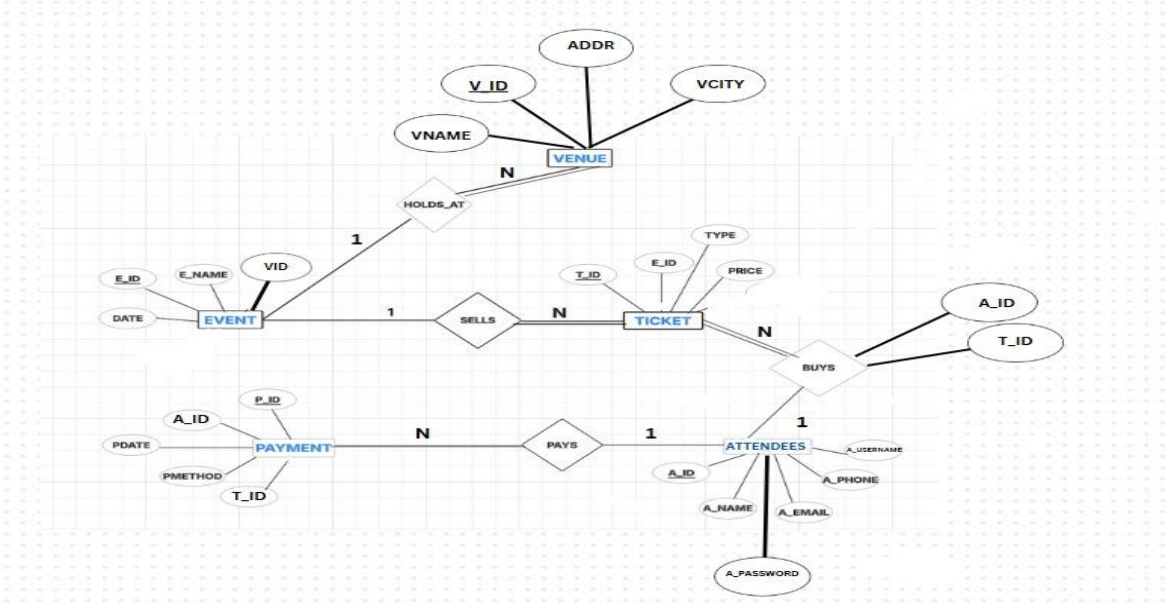

Fig 3.1: ER Diagram

3.2 Mapping from ER diagram to Schema Diagram

Step 1: Mapping of Regular (Strong) Entity Types

In the event ticket management system ER Diagram, the strong entities are:

- **Event**
- **Venue**
- **Ticket**
- **Payment**
- **Attendees**

These are considered strong entities because they each have a primary key attribute (E_ID, V_ID, T_ID, P_ID, A_ID) that uniquely identifies each instance of the entity & do not depend on any other entity for existence.

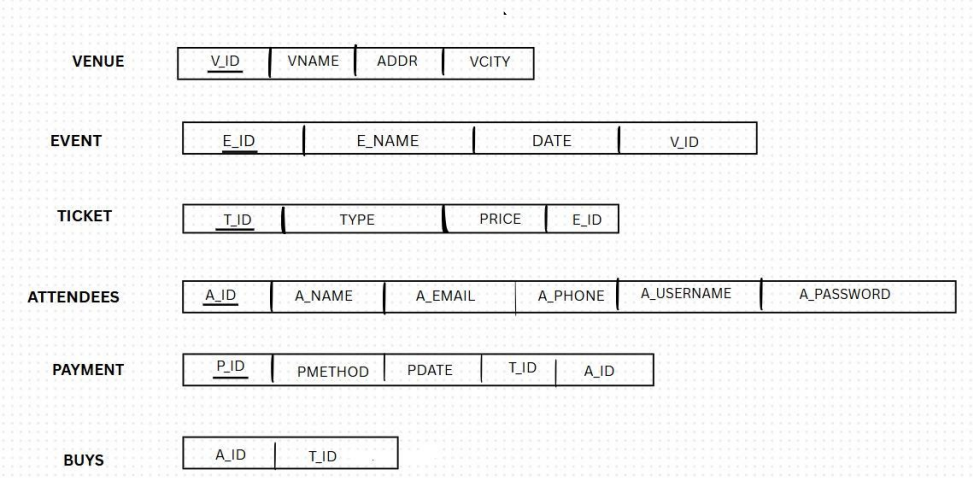

Fig 3.2.1: Mapping of Strong Entity types

Step 2: Mapping of Weak Entity types

In the ER Diagram, there are no weak entities. A weak entity is one that does not have a primary key of its own and depends on a strong entity for identification-usually connected by a total participation and an identifying relationship. In this diagram: All entities (Event, Venue, Ticket, Payment, Attendees) have their own unique primary keys (E_ID, V_ID, T_ID, P_ID, A_ID), so none of them qualify as weak entities.

