



Mini Project Report On

TicketBuddy

*Submitted in partial fulfillment of the requirements for the
award of the degree of*

Bachelor of Technology

in

Computer Science & Engineering

By

Nithin John (U2103160)

Paul Allen Kadayaparambil (U2103164)

Ryan Thomas (U2103186)

Thomas T Thayyil (U2103210)

Under the guidance of

Dr. Uma Narayanan

**Department of Computer Science & Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Affiliated to APJ Abdul Kalam Technological University)**

Rajagiri Valley, Kakkanad, Kochi, 682039

May 2024

CERTIFICATE

*This is to certify that the mini project report entitled **TicketBuddy** is a bonafide record of the work done by **Nithin John (U2103160)**, **Paul Allen Kadayaparambil (U2103164)**, **Ryan Thomas (U2103186)**, **Thomas T Thayyil (U2103210)**, submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Project Guide
Dr. Uma Narayanan
Asst. Professor
Dept. of CSE
RSET

Project Coordinator
Dr. Uma Narayanan
Asst. Professor
Dept. of CSE
RSET

Head of the Department
Dr. Preetha K. G.
Professor
Dept. of CSE
RSET

ACKNOWLEDGEMENTS

We wish to express our sincere gratitude towards **Dr P. S. Sreejith**, Principal of RSET, and **Dr. Preetha K.G.**, Head of the Department of Computer Science and Engineering for providing us with the opportunity to undertake our mini project, **TicketBuddy**.

We are highly indebted to our project coordinators, **Dr. Uma Narayanan**, Asst. Professor, Department of Computer Science and Engineering, and **Ms. Liya Joseph**, Asst. Professor, Department of Computer Science and Engineering for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Dr. Uma Narayanan** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, We would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Nithin John

Paul Allen Kadayaparambil

Ryan Thomas

Thomas T Thayyil

Abstract

TicketBuddy is an innovative ticketing platform designed to optimize the purchase, vending, and refund processes for event tickets. It serves as a dynamic marketplace that connects buyers and sellers while ensuring fairness and user-friendliness. Through its disruptive approach, TicketBuddy revolutionizes traditional ticketing, providing a seamless and enjoyable experience for all involved in the event ecosystem. Operating at the intersection of innovation and user-centric design, TicketBuddy ensures a hassle-free and enjoyable experience for both buyers and sellers, setting a new standard in the ticketing industry with its commitment to excellence and equity.

TicketBuddy's unique features distinguish it from existing ticketing platforms in several key ways. Firstly, its innovative bidding mechanism maximizes profits for both buyers and sellers, creating a more dynamic and competitive marketplace. Secondly, TicketBuddy adopts a holistic approach, offering tickets across a diverse range of events, from music concerts to sports matches and theater shows, catering to the varied interests of its user base.

Beyond these core advantages, TicketBuddy stands out for its user-friendly interface and intuitive design, making it easier than ever for users to navigate and purchase tickets. Its innovative bidding mechanism further incentivizes sellers to list their tickets on the platform, enhancing ticket availability and variety. Additionally, TicketBuddy's commitment to transparency and security ensures that users can buy and sell tickets with confidence, knowing that they are getting a fair deal and that their personal information is protected.

Contents

| | |
|--|-------------|
| Acknowledgements | i |
| Abstract | ii |
| List of Figures | vi |
| List of Tables | vii |
| List of Abbreviations | viii |
| 1 Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Problem Definition | 2 |
| 1.3 Scope and Motivation | 2 |
| 1.3.1 Scope | 2 |
| 1.3.2 Motivation | 2 |
| 1.4 Objectives | 3 |
| 1.5 Challenges | 3 |
| 1.6 Assumptions | 3 |
| 1.7 Societal/Industrial Relevance | 4 |
| 1.8 Organization of the Report | 4 |
| 2 Software Requirements Specification | 5 |
| 2.1 Introduction | 5 |
| 2.1.1 Purpose | 5 |
| 2.1.2 Product Scope | 5 |
| 2.2 Overall Description | 5 |
| 2.2.1 Product Perspective | 5 |
| 2.2.2 Operating Environment | 6 |

