



TICKETING MANAGEMENT SYSTEM T.M.S

Abdul Raheem Goraya 2023170

Taha Shafique 2023716

Muhammad Hamaad 2023420

Abstract:

This project presents the development of a complete ticketing management system designed to make event planning easier and improve how participants interact with events. By applying basic principles of database management, the system helps with smooth handling of bookings and event logistics. Key features include event scheduling, ticket reservations, coordination with performers or guests, and keeping track of finances. The goal was to turn classroom knowledge into something practical and useful, and this system does just that. It's built to meet the real needs of university societies, offering a simple and effective way to manage their events. Overall, the project hopes to make organizing university events more efficient, more engaging, and a better experience for everyone involved.

Introduction

Organizing university events can be exciting, but it often comes with a lot of challenges like from handling ticket bookings to managing venues and keeping everything on schedule. To make things easier, we developed a Ticketing Management System (TMS) aimed at simplifying how student clubs and societies plan and manage their events.

Picture a campus filled with active clubs, each hosting different types of events. Without a centralized system, managing these can get messy. That's where TMS comes in. It acts like a smart helper that brings everything together like event planning, ticketing, guest lists, venue management, and even payment handling.

One of the biggest benefits of TMS is how user-friendly it is. Whether it's booking tickets, updating event details, or cancelling plans, everything can be done quickly and easily. The system also includes secure payment options, so students and organizers don't have to worry about safety when handling transactions.

Behind the scenes, TMS is built with features to make sure events don't exceed capacity, and all data stays accurate and consistent. It's a big step up from using scattered tools or doing things manually.

In short, this system is designed to save time, reduce stress, and improve how events are managed on campus. It's not just a tool, it's a better way to bring student-led events to life.

Problem Statement:

Planning and managing university events often involves outdated methods that can lead to confusion, errors, and unnecessary stress for both organizers and attendees. The Ticketing Management System (TMS) aims to solve several major issues commonly faced in traditional event setups.

One of the main problems is the lack of efficiency in managing event logistics. Paper-based ticketing, manual spreadsheets, and last-minute coordination often result in errors, delays, and miscommunication. By digitizing these tasks, TMS reduces the chances of things slipping through the cracks and helps keep everything organized from start to finish, including tasks like handling registration desks.

Another challenge is the limited and sometimes frustrating booking process. When tickets are sold manually by society members, at on-campus cafés, or through informal means, the experience can feel unorganized or unclear to attendees. There's also the risk of overbooking or losing track of ticket sales. TMS tackles this with a user-friendly interface and clear, automated features that make booking smooth and transparent for everyone.

To summarize, TMS directly addresses the following problems:

- Inefficient Event Management: Manual processes like paper-based systems and spreadsheets are slow, error-prone, and hard to coordinate.
- Poor Booking Experience: Traditional ticket sales methods often lack convenience, clarity, and flexibility for both organizers and attendees.
- By offering a modern, all-in-one platform, the Ticketing Management System brings structure, speed, and simplicity to campus event planning.

Product overview

The Optimized Ticketing Management System is a powerful and efficient platform designed to manage events, societies, and related activities for universities. This system offers a userfriendly interface, advanced features, and improved performance, ensuring a seamless and enjoyable experience for both administrators and users.

Key Features:

1. **Database Schema Optimization:** The system features a thoroughly analyzed and optimized database schema, ensuring performance, efficiency, and adherence to industry best practices and standards.
2. **Data Normalization and Integrity:** The system enforces data normalization principles and ensures data consistency, maintaining accurate records and efficient database operations.
3. **Best Practices and Standards:** The system adopts and implements industry best practices and standards to improve maintainability, scalability, and security.
4. **User-friendly Interface:** The system maintains an intuitive and user-friendly interface, making it easy for users to manage events, societies, and related activities.
5. **Advanced Features:** The system includes a wide range of advanced features, such as event scheduling, resource management, society management, and user management.