Step 3: Mapping of 1:1 Relationship types

In the ER diagram of the Event Ticket Management System, there are no 1:1 (one-to-one) relationships between any of the entities. A 1:1 relationship means that one instance of an entity is associated with exactly one instance of another entity, and vice versa.

Step 4: Mapping of 1: N Relationship types

1. **VENUE - EVENT**
    - Relationship: HOLDS_AT
    - Type: 1: N
    - A venue can hold multiple events, but each event is held at exactly one venue.
2. **EVENT - TICKET**
    - Relationship: SELLS
    - Type: 1: N
    - A single event can have many tickets, but each ticket is associated with only one event.
3. **ATTENDEES - TICKET**
    - Relationship: BUYS
    - Type: 1: N
    - An attendee can buy many tickets, but each ticket is bought by one attendee.
4. **ATTENDEES - PAYMENT**
    - Relationship: PAYS
    - Type: 1: N
    - An attendee can make multiple payments, but each payment is made by exactly one attendee.

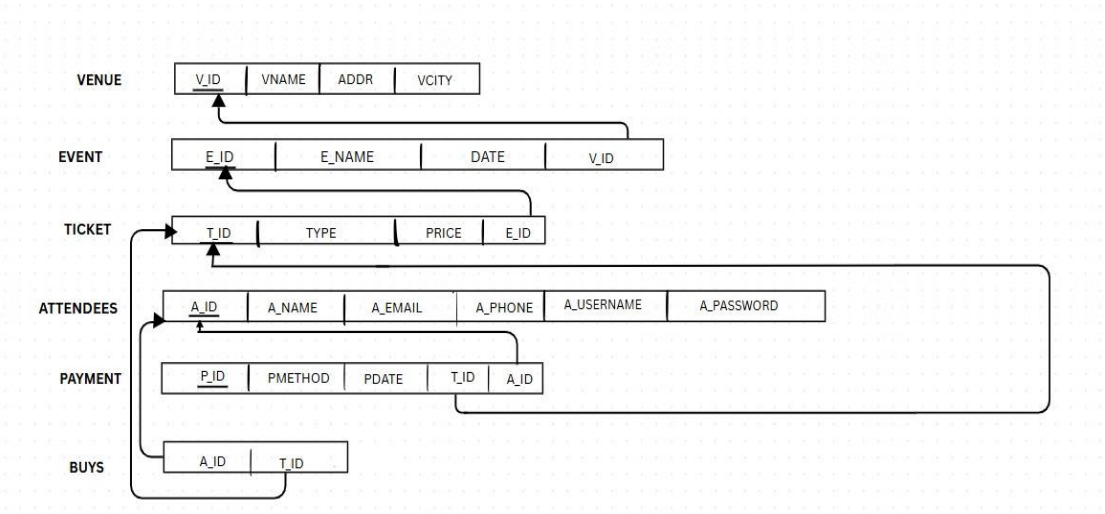

Fig 3.2.2: Mapping of 1: N Relational types

Step 5: Mapping of M: N Relation types

In the Event Ticket Management System, there are no many-to-many (M: N) relationships between any of the entities. A many-to-many relationship occurs when multiple instances of one entity are associated with multiple instances of another. However, in this ER diagram, all relationships are either one-to-many (1: N) or many-to-one (N:1). For example:

- One VENUE holds many EVENTS - a 1: N relationship
- One EVENT sells many TICKETS - another 1: N relationship
- One ATTENDEE buys many TICKETS - again a 1: N relationship
- One ATTENDEES make many PAYMENT - again a 1: N relationship

Since none of the entities share a mutual many-to-many connection, it is confirmed that no M: N relationship exists in this ER diagram.