| | | |
|----------|--|-----------|
| 2.2.3 | Product Functions | 6 |
| 2.2.4 | Design and Implementation Constraints | 7 |
| 2.2.5 | Assumptions and Dependencies | 8 |
| 2.3 | External Interface Requirements | 9 |
| 2.3.1 | User Interfaces | 9 |
| 2.3.2 | Hardware Interfaces | 11 |
| 2.3.3 | Software Interfaces | 13 |
| 2.3.4 | Communication Interfaces | 14 |
| 2.4 | System Features | 16 |
| 2.4.1 | Ticket Listing Management | 16 |
| 2.4.2 | Bid Management | 18 |
| 2.4.3 | Ticket Return Management | 20 |
| 2.5 | Other Nonfunctional Requirements | 22 |
| 2.5.1 | Performance Requirements | 22 |
| 2.5.2 | Safety Requirements | 23 |
| 2.5.3 | Security Requirements | 25 |
| 2.5.4 | Software Quality Attributes | 27 |
| 3 | System Architecture and Design | 29 |
| 3.1 | System Overview | 29 |
| 3.2 | Architectural Design | 31 |
| 3.3 | Proposed Methodology/Algorithms | 31 |
| 3.4 | User Interface Design | 33 |
| 3.5 | Database Design | 37 |
| 3.6 | Description of Implementation Strategies | 38 |
| 3.7 | Module Division | 39 |
| 3.8 | Work Schedule - Gantt Chart | 41 |
| 4 | Results and Discussions | 42 |
| 4.1 | Overview | 42 |
| 4.2 | Testing | 42 |
| 4.3 | Discussion | 46 |

| | |
|--|-----------|
| 5 Conclusion | 47 |
| 5.1 Conclusion | 47 |
| 5.2 Future Scope | 47 |
| Appendix A: Presentation | 49 |
| Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes | 65 |
| Vision, Mission, POs, PSOs and COs | 2 |
| Appendix C: CO-PO-PSO Mapping | 6 |

List of Figures

| | | |
|------|---------------------------------------|----|
| 3.1 | System Architecture Diagram | 30 |
| 3.2 | Use Case Diagram | 31 |
| 3.3 | Sign Up Page | 33 |
| 3.4 | Sign In Page | 34 |
| 3.5 | Home Page | 34 |
| 3.6 | Home Page Dropdown Menu | 35 |
| 3.7 | Booking Page | 35 |
| 3.8 | Bidding Page | 36 |
| 3.9 | Your Tickets Page | 36 |
| 3.10 | ER Diagram | 37 |
| 3.11 | Gantt Chart | 41 |
| 4.1 | Sign Up Page | 42 |
| 4.2 | Sign In Page | 43 |
| 4.3 | Home Page | 43 |
| 4.4 | Home Page Dropdown Menu | 44 |
| 4.5 | Booking Page | 44 |
| 4.6 | Bidding Page | 45 |
| 4.7 | Your Tickets Page | 45 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | Ticket Listing Management Priority Components | 17 |
| 2.2 | Ticket Listing Management Requirements | 18 |
| 2.3 | Bid Management Priority Components | 19 |
| 2.4 | Bid Management Requirements | 20 |
| 2.5 | Ticket Return Management Priority Components | 20 |
| 2.6 | Ticket Return Management Requirements | 21 |

List of Abbreviations

- ACID - Atomicity, Consistency, Isolation, Durability
- API - Application Programming Interface
- CCPA - California Consumer Privacy Act
- CORS - Cross-Origin Resource Sharing
- CSS - Cascading Style Sheets
- DBMS - Database Management System
- DDoS - Distributed Denial of Service
- GDPR - General Data Protection Regulation
- GUI - Graphical User Interface
- GraphQL - Graph Query Language
- HTML - HyperText Markup Language
- HTTP - Hypertext Transfer Protocol
- HTTPS - Hypertext Transfer Protocol Secure
- IP - Internet Protocol
- ISO - International Organization for Standardization
- ISO 27001 - Information Security Management System
- JS - JavaScript
- NIST SP 800-63 - National Institute of Standards and Technology Special Publication 800-63
- ORM - Object-Relational Mapping
- PCI DSS - Payment Card Industry Data Security Standard
- RESTful API - Representational State Transfer Application Programming Interface
- SLA - Service Level Agreement
- SMTP - Simple Mail Transfer Protocol
- SRS - Software Requirement Specification
- SQL - Structured Query Language
- SOC 2 - Service Organization Control 2

TCP - Transmission Control Protocol

TLS - Transport Layer Security

UI - User Interface

Chapter 1

Introduction

1.1 Background

In today's digital age, the event ticketing industry has witnessed significant growth, with an increasing number of consumers preferring online platforms for purchasing tickets to their favorite events. Traditional ticketing methods, characterized by long queues, limited availability, and often inflated prices due to middlemen, have become increasingly inconvenient for both buyers and sellers.