Benefits:

6. **Enhanced Performance:** The optimized database schema and query optimizations lead to faster response times and improved overall system performance.
7. **Improved Data Accuracy:** Data normalization and integrity measures ensure accurate records and efficient database operations.
8. **Scalability and Maintainability:** Adherence to best practices and standards ensures the system remains scalable and maintainable as it grows and evolves.
9. **Better User Experience:** The user-friendly interface and advanced features make it easy for users to manage events, societies, and related activities, improving the overall user experience.

Product Features:

Intuitive and Modern front end Design: The system boasts a clean and modern user interface, ensuring a pleasant user experience for both administrators and members of the society.

User-Friendly Interface: Designed with simplicity in mind, the system is easy to navigate and requires minimal training for users to get acquainted with its functionalities.

Scalable Architecture: Built on a flexible and scalable architecture, the system can accommodate the evolving needs of your society as it grows in size and complexity.

Precision and Reliability: The system operates with utmost precision, ensuring accurate management of society-related tasks and data.

Seamless Workflow: With a smooth and streamlined workflow, users can effortlessly perform tasks such as event planning, member management, and communication within the society.

Efficient Information Retrieval: Users can quickly and easily retrieve relevant information, such as event details saving time and effort.

Powerful Database Backend: Supported by a robust and scalable database backend, the system efficiently stores and manages society-related data, ensuring reliability and performance.

Easy Information Updates: Administrators can effortlessly update and maintain society-related information, such as event schedules, artist lists, and ticket price, keeping the system up-to-date at all times.

Unique Booking Numbers : Automate Unique Booking Numbers ; the generation of unique booking numbers for each booking. This feature streamlines the booking process and improve customer service by providing a straightforward way for customers to track their bookings.

Decreases the Load of the Person Involved in Existing Manual System: The system automates repetitive tasks and processes, reducing the manual workload of society organizers. By eliminating tedious administrative tasks, such as manual record-keeping and coordination, individuals involved in managing the society can focus their efforts on more strategic and impactful activities.

Time Efficiency: Users can accomplish tasks more efficiently and effectively, thanks to the system's streamlined workflows and intuitive interface, saving valuable time and resources.

Space Efficiency: The system optimizes storage space by efficiently organizing and storing data, minimizing redundant information and maximizing resource utilization.

Functional Requirements:

Event Details and Schedule Viewing: Users can access comprehensive information about upcoming events, including their date, time, and venue. This feature enables clients to plan their schedules effectively and make informed decisions about which events to attend, thereby enhancing user experience and event management efficiency.

Event Pricing Display: Clients can easily ascertain the cost associated with attending different events, aiding in budget planning and decision-making. This feature provides transparency and clarity regarding event expenses, ensuring clients can make financially informed choices.

Event Catalog: The system categorizes events by type, such as concerts, qawwalis, fashion shows, Pakistani cultural events, and theme dinners, facilitating event discovery and exploration. Users can browse through a diverse range of options based on their preferences, thereby enhancing engagement and satisfaction.

Artist Information: Details about the artists performing at each event add excitement and prestige to the offerings. This feature enables users to learn about the talent lineup and make event selections based on their preferences and interests, thereby enhancing the overall event experience.

Event Dates: Clients can access event dates to pre-plan their schedules effectively, ensuring they can attend desired events without scheduling conflicts. This functionality enhances user convenience and attendance rates while optimizing event planning and logistics.

Ticket Booking: Users can conveniently book tickets for desired events, streamlining the booking process and accommodating group bookings. The system allows for the booking of multiple tickets in a single transaction, enhancing user convenience and satisfaction.

Booking Confirmation and Cancellation: Upon successful booking, clients receive a confirmation message, instilling confidence in the booking process and ensuring peace of mind. Similarly, a confirmation message for cancellation reassures clients of successful cancellation and refunds if applicable, enhancing user trust and satisfaction.