Step 6: Mapping of Multivalued attributes

In the provided ER diagram of the Event Ticket System, there are no multivalued attributes explicitly shown. A multivalued attribute is typically represented using a double oval in ER diagrams, indicating that an entity can have multiple values for that attribute. Attributes Overview from the Diagram:

1. **Venue**
   Attributes: V_ID, VNAME, ADDR, VCITY
2. **Event**
   Attributes: E_ID, E_NAME, DATE, V_ID
3. **Ticket**
   Attributes: T_ID, TYPE, PRICE, E_ID
4. **Attendees**
   Attributes: A_ID, A_EMAIL, A_PHONE, A_NAME, A_PASSWORD, A_USERNAME
5. **Payment**
   Attributes: P_ID, PMETHOD, PDATE, A_ID, T_ID
6. **BUYS**
   Attributes: A_ID, T_ID

None of these attributes are shown with a double oval, which is the conventional symbol for multivalued attributes.

Step 7: Mapping of N-Ary Relationship types

In ER modeling, an N-Ary relationship is a relationship that involves three or more entities simultaneously. These are represented as diamonds connected to three or more rectangles (entities). Each relationship in the diagram connects only two entities. No diamond is connected to three or more entities. Therefore, all relationships are binary - no N-Ary relationship exists.

3.3 Assumption

- Each event is held at one venue, but a venue can host multiple events.
- Each event can offer multiple tickets, but each ticket is for one event only.
- Each attendee can purchase multiple tickets, but each ticket is purchased by one attendee.
- Each payment is made by one attendee and corresponds to one ticket.
- Attendees must register with a username and password to make purchases.

- Each ticket has a fixed price with no discounts or dynamic pricing.

3.4 Schema Diagram

A schema diagram is a visual representation of the structure of a relational database. It shows:

- Tables (which come from entities and relationships)
- Attributes of each table (columns)
- Primary keys and foreign keys
- Relationships between tables

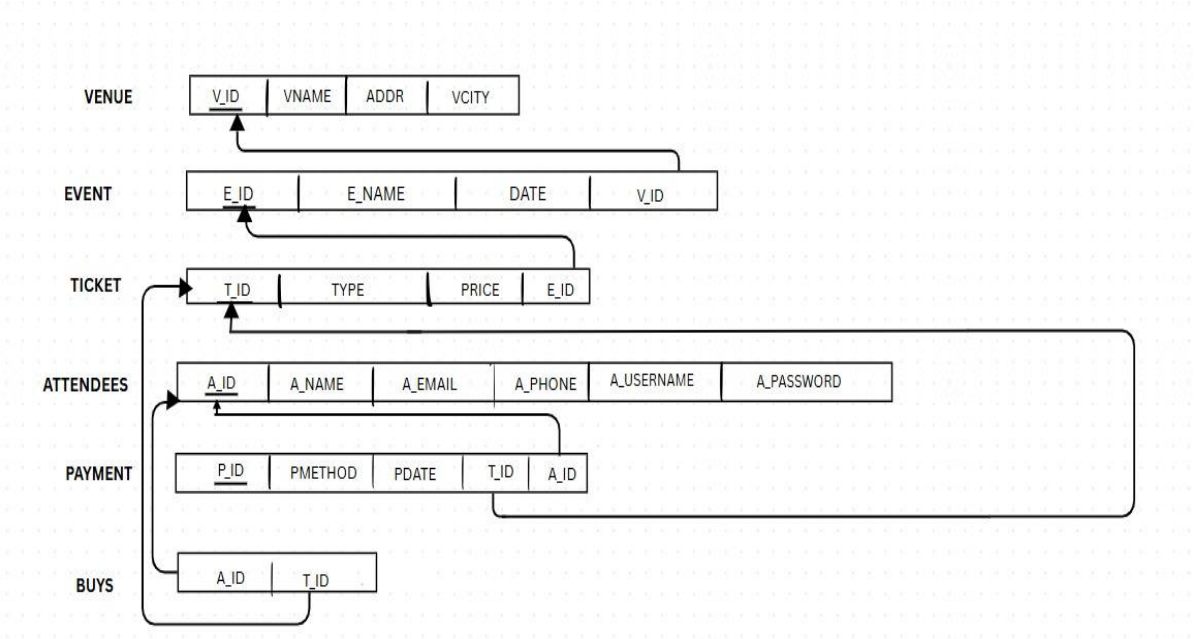

Fig 3.4: Schema Diagram

Chapter 4

Implementation

4.1 Pseudocode used

User Sign-Up

IF user submits sign-up form, THEN
  IF email or username exists in database THEN
    Show error message
  ELSE
    Insert user into ATTENDEES table
    Redirect to login page