Moreover, the surge in various types of events, from concerts and sports matches to theater shows and festivals, has led to a fragmented market with multiple platforms catering to specific niches. This fragmentation often results in users needing to juggle between different platforms to find tickets for different events, leading to a disjointed user experience.

Amidst this backdrop, TicketBuddy emerges as a transformative solution designed to address these challenges head-on. It aims to streamline the ticketing process, making it more efficient, transparent, and user-centric. By connecting buyers directly with sellers in a dynamic marketplace, TicketBuddy eliminates the need for middlemen, ensuring fair prices and a more competitive environment.

Furthermore, TicketBuddy's innovative bidding mechanism adds a new dimension to ticket purchasing. This not only benefits buyers but also incentivizes sellers to list their tickets on the platform, thereby enhancing ticket availability and variety.

In essence, TicketBuddy is not just another ticketing platform; it's a revolution in the way tickets are bought, sold, and managed. By combining innovation with user-centric

design, TicketBuddy sets a new standard in the ticketing industry, promising a seamless and enjoyable experience for all stakeholders involved in the event ecosystem.

1.2 Problem Definition

The TicketBuddy project aims to revolutionize the event ticketing industry with a web-based platform enabling sellers to optimize revenue through dynamic pricing and auctions, and buyers to benefit from innovative compensation mechanisms. This comprehensive solution addresses inefficiencies, ensuring transparency, fairness, and profitability for all stakeholders.

1.3 Scope and Motivation

1.3.1 Scope

The scope of the TicketBuddy project encompasses the development of a web-based platform tailored for both sellers and buyers in the event ticketing market. For sellers, TicketBuddy offers a platform to list and sell tickets with tools designed to maximize profit through dynamic pricing and auction mechanisms. Concurrently, for buyers, the platform provides an opportunity to purchase tickets while potentially profiting from TicketBuddy's compensation mechanisms. This platform will feature functionalities such as ticket listing management, real-time bid processing, and ticket return management. Additionally, it will support electronic forms for user interactions and adhere to industry-standard communication protocols and security measures. The platform aims to provide a user-friendly interface that allows sellers to easily manage listings and bids, and buyers to bid on available tickets, receive real-time updates, and make secure purchases.

1.3.2 Motivation

The motivation behind the TicketBuddy project stems from the identified gaps and challenges in the current event ticketing landscape. Many existing platforms lack transparency, leading to unfair ticket pricing and distribution. Additionally, users often face difficulties in managing their listings, bids, and returns efficiently. The increasing demand for online ticketing solutions, coupled with the potential for improving user experience and security, provides a compelling reason to develop TicketBuddy. By addressing these

issues, TicketBuddy aims to revolutionize the event ticketing industry by offering a transparent, efficient, and secure platform for all stakeholders.

1.4 Objectives

- Boost Seller Revenue through Dynamic Pricing: Implement tools that allow sellers to set flexible pricing strategies based on demand, maximizing their profitability.
- Enable Buyers to Profit from Refunds: Offer buyers the opportunity to purchase tickets and potentially profit from TicketBuddy's compensation mechanisms.
- Real-Time Bid Processing: Enable quick and accurate bid updates for sellers, ensuring they stay informed and can adjust strategies accordingly.
- Transparent Ticket Pricing: Ensure fairness and transparency in ticket pricing to eliminate inflated costs often seen with middlemen.
- Enhanced User Experience: Design a user-friendly interface for both buyers and sellers, focusing on ease of use and intuitive navigation.
- Robust Security Measures: Implement industry-standard security protocols to safeguard user data and transactions, ensuring trust and reliability.