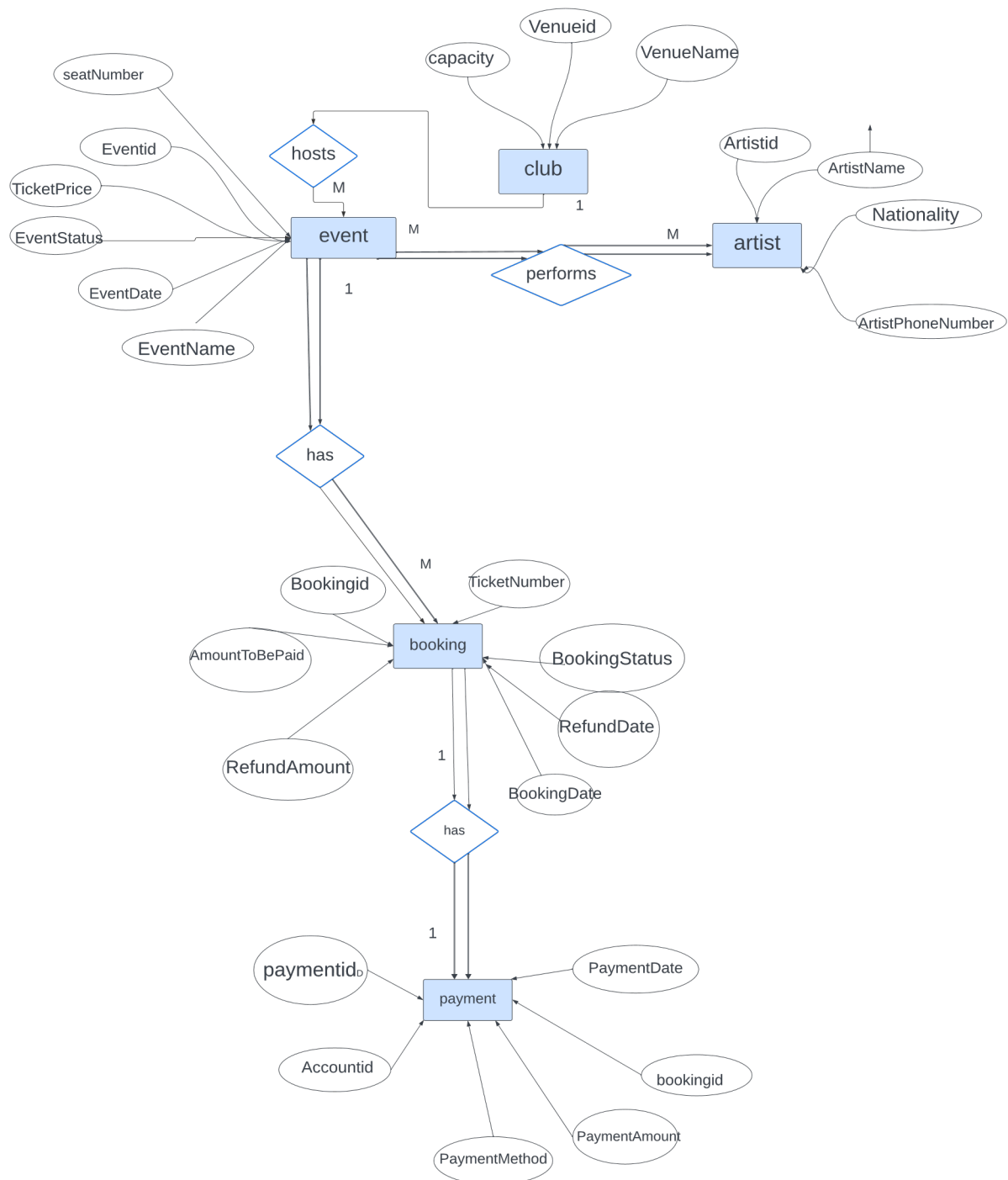
Online Payment Options: The system offers various secure online payment options, providing flexibility and convenience for clients to choose their preferred payment method. This feature ensures seamless and secure transactions, thereby enhancing user experience and satisfaction.