User Login

IF login form is submitted THEN
  IF credentials match an existing user, THEN
    Start session
    Redirect to homepage/dashboard
  ELSE
    Show invalid login error

Event Ticket Booking

IF user selects a ticket and proceeds to book **THEN**
  Store selected ticket ID in session
  Redirect to payment page

Payment Processing

IF payment form is submitted **THEN**
  IF entered password matches user password **THEN**
    Insert payment into PAYMENT table
    Insert booking into BUYS table
    Redirect to booking history with success message

ELSE

Show invalid password error

4.2 Tables Used

Table Name	What It Stores
VENUE	Venue details – name, address, and city
EVENT	Event info – name, date, and associated venue
TICKET	Ticket details – type, price, and linked event
ATTENDEES	Attendee details – name, contact info, username, password
PAYMENT	Payment details – date, method, and attendee reference
BUYS	Purchase records – which attendee bought which ticket

Chapter 5

Results and Discussion

The Event Ticket Management System was successfully developed and tested, demonstrating robust performance across all modules. Key features include:

- **User Registration & Login**: Attendees can create accounts, log in, and manage bookings via a personalized dashboard.
- **Event & Ticket Browsing**: Real-time listings allow users to view and book tickets, while organizers manage events and ticket categories.
- **Secure Booking & Payment**: Users can securely select tickets and proceed with payment; all records are stored accurately.
- **Search & Filter**: Events can be filtered by date, venue, or type for ease of access.
- **Dashboard Display**: Real-time dashboards are available for both users and admins.
- **Database Integrity**: Relational consistency is ensured through proper foreign key constraints.

The Fig 5.1 is the **cover page** where there is log in or sign up to manage their tickets and event preferences. Clean design with quick access to **About** and **Contact** pages ensures a smooth user experience.

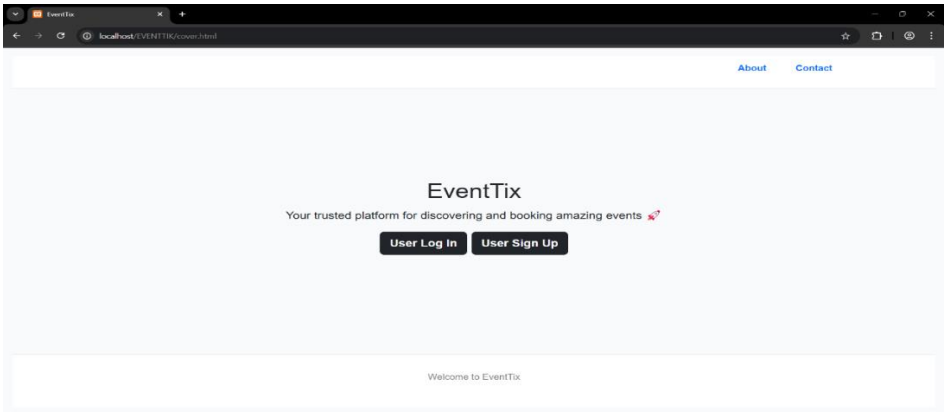

Fig 5.1: Cover page

The Fig 5.2 is the **SIGN-UP** page where The **Create an account** page allows users to register with essential details like name, email, phone, and password. It includes terms agreement and offers sign-up via Google or Facebook for convenience. A clean layout ensures a smooth and user-friendly onboarding experience.