1.5 Challenges

- Addressing the fragmented nature of the event ticketing market with its multiple platforms, which can lead to a disjointed user experience.
- Implementing a dynamic pricing and auction system while ensuring real-time bid processing, robust security, and a user-friendly interface to enhance overall platform experience and trust.

1.6 Assumptions

- Sellers and buyers are motivated to actively engage with and utilize the TicketBuddy platform's features.

- The technological infrastructure is robust enough to facilitate seamless real-time bid processing and dynamic pricing adjustments.
- Users have confidence in TicketBuddy's compensation mechanisms, promoting adoption and trust in ticket transactions.

1.7 Societal/Industrial Relevance

TicketBuddy aligns with the contemporary shift towards digital solutions in the event ticketing sector, offering a platform that addresses critical inefficiencies and promotes transparency. In an industry where trust and user experience are paramount, TicketBuddy's innovative approach not only elevates the standards of ticketing transactions but also sets a precedent for how technology can harmonize buyer and seller interests. Its application holds significant potential to reshape the dynamics of the event ticketing market, fostering a more streamlined and equitable marketplace for all stakeholders involved.

1.8 Organization of the Report

The report is organized into five main sections. The introduction contextualizes the event ticketing landscape, outlines TicketBuddy's objectives, and highlights its societal and industrial relevance. The Software Requirements Specification provides an overview of TicketBuddy's technical requirements, including system features and nonfunctional considerations. Moving to the System Architecture and Design section, it details TicketBuddy's technical blueprint, covering system overview, architectural design, user interactions, and project timeline. Results and Discussions present project outcomes, testing procedures, detailed results, and analysis, aligning with project objectives. Finally, the Conclusion summarizes key findings, discusses future scope, and suggests areas for further research and development.

Chapter 2

Software Requirements Specification

2.1 Introduction

2.1.1 Purpose

The purpose of this chapter is to present a detailed description of the TicketBuddy ticket booking application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do and how it will benefit all stakeholders using the application. This document is intended for both the stakeholders and the developers of the system.

2.1.2 Product Scope

This software system has three primary purposes. Firstly, it aims to provide a platform that allows entities to sell tickets to the venue(s) pertaining to the event(s) conducted by them. Secondly, it strives to maximize the profits of entities that will use TicketBuddy to sell tickets via its auctioning mechanism. Lastly, it seeks to utilize TicketBuddy's compensations mechanism to allow individuals to potentially profit off of its auctioning system. Further details of the compensation mechanism will be disclosed later in the document.

2.2 Overall Description

2.2.1 Product Perspective

TicketBuddy is a new concept created by our team and its manifestation as an application is completely independent of any other such applications. The context of TicketBuddy arises from the need to address challenges and inefficiencies in the ticket purchasing process, such as limited availability, high demand, and scalping. It aims to provide a fair and

transparent platform for both ticket sellers and buyers, enhancing the overall experience and increasing accessibility to tickets for various events.

2.2.2 Operating Environment

The TicketBuddy platform will be developed as an online web application, ensuring compatibility across a variety of devices. It will be accessible on desktop computers, laptops, tablets, and smartphones, provided the device has an active internet connection. The application will support major operating systems, including Windows, macOS, Linux, iOS, and Android. Additionally, it will be compatible with all modern web browsers such as Chrome, Firefox, Safari, and Edge, ensuring a seamless user experience regardless of the device or browser used.

2.2.3 Product Functions

1. Ticket Selling:

- Allows ticket sellers to list tickets for events.
- Provides a platform for sellers to manage ticket inventory and pricing.

2. Ticket Purchasing:

- Enables users to purchase tickets through both normal mode and auction mode.
- Implements a bidding mechanism for users to compete for tickets in auction mode.

3. Auction Mode:

- Activates when all available tickets are waitlisted in normal mode.
- Facilitates bidding on tickets, with higher bids securing higher positions in the waitlist.

4. Waitlist Management:

- Maintains primary and secondary waitlists to keep track of bidders.
- Automatically upgrades positions based on ticket returns and bidding outcomes.

5. Transparency:

- Displays all bids publicly to allow strategic bidding.
- Ensures the identity of bidders is kept discreet.

