



# Ticket Booking Website

DBMS Project Report

March 29, 2025

Rakesh Divvela - CS23BTECH11017

Sai Deepak Sana - CS23BTECH11055

Arjun Konidala - CS23BTECH11026

Sukesh Samagalla - AI23BTECH11024

# Contents

1. Introduction
2. Front End
3. Back End
4. DBMS
5. Project Overview
6. Future Enhancements

## **Contributors:**

Rakesh Divvela  
Sai Deepak Sana  
Arjun Konidala  
Sukesh Samagalla

# Introduction

## Welcome to TIXORA

TIXORA is a dynamic platform that connects event enthusiasts with a wide array of Events, shows, festivals, and activities. It streamlines the discovery and booking process, ensuring a smooth user experience. Our curated listings cover everything from concerts and comedy nights to workshops and sports events. We empower organizers with tools to easily promote and manage their events.

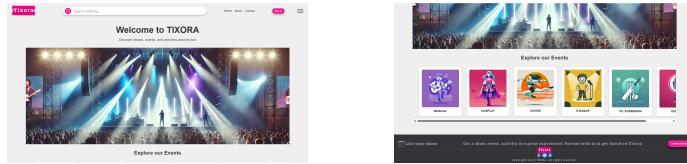
In our website development process, we create the front-end using HTML, CSS, and JavaScript. For the back-end, we utilize Node.js and React.js, while MySQL serves as our database management system. This combination ensures that our website is both dynamic and responsive. The front-end handles everything the user interacts with, creating an intuitive and engaging experience. Meanwhile, the back-end manages data processing and business logic, ensuring smooth performance and reliability. Overall, these technologies work together to deliver a seamless, efficient, and user-friendly platform.

The project is organized within a main folder named Tixora, which serves as the root directory for all essential files and resources. Inside this folder, there is an images directory that contains all the images used during the website's design and development. Additionally, a css folder is present, which includes the style.css file responsible for styling the website. Apart from these, the Tixora folder also holds various HTML and JavaScript files that structure and add functionality to the website. Moreover, it includes SQL files to manage the database operations efficiently. This well-structured organization ensures easy navigation and maintainability of the project.

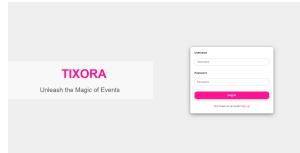
# FrontEnd

The front-end of the TIXORA website is developed using **HTML**, **CSS**, and **JavaScript**. The project is structured with a modular approach to ensure scalability and maintainability. In total, we have created around 11 HTML files for different purposes:

- **index.html** — The home page that serves as the gateway to the website.



- **login.html** — The login page where users can sign in to book tickets.



- **signup.html** — The registration page for new users.



- **user.html** — A profile page where users can edit their personal details.
- **categoryevents.html** — A page that displays events under a specific category when a user selects one from the home page.
- **eventdetails.html** — The page that provides comprehensive details about a selected event, helping users review information before booking.
- **book.html** — The booking page where users can finalize their ticket purchases.
- **transaction.html** — A page displaying transaction details for recently booked tickets.
- **yourtickets.html** — A page where users can view all their current and past bookings, with an option to cancel tickets if allowed.



- **about.html** — An informational page that describes the website and its mission.
- **contact.html** — A page designed for event hosts or users to contact us with any queries.

In addition, the main **Tixora** folder includes:

- An **images** folder containing all the graphical assets used during the website design.
- A **css** folder that holds the **style.css** file, where the CSS code is organized in a modular fashion for easy maintenance for future changes.

# Branding and Styling

The visual identity of the TIXORA website is driven by a carefully chosen color palette and distinctive logos. Our primary color is **#ff1493**, which is used extensively throughout the site to create a vibrant and energetic feel. In addition, the background color **#efefef** is applied in various sections to maintain a clean, modern look while ensuring content readability.



Two key logos reinforce our brand identity:

- **Mainlogo.png** — This logo represents the core brand identity of TIXORA.



- **TicketLogo.png** — This alternative logo is used in contexts where a secondary or complementary brand mark is needed.