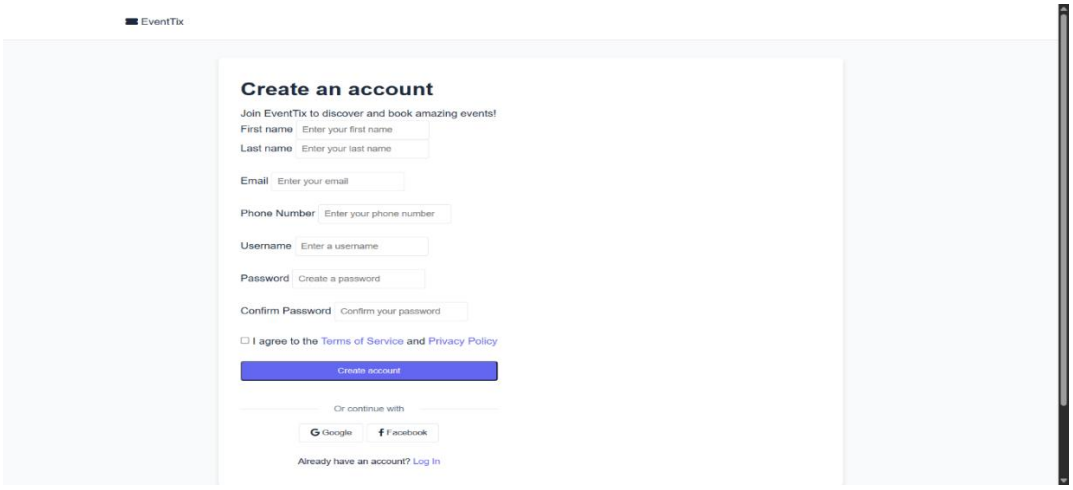

Fig 5.2: User Sign-Up Page

The **Fig 5.3** is the **Log In** page enables returning users to securely access their accounts using their credentials.
Features like “Remember me” and “Forgot password?” enhance usability and accessibility.
Its minimal and intuitive design ensures a quick and smooth login experience.

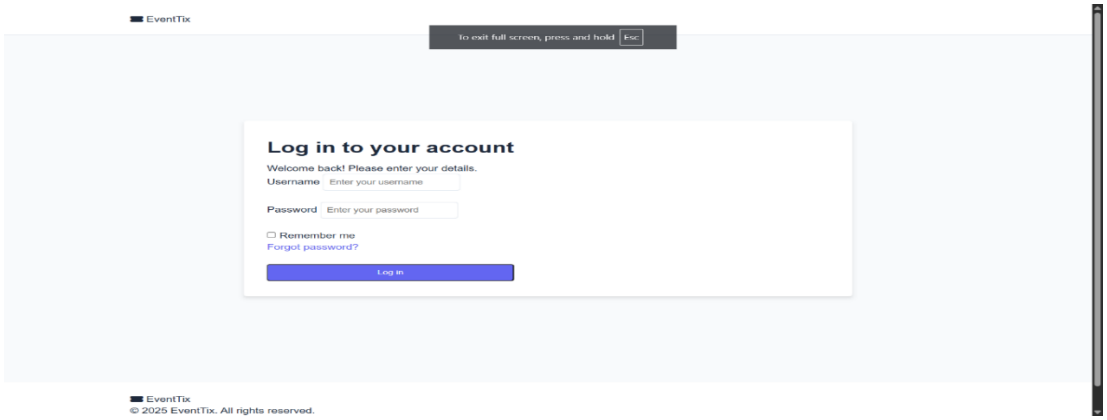

Fig 5.3: User Login Page

The Fig 5.4 is the **User Dashboard** welcomes logged-in users with a clean and friendly interface. It offers quick access to explore upcoming events or view past bookings. With intuitive navigation and a logout option, it ensures a smooth user journey.