6. Compensation Mechanism:

- Provides compensation to individuals who are outbid and removed from the waitlist.
- Incentivizes individuals to bid higher to increase their chances of securing tickets.

7. Return Policy Management:

- Handles ticket returns, including collecting cancellation fees and adjusting waitlist positions accordingly.
- Facilitates communication with users regarding reclaimed tickets and waitlist status updates.

2.2.4 Design and Implementation Constraints

1. Regulatory Compliance: Developers must ensure compliance with regulations governing ticket sales, auctions, and online transactions, including consumer protection laws, privacy regulations, and payment processing regulations.
2. Security Considerations: Given the sensitive nature of financial transactions and personal data involved in ticket purchasing, the website must implement robust security measures to protect against unauthorized access, data breaches, and fraud.
3. Payment Processing Integration: The website will need to integrate with third-party payment processors to facilitate transactions securely. Developers may be limited in their choice of payment gateways based on factors such as compatibility, transaction fees, and security features.
4. Scalability and Performance: As the website may experience fluctuations in traffic during peak times or when popular events are on sale, developers need to design the

system to handle high volumes of traffic efficiently. This includes considerations for load balancing, caching, and database optimization.

5. Technological Stack: The choice of technologies, tools, and databases used in developing the website may be influenced by factors such as the expertise of the development team, compatibility with existing systems, scalability requirements, and budget constraints.
6. User Interface Design: The website's design must adhere to usability and accessibility standards to ensure an optimal user experience for a diverse range of users. Developers may need to consider factors such as responsive design, compatibility with different devices and browsers, and accessibility features for users with disabilities.
7. Data Privacy and Compliance: Developers must adhere to data privacy regulations.
8. Maintenance and Support: The developers may need to adhere to specific programming standards or design conventions if the customer's organization will be responsible for maintaining the software after deployment. This could include documentation requirements, coding standards, and version control practices.

2.2.5 Assumptions and Dependencies

1. Third-party APIs and Services:
 - Assumption: Integration with third-party payment processors, email services, and other APIs will be smooth and reliable.
 - Impact: If the APIs change or become unavailable, it could affect payment processing, communication with users, and other critical functions.
2. Scalability and Performance:
 - Assumption: The anticipated user base and traffic patterns align with the initial scalability and performance estimates.
 - Impact: If the actual usage exceeds expectations, it may require significant adjustments to the system architecture and infrastructure to ensure optimal performance.

3. Security Measures:

- Assumption: The implemented security measures, such as encryption, authentication, and authorization, are sufficient to protect user data and prevent unauthorized access.
- Impact: Inadequate security measures could lead to data breaches, financial loss, and damage to the reputation of TicketBuddy.

4. User Behavior and Adoption:

- Assumption: Users will understand and adopt the auction-style bidding mechanism without significant resistance.
- Impact: If users find the bidding process confusing or inconvenient, it could affect user satisfaction, impacting the success of TicketBuddy.

5. Regulatory Compliance:

- Assumption: TicketBuddy complies with all relevant regulations and legal requirements related to ticket sales, auctions, and online transactions.
- Impact: Non-compliance could lead to legal consequences, fines, and reputational damage.

6. Development Team Expertise:

- Assumption: The development team possesses the necessary skills and expertise to implement all features and functionalities outlined in the SRS.
- Impact: Lack of expertise or resources could result in delays, quality issues, and incomplete implementation of requirements.

2.3 External Interface Requirements

2.3.1 User Interfaces

1. User Interface Design:

- The user interface of TicketBuddy will follow modern design principles, with a clean and intuitive layout to enhance user experience.

- GUI standards and design guidelines will be followed to ensure consistency and familiarity for users across all screens.
- Screens will be designed with responsive layouts to accommodate various screen sizes and devices.

2. Screen Layout Constraints:

- Each screen will have a consistent layout with a header containing the Ticket-Buddy logo and navigation menu, a main content area, and a footer with links to relevant pages.
- Content will be organized hierarchically, with clear visual hierarchy to guide users' attention to important elements.
- Elements such as buttons, input fields, and dropdown menus will be appropriately spaced and sized for ease of use.