Payment Details Management: Secure processing and storage of client payments in the backend database ensure data integrity and financial transparency. This functionality is essential for maintaining accurate records and facilitating financial transactions seamlessly, thereby enhancing overall system efficiency and reliability.

Valid booking ID Generation: A function is implemented in the backend code that takes a booking ID as input and queries the database to check if the ID exists. This function returns a message indicating whether the ID is valid. This message clearly indicates that the booking ID is invalid (e.g., checking the booking ID for typos).

Preventing OverBooking: Constarint implemented to prevent booking for an event if the capacity is reached for the number of seats.

ER Diagram:



Explanation:

In this diagram each rectangle represents an entity (Club, Artist, Events, Booking, Payment).

Each oval represents an attribute within the entity.

Diamonds represent relationships between entities, and lines connect these relationships. The notation "1" and "N" indicate one-to-many relationships, where "1" denotes one occurrence and "N" denotes many occurrences. The detailed explanation of this diagram is given below

Entity 1 : Club

Attributes: Venue Id, Venue Name, Capacity

Entity 2: Artist

Attributes: Artist Id , Artist Name , Nationality, Artist Phone Number

Entity 3: Events

Attributes: Event Id , Event Name, Seat Number, Ticket Price, Event Status, Date

Entity 4: Booking

Attributes: Booking Id , Ticket Number, Amount To Be Paid, Booking Status, Booking Date, Refund Amount, Refund Date

Entity 5: Payment

Attributes: Payment Id , Payment Amount, Payment Date, Payment Method, Account id

Relationships:

One to Many:

Many Bookings can be made for One Event (1:N relationship)

Many Bookings can be made for One Event (N:1 relationship)

Many Events can be held at One Club (1:N relationship)

One Club can host Many Events (N:1 relationship)

One to One:

One Payment can be made for One Booking

Many to Many:

Many Artists can participate in Many Events

TABLE DESCRIPTIONS

Club

Attribute	Description	Data type	Value required	Keys
VenueId	Unique set of numbers used as an identifier	INTEGER	Yes	PK
VenueName	Name of the event	VARCHAR (50)	Yes	-
Capacity	The max amount the venue can hold	INTEGER	Yes	-

Event

Column	Description	Data type	Value required	Keys	Special domain
EventId	Unique set of numbers used as an identifier	INTEGER	Yes	PK	-
EventName	Name of the event	VARCHAR(50)	Yes	-	
SeatNumber	Specifies the number of seats for the event	INTEGER	Yes	-	-
TicketPrice	price of each ticket to the event	INTEGER	Yes	-	
ArtistId	Unique set of numbers used as an identifier	INTEGER	Yes	FK	
VenueId	Unique set of numbers used as an identifier	INTEGER	Yes	FK	
EventStatus	Status of the event	VARCHAR (50)	YES	-	Cancelledpastactive
EventDate	the date of the event	DATE	Yes	-	dd/mm/yyyy

ENTITY TYPE DESCRIPTIONS (Admin Controlled)

ENTITY TYPE	Description	Occurrence
Club	The venue that host all the events	Used every time there is an event
Artist	An artist who is going to be performing	Multiple occurrences, can be used in many events
Event	DiTerent types of performances	Occasional occurrences, one for each event.