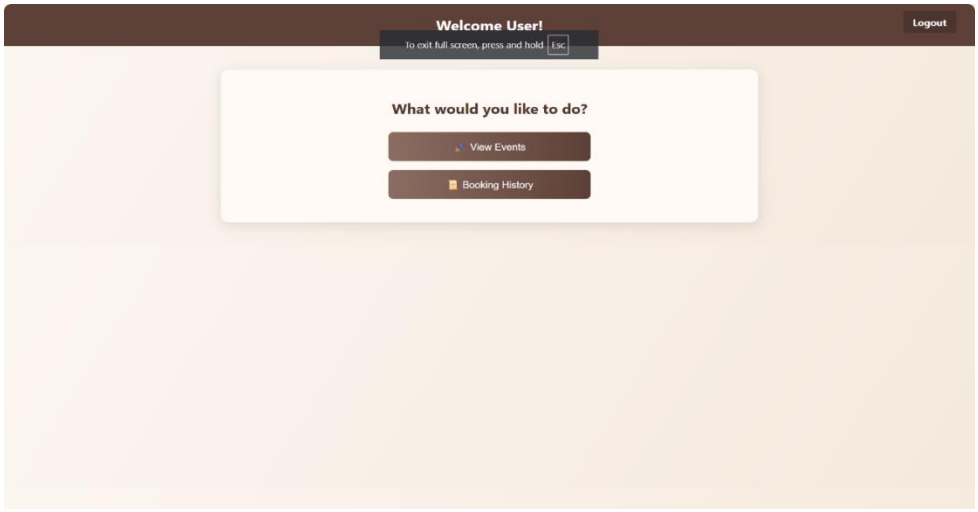

Fig 5.4: Dashboard

The Fig 5.5 is the **Available Events** page lists all upcoming events with details like date, venue, and ticket types. Users can browse through events, choose their preferred ticket category, and proceed to booking. The design is scrollable and user-friendly, enabling smooth ticket selection.

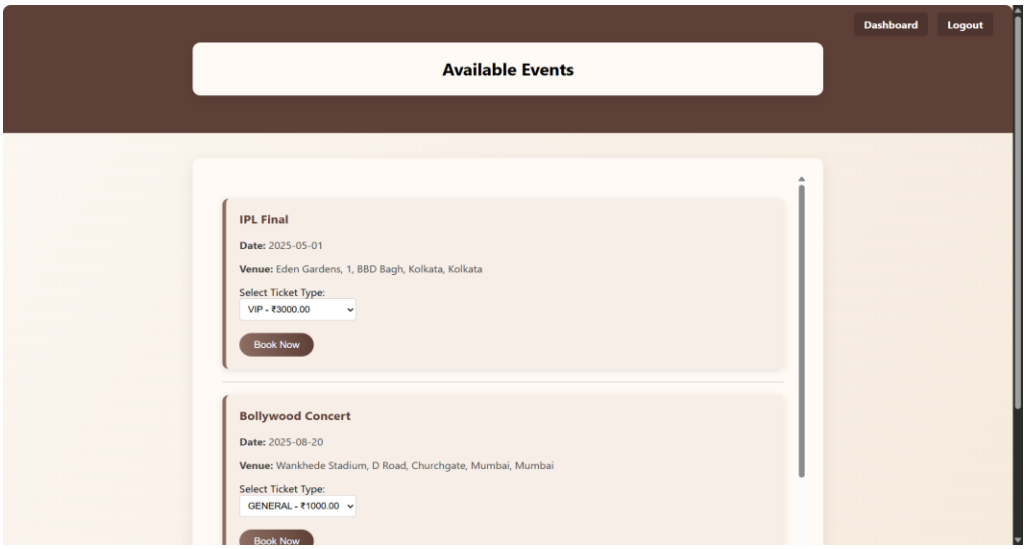

Fig 5.5: View Events Page

The Fig 5.6 is **the Confirm Payment page** is the final step in the booking process where users verify their ticket type and amount, choose a payment method (e.g., Credit Card, UPI), and confirm their identity via password. Clicking "Pay Now" completes the transaction securely and efficiently.