3. Standard Buttons and Functions:

- Each screen will include standard buttons for primary actions such as "Submit", "Cancel", and "Back".
- Navigation elements such as breadcrumbs or a navigation bar will be present to help users easily navigate between screens.
- A "Help" button or link will be available on every screen to provide users with access to help documentation or support resources.

4. Keyboard Shortcuts:

- Keyboard shortcuts will be implemented for common actions to improve efficiency for power users.
- Shortcuts will be clearly documented and accessible to users through tooltips or keyboard shortcut guides.

5. Error Message Display Standards:

- Error messages will be displayed inline with form fields or in a prominent location on the screen.

- Messages will be clear, concise, and descriptive, providing users with actionable information to resolve issues.
- Error messages will include instructions on how to correct the error and may include links to relevant help resources.

6. Software Components Requiring User Interface:

- Ticket Listing Management: Interface for sellers to add, update, and remove ticket listings.
- Bid Management: Interface for users to place bids on ticket listings and view bid statuses.
- Ticket Return Management: Interface for users to initiate ticket returns and view return status.
- Notification Management: Interface for users to manage notification preferences and view alerts.

2.3.2 Hardware Interfaces

1. Supported Device Types:

- TicketBuddy will be accessible on various hardware platforms, including:
 - Desktop computers: Windows, macOS, Linux
 - Mobile devices: iOS, Android
 - Web browsers: Chrome, Firefox, Safari, Edge

2. Nature of Data and Control Interactions:

- Data Interactions: The software interacts with hardware components to send and receive data related to user interactions, such as ticket purchases, bid placements, and notification delivery.
- Control Interactions: The software controls hardware components to perform tasks such as displaying user interface elements, processing user inputs, and communicating with external systems.

3. Communication Protocols:

- HTTP/HTTPS: Used for communication between the software client (e.g., web browser, mobile app) and the TicketBuddy server.
- WebSocket: Utilized for real-time bid updates, notifications, and other asynchronous communication between the client and server.
- RESTful API: Provides a standardized interface for communication between TicketBuddy and external systems, allowing integration with third-party services and APIs.
- TCP/IP: Facilitates communication between hardware components (e.g., server, database) within the system architecture.

4. Physical Characteristics:

- The software interacts with hardware components such as:
 - Servers: Hosting the TicketBuddy application, databases, and other backend services.
 - Network Infrastructure: Including routers, switches, and firewalls for data transmission between clients and servers.
 - Client Devices: Desktop computers, laptops, smartphones, and tablets used by users to access TicketBuddy.

5. Supported Input and Output Devices:

- Input Devices: Keyboards, mice, touchscreens, and other input devices used by users to interact with TicketBuddy.
- Output Devices: Displays (e.g., monitors, screens) used to present the TicketBuddy user interface to users, as well as printers for generating tickets and receipts.

6. Data Transmission and Storage:

- Data is transmitted between hardware components using secure communication protocols to ensure data integrity and confidentiality.
- Data is stored securely in databases and other storage systems, with appropriate backup and disaster recovery mechanisms in place to prevent data loss.

2.3.3 Software Interfaces

1. Database Management System (DBMS):

- PostgreSQL (version 13)
- Data Items:
 - User information (e.g., username, email, password hash)
 - Ticket listings (e.g., event name, price, quantity available)
 - Bid information (e.g., bidder ID, bid amount, timestamp)
 - Transaction records (e.g., purchase history, payment details)
- Purpose:
 - PostgreSQL is used as the backend database to store and retrieve various data entities required by TicketBuddy.
- Communication:
 - TicketBuddy backend communicates with PostgreSQL using SQL queries over the PostgreSQL protocol (TCP/IP).

2. Frontend Technologies:

- HTML, CSS, JavaScript (React.js)
- Data Items:
 - User interface elements (HTML/CSS)
 - User interactions and state management (JavaScript/React.js)
- Purpose:
 - Frontend technologies are used to create the user interface of TicketBuddy, allowing users to interact with the application through web browsers.
- Communication:
 - Frontend interacts with the backend via HTTP requests to fetch and update data.

3. Backend Framework:

- Python (version 3.9) with Flask or Django framework
- Data Items:
 - Business logic and application workflow
 - HTTP request handling and routing
 - Data processing and manipulation
- Purpose:
 - Python backend handles the business logic of TicketBuddy, including user authentication, ticket management, bid processing, and transaction handling.
- Communication:
 - Backend communicates with frontend and database using HTTP requests and responses, processing data and generating appropriate responses.