Artist

Column	Description	Data type	Value required	Keys	Special domain
ArtistId	Unique set of numbers used as an identifier	INTEGER	Yes	PK	
ArtistName	Unique name of the artist	VARCHAR (50)	Yes	-	-
Nationality	Specifies where the artist is from	VARCHAR (50)	Yes	-	-
ArtistPhoneNumber	Phone number of artists	INTEGER	YES	-	-

Booking

Attribute	Description	Data type	Value required	Keys	Special domain
bookingNumber	Unique set of numbers used as an identifier	INTEGER	Yes	PK	-
ticketNumber	Number of tickets booked for the event	INTEGER	Yes	-	-
eventId	Unique set of numbers used as an identifier	INTEGER	Yes	FK	-
bookingStatus	Shows the status of the booking	VARCHAR (50)	Yes	-	Cancelledactivesold
AmountTobePaid	Total amount to be paid for each booking	INTEGER	Yes	-	-
BookingDate	Date the booking was created	DATE	Yes	-	-

Payment

Column	Description	Data type	Value required	Keys	Special domain
Account Number	Unique set of numbers used as an identifier	INTEGER	Yes	PK	-
BookingNumber	Unique set of numbers used as an identifier	INTEGER	Yes	FK	-
PaymentAmount	the amount payed	INTEGER	Yes	-	-

FRONT END

Our user interface serves as the gateway for users to interact with our society management system seamlessly. In designing our front end, we prioritized aesthetics while maintaining a minimalist approach to enhance user experience. Opting for Flask over Node.js for our frontend development provided us with a robust framework offering a plethora of options and functionalities, aligning perfectly with our project requirements and ensuring efficient design implementation.

To bring our front end to life, we utilized Flask to integrate the HTML pages seamlessly with our Python backend, ensuring smooth navigation and interaction for users. Leveraging the Flask framework allowed us to streamline development and maintain consistency throughout the interface.

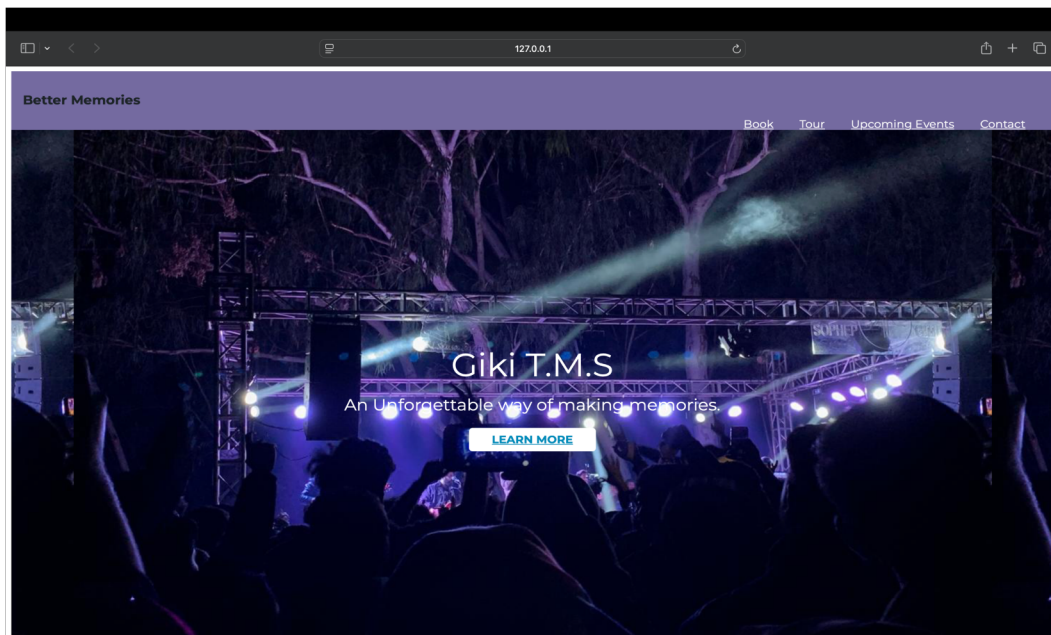
In crafting our front end, we relied on a selection of essential libraries and tools, for advanced design capabilities, ensuring visually appealing and user-friendly interfaces. Additionally, we utilized pgAdmin to establish a connection between our implementation and the underlying database, facilitating seamless data management and integration.