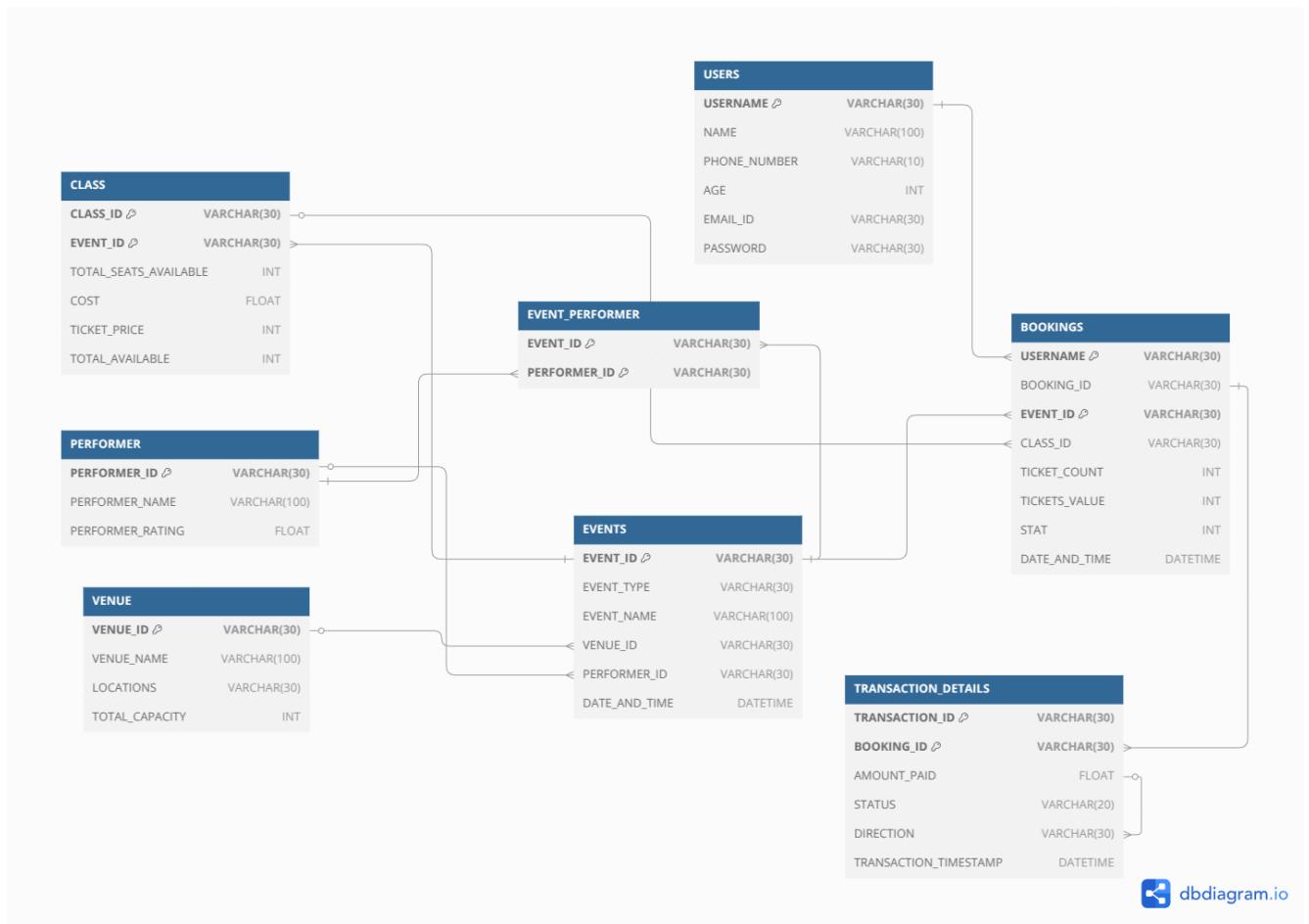


Together, these design elements create a cohesive and recognizable brand image that enhances user experience and reflects the dynamic nature of our platform.

# DBMS

## Introduction

Functional dependencies (FDs) play a crucial role in ensuring data integrity and normalization within a database. They help determine how attributes depend on one another, guiding the design of efficient and reliable schemas. This report analyzes the Tixora database model by detailing the entities, relationship sets, and their corresponding functional dependencies. A normalization analysis is also provided to demonstrate that the design adheres to Third Normal Form (3NF).



## Entities and Their Details

- **USERS:** Stores information about individuals who have created an account on our event ticket booking system. It includes personal details necessary for user identification and enables users to book tickets.
- **EVENTS:** Holds data related to the events available on our platform. It contains details such as dates, descriptions, and other event-specific information.
- **VENUE:** Records the addresses and additional details of the locations where events take place, including attributes like maximum capacity. The VENUE entity is independent of EVENTS, meaning a venue can exist without being directly linked to a specific event.
- **PERFORMERS:** Contains information about various performers who may participate in events, even if they are not currently associated with any event.
- **CLASS:** Defines categories or classes associated with an event. Each class has a unique `class_id` and includes restrictions such as the maximum number of seats that can be booked.
- **TRANSACTION\_DETAILS:** Maintains records of all transactions related to ticket bookings. It is designed to ensure accuracy and provide a reliable audit trail when discrepancies arise.