4. Integrated Commercial Components:

- As of now, TicketBuddy, being in its initial stages of development, does not integrate any third-party services.
- Future Integrations: If TicketBuddy integrates with any third-party services or APIs in the upcoming future (e.g., payment gateways, email services), they would be listed here along with the data items exchanged and the purpose of integration.

2.3.4 Communication Interfaces

1. Email Communication:

- Requirement: The system must support email notifications for various events such as bid updates, ticket purchases, and password resets.
- Message Formatting: Email notifications should include relevant information such as event details, bid status, transaction summaries, and instructions for action.
- Communication Protocol: SMTP (Simple Mail Transfer Protocol) will be used for sending email notifications.

- Security/Encryption: Email communication should be encrypted using TLS (Transport Layer Security) to ensure the confidentiality of email contents.
- Data Transfer Rates: Email delivery should occur within a reasonable time-frame, typically within a few seconds to a few minutes.
- Synchronization: Email delivery and receipt should be synchronized with system events to ensure timely notifications.

2. Web Browser Communication:

- Requirement: The system must support communication with web browsers for accessing the TicketBuddy web application.
- Message Formatting: Web pages should be formatted using HTML, CSS, and JavaScript to provide an interactive and visually appealing user interface.
- Communication Protocol: HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure) will be used for communication between web browsers and the TicketBuddy server.
- Security/Encryption: HTTPS should be enforced to encrypt data transmitted between the web browser and server, preventing eavesdropping and tampering.
- Data Transfer Rates: Web pages should load quickly to provide a responsive user experience, with minimal latency in data transmission.
- Synchronization: Client-side scripts and server-side logic should be synchronized to ensure consistent behavior and data integrity.

3. Network Server Communication:

- Requirement: The system must support communication between network servers for data transmission and synchronization.
- Communication Protocol: TCP/IP (Transmission Control Protocol/Internet Protocol) will be used for reliable and efficient communication between network servers.
- Security/Encryption: Data transmitted between servers should be encrypted using TLS to protect against unauthorized access and data interception.

- Data Transfer Rates: Network server communication should occur at high speeds to support real-time data synchronization and processing.
- Synchronization: Synchronization mechanisms such as data replication, message queues, or distributed databases may be employed to ensure consistency across network servers.

4. Electronic Forms:

- Requirement: The system must support electronic forms for collecting user input and submitting data.
- Message Formatting: Form fields should be formatted using HTML or a similar markup language, with validation rules to ensure data integrity and completeness.
- Communication Protocol: Form submissions will be transmitted using HTTP POST requests to the server.
- Security/Encryption: Form data should be transmitted over HTTPS to protect sensitive information such as passwords or payment details.
- Data Transfer Rates: Form submissions should be processed quickly, with minimal delay in data transmission.
- Synchronization: Form submissions should be synchronized with backend databases or systems to ensure data consistency and availability.

2.4 System Features

2.4.1 Ticket Listing Management

1. Description and Priority:

- This feature involves managing the listing of tickets on TicketBuddy, including adding, updating, and removing ticket listings. It is of high priority as it forms the core functionality of the platform.

- Priority Components:

Table 2.1: Ticket Listing Management Priority Components

| Component | Value |
|------------------|--------------|
| Benefit | 9 |
| Penalty | 3 |
| Cost | 5 |
| Risk | 7 |

2. Stimulus/Response Sequences:

- Stimulus: Seller initiates the process to contact TicketBuddy to list tickets.
Response: System prompts the seller to enter information about the event and contact details.
- Stimulus: Seller initiates the process to contact TicketBuddy to edit tickets.
Response: System prompts the seller to enter information about the event and contact details.
- Stimulus: Seller initiates the process to contact TicketBuddy to remove tickets.
Response: System prompts the seller to enter information about the event and contact details.