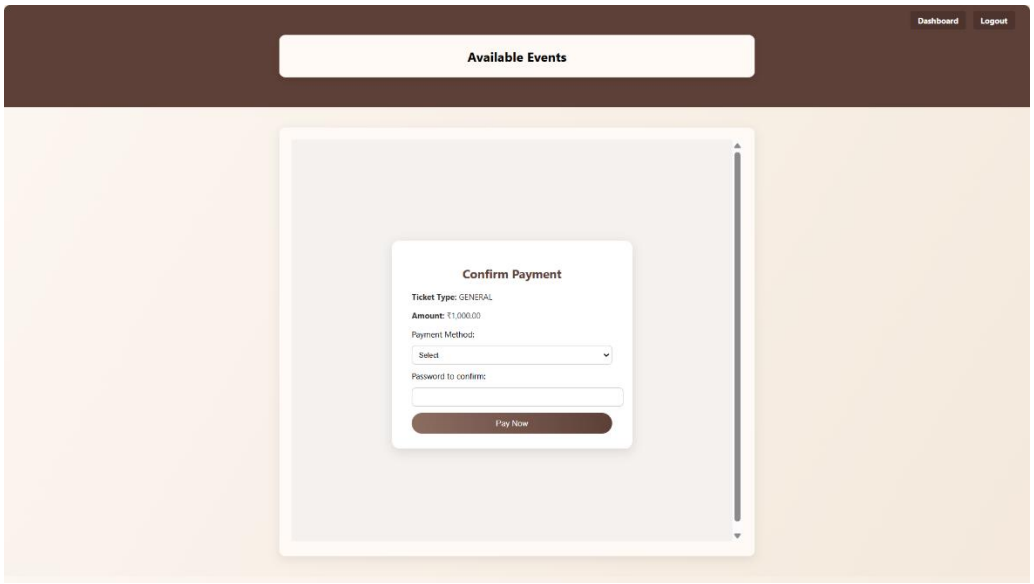

Fig 5.6: Payment Confirmation Page

The Fig 5.7 is the **Booking History page** displays details of past event bookings including event name, date, venue, ticket type, and payment information. Users can view their transactions and cancel bookings if needed, providing transparency and easy management of reservations.

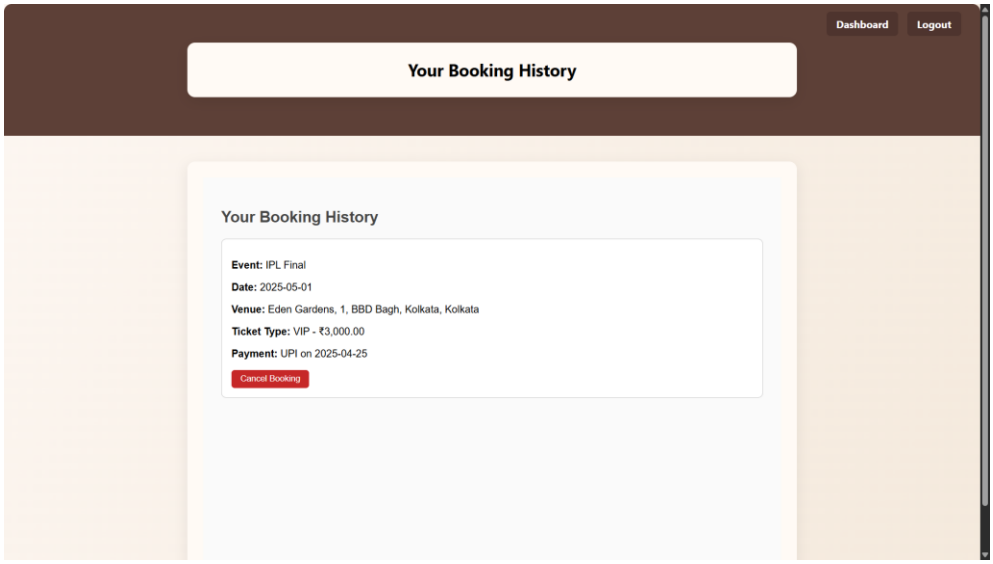

Fig 5.7: View Booking History

Chapter 6

Conclusion and Future Work

EventTix successfully delivers a digital platform that simplifies the end-to-end process of event ticket booking and management. By allowing users to register, browse events, book tickets, and make payments securely, the system improves both user experience and operational efficiency for event organizers. By integrating dynamic queries and structured tables, it enables:

- Efficient management of event and venue data
- Real-time tracking of ticket availability and sales
- Easy access to attendee and payment records

The project showcases the application of web development concepts using technologies like PHP, MySQL, HTML, CSS, and JavaScript, resulting in a fully functional and responsive ticketing system.

Some future work may include:

1. **QR Code Ticketing**
    - Generate QR codes for booked tickets to allow easy scanning and entry verification at event venues.
    - Prevent duplicate ticket usage through real-time check-in.
2. **Mobile App Development**
    - Develop a companion mobile app for Android/iOS to improve accessibility and user engagement.
    - Enable push notifications for event updates, ticket confirmations, and reminders.
3. **Feedback & Ratings**
    - Add a feature for attendees to rate and review events, helping organizers gather insights.
4. **Multi-language Support**
    - Add language options for a broader user base, especially in multilingual regions or institutions.

References

[1] Database systems Models, Languages, Design and Application Programming, Ramez El- masri and Shamkant B. Navathe, 7th Edition, Pearson.

[2] MySQL. MySQL Official Website. 2025. Available at: https://www.mysql.com/. Ac- cessed: 2025-04-16.

[3] W3Schools. PHP MySQL Introduction. 2025. Available at: https://www.w3schools.com/php/. Accessed: 2025-04-16.

[4] Microsoft. Visual Studio Code. 2025. Available at https://code.visualstudio.com/. Ac- cessed: 2025-04-16.