## Relationship Sets and Their Details

- **BOOKINGS:** Links users with the events for which they have booked tickets. This relationship uses a composite candidate key consisting of `event_id` and `username` and enforces proper foreign key constraints to ensure data integrity between USERS and EVENTS.
- **EVENT\_PERFORMER:** Connects events with the performers participating in them. This many-to-many relationship captures associations where an event can feature multiple performers and a performer can participate in multiple events.

## Functional Dependencies in the Tixora Database

### USERS Table

Schema: (USERNAME, NAME, PHONE\_NUMBER, AGE, EMAIL\_ID, PASSWORD)

Functional Dependency:

**USERNAME** is the primary key, which uniquely determines **NAME**, **PHONE\_NUMBER**, **AGE**, **EMAIL\_ID**, and **PASSWORD**.

## EVENTS Table

**Schema:** (EVENT\_ID, EVENT\_TYPE, EVENT\_NAME, VENUE\_ID, PERFORMER\_ID, DATE\_AND\_TIME)

**Functional Dependencies:**

EVENT\_ID uniquely identifies EVENT\_TYPE, EVENT\_NAME, VENUE\_ID, and DATE\_AND\_TIME. Additionally, if no two events occur at the same venue simultaneously, then (VENUE\_ID, DATE\_AND\_TIME) determines EVENT\_ID.

## VENUE Table

**Schema:** (VENUE\_ID, VENUE\_NAME, LOCATION, TOTAL\_CAPACITY)

**Functional Dependency:**

VENUE\_ID is the primary key and determines VENUE\_NAME, LOCATION, and TOTAL\_CAPACITY.

## CLASS Table

**Schema:** (CLASS\_ID, TOTAL\_SEATS\_AVAILABLE, COST, TICKET\_PRICE, TOTAL\_AVAILABLE, EVENT\_ID)

**Functional Dependency:**

CLASS\_ID uniquely determines TOTAL\_SEATS\_AVAILABLE, COST, TICKET\_PRICE, TOTAL\_AVAILABLE, and EVENT\_ID.

## PERFORMERS Table

**Schema:** (PERFORMER\_ID, PERFORMER\_NAME, PERFORMER\_RATING)

**Functional Dependency:**

PERFORMER\_ID determines PERFORMER\_NAME and PERFORMER\_RATING.

## EVENT\_PERFORMER Table

**Schema:** (EVENT\_ID, PERFORMER\_ID)

**Functional Dependency:**

Each row is identified by the composite key (EVENT\_ID, PERFORMER\_ID), capturing a many-to-many relationship between events and performers.

## BOOKINGS Table

**Schema:** (USERNAME, BOOKING\_ID, EVENT\_ID, CLASS\_ID, TICKET\_COUNT, TICKETS\_VALUE, STAT, DATE\_AND\_TIME)

**Functional Dependency:**

(USERNAME, EVENT\_ID, BOOKING\_ID) collectively determines CLASS\_ID, TICKET\_COUNT, TICKETS\_VALUE, STAT, and DATE\_AND\_TIME.

## TRANSACTION\_DETAILS Table

**Schema:** (TRANSACTION\_ID, BOOKING\_ID, AMOUNT\_PAID, STATUS, DIRECTION, TRANSACTION\_TIMESTAMP)

**Functional Dependencies:**

TRANSACTION\_ID uniquely determines BOOKING\_ID, AMOUNT\_PAID, STATUS, DIRECTION, and TRANSACTION\_TIMESTAMP.

Furthermore, each BOOKING\_ID is associated with a unique TRANSACTION\_ID.

## Normalization Analysis

Based on the functional dependencies described above, the Tixora database design adheres to Third Normal Form (3NF) because:

- Every table has a clearly defined primary key.
- All non-key attributes are functionally dependent on the primary key.
- There are no transitive dependencies, as no non-key attribute depends on another non-key attribute.

## Conclusion

The Tixora database is designed to ensure data integrity and support efficient query processing through proper management of functional dependencies and normalization. By clearly organizing the entities and relationship sets and enforcing necessary foreign key constraints, the design provides a robust foundation for our event ticket booking system.