Some notable examples of our front end design include:

1. Intuitive navigation events for effortless exploration of system functionalities.
2. Clean and visually appealing layouts, enhancing readability and user engagement.
3. Interactive elements such as buttons and icons for seamless interaction and navigation.
4. Consistent design elements and color schemes for a cohesive user experience across all pages.
5. Responsive design ensuring, enhancing accessibility and usability.

By prioritizing aesthetics, functionality, and user experience, our front end design ensures a seamless and enjoyable interaction for users, empowering them to efficiently manage ticketing system.

Screen Shots of our site:



127.0.0.1

BOOK YOUR TICKETS HERE

Select Event:

Festival 1

Number of Tickets:

Book Now

Enter Booking ID to Cancel:

Cancel Booking

127.0.0.1

f i t

ADDITIONAL BOOKING

Username:


Account ID:

Email:

Payment Method:

Credit Card

Confirm Booking



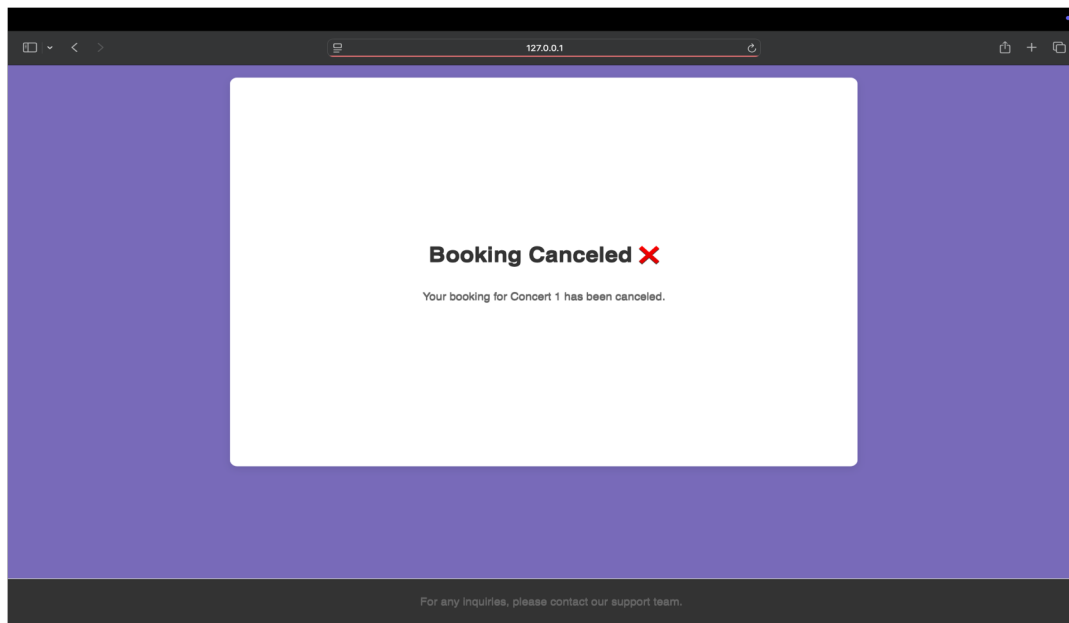
127.0.0.1

Booking Confirmation

Your booking has been confirmed:

Event ID:	3
Event Name:	Festival 1
Tickets:	1
Account Name:	None
Account ID:	30
Email:	haah@gmail.com
Total Price:	RSV- 80.0
Booking ID:	23

To cancel this booking, please use the **Booking ID: 23**. Follow the instructions provided by our customer service team.



Normalization Process

Step 1: Unnormalized Form (UNF)

In UNF, the data may include repeating groups and non-atomic values. From the ERD, we can see that all attributes are atomic, and no multivalued fields are present, so we begin with structured data suitable for 1NF.

Step 2: First Normal Form (1NF)

1NF Criteria:

- Atomic attributes (no lists or sets)
- Each record is uniquely identifiable

We separate all entities into their own tables with clearly defined primary keys.

Tables in 1NF:

1. **Club** (ClubID, VenueName, Capacity)
2. **Artist** (ArtistID, ArtistName, Nationality, ArtistPhoneNumber)

3. **Event** (EventID, EventName, EventDate, SeatNumber, TicketPrice, EventStatus, ClubID, ArtistID)
4. **Booking** (BookingID, TicketNumber, AmountToBePaid, BookingStatus, BookingDate, RefundAmount, RefundDate, EventID)
5. **Payment** (PaymentID, PaymentDate, PaymentMethod, PaymentAmount, AccountID, BookingID)
6. **Account** (AccountID, AccountName, Email) (*Inferred from AccountID used in Payment table*)

Step 3: Second Normal Form (2NF)

2NF Criteria:

- Meet First Normal Form (1NF)
- No partial dependency (all non-key attributes fully depend on the whole primary key)

Since all primary keys are atomic (i.e., not composite), each table in 1NF already meets 2NF.

Step 4: Third Normal Form (3NF)

3NF Criteria

To be in Third Normal Form (3NF), a table must:

- Already meet the requirements of Second Normal Form (2NF)
- Have no transitive dependencies (i.e. non-key attributes should not depend on other non-key attributes)

Club

There are no transitive dependencies in this table.
This table is already in 3NF.

Artist

All non-key attributes directly depend on the primary key. No transitive dependencies found.
This table is in 3NF.

Event

Both ClubID and ArtistID are foreign keys, and the rest of the attributes depend only on EventID.
The table satisfies 3NF.

Booking

EventID is a foreign key, and all other fields are dependent on BookingID.
This table is in 3NF as well.

Payment

This table includes AccountID and BookingID as foreign keys, while all other attributes are directly related to PaymentID. No transitive dependencies so it's in 3NF.

Account

There are no derived or transitive attributes in this table. It also meets the criteria for 3NF.

Final Tables (3NF)

Table	Attributes
Club	ClubID (PK), VenueName, Capacity
Artist	ArtistID (PK), ArtistName, Nationality, ArtistPhoneNumber
Event	EventID (PK), EventName, EventDate, SeatNumber, TicketPrice, EventStatus, ClubID (FK), ArtistID (FK)
Booking	BookingID (PK), TicketNumber, AmountToBePaid, BookingStatus, BookingDate, RefundAmount, RefundDate, EventID (FK)
Payment	PaymentID (PK), PaymentDate, PaymentMethod, PaymentAmount, AccountID (FK), BookingID (FK)
Account	AccountID (PK), AccountName, Email

Practical Applications:

Mobile Application Development: Developing a mobile application companion for the system can enhance accessibility and convenience for users, allowing them to book tickets, receive updates, and manage bookings on-the-go, thereby increasing user engagement and satisfaction.

Customization and Personalization: Introducing features for customizable event listings, personalized recommendations, and tailored promotional offers can enhance the user experience, cater to diverse preferences, and drive increased participation in society events.

Ticket Booking Database project holds immense potential to transform the event management landscape within university settings. Through continued development, integration of innovative features, and strategic partnerships, it can further elevate the user experience, expand its

impact, and solidify its position as a indispensable tool for organizing and promoting society event.

Conclusion

In conclusion, our project represents a significant step forward in addressing the needs of project management within our campus. Through the adoption of user-friendly coding techniques and a comprehensive approach to software planning, we have developed a powerful package that meets the requirements of both our students and our university.

Throughout the development process, we have focused on key areas to ensure the success of our project. We began by providing a thorough introduction and context for the project, highlighting its relevance and relationship to existing work in the field. This laid the foundation for a clear understanding of the project scope, purpose, and team members involved.

We meticulously defined the project, outlining its objectives, functionalities, and requirements. This helped us establish a framework for development and guided our efforts towards meeting the needs of our users effectively.

One of our primary goals was to design a user interface that is intuitive and ensures the privacy of sensitive information. By incorporating the best practices in interface design, we created an interface that is both user-friendly and secure.