3. Functional Requirements:

Table 2.2: Ticket Listing Management Requirements

| Requirement ID | Requirement Description |
|----------------|--|
| REQ-1 | The system shall provide a user interface for sellers to contact the developers to bring about a potential partnership where the app sells the tickets to the venue. |
| REQ-2 | The system shall validate input data for new ticket listings to ensure accuracy and completeness. |
| REQ-3 | The system shall allow sellers to update existing ticket listings, enabling modification of ticket details such as price, quantity, and auction mode availability. |
| REQ-4 | The system shall enforce validation checks when updating ticket listings to prevent invalid data entry. |
| REQ-5 | The system shall provide sellers with the option to remove ticket listings from the platform. |
| REQ-6 | Upon removal of a ticket listing, the system shall confirm the action with the seller before permanently deleting the listing. |
| REQ-7 | The system shall update the platform interface to reflect changes in ticket listings in real-time for users browsing available tickets. |
| REQ-8 | The system shall generate notifications to alert users when new ticket listings are added or existing listings are updated or removed. |

2.4.2 Bid Management

1. Description and Priority:

- This feature involves managing bids placed by users on TicketBuddy, including processing bids, updating bid statuses, and notifying users of bid outcomes. It is of high priority as it directly impacts the auction mode functionality.

- Priority Components:

Table 2.3: Bid Management Priority Components

| Component | Value |
|------------------|--------------|
| Benefit | 8 |
| Penalty | 4 |
| Cost | 6 |
| Risk | 7 |

2. Stimulus/Response Sequences:

- Stimulus: User places a bid on a ticket listing.

Response: System updates the bid status and position in the waitlist accordingly.

- Stimulus: Auction mode is activated due to all tickets being waitlisted.

Response: System initiates bid processing, determining the highest bidders for each ticket listing.

- Stimulus: Bid time limit expires for a ticket listing.

Response: System finalizes ticket allocation based on the highest bidders and notifies successful bidders.

3. Functional Requirements:

Table 2.4: Bid Management Requirements

| Requirement | Description |
|-------------|--|
| REQ-1 | The system shall record bids placed by users, including bid amount and timestamp. |
| REQ-2 | The system shall update the bid status and position in the waitlist for each user based on bid amount and time. |
| REQ-3 | When auction mode is activated, the system shall process all pending bids to determine the highest bidders for each ticket listing. |
| REQ-4 | The system shall enforce bid time limits and automatically finalize ticket allocation once the time limit expires. |
| REQ-5 | The system shall notify users of bid outcomes, including successful allocation or being outbid, via in-app notifications and/or email. |

2.4.3 Ticket Return Management

1. Description and Priority:

- This feature involves managing ticket returns initiated by users, including processing returns, updating waitlists, and issuing refunds. It is of medium priority as it supports user flexibility and ensures a smooth ticketing experience.
- Priority Components:

Table 2.5: Ticket Return Management Priority Components

| Component | Value |
|-----------|-------|
| Benefit | 7 |
| Penalty | 5 |
| Cost | 5 |
| Risk | 6 |

2. Stimulus/Response Sequences:

- Stimulus: User requests to return a purchased ticket.
Response: System initiates the return process, collecting a cancellation fee and updating the waitlist accordingly.
- Stimulus: Secondary waitlist customer is upgraded due to a primary waitlist return.
Response: System updates the waitlist positions and notifies affected users of their new status.
- Stimulus: User fails to respond to a ticket reclaim deadline.
Response: System removes the user from both waitlists and reallocates the ticket to the next eligible bidder.

3. Functional Requirements:

Table 2.6: Ticket Return Management Requirements

| Requirement | Description |
|-------------|---|
| REQ-1 | The system shall provide a user interface for users to initiate ticket returns, including options for reason selection and confirmation. |
| REQ-2 | The system shall collect a cancellation fee from users returning tickets, deducting it from the refund amount if applicable. |
| REQ-3 | Upon ticket return, the system shall update the waitlist positions, moving users below the returnee up by one position. |
| REQ-4 | The system shall enforce deadlines for ticket reclaim by users in the secondary waitlist and automatically reallocate tickets if deadlines are not met. |
| REQ-5 | The system shall issue refunds to users returning tickets, processing refunds to the original payment method within a specified timeframe. |

2.5 Other Nonfunctional Requirements

2.5.1 Performance Requirements

1. Real-Time Bid Processing:

- Requirement: The system should process bids and update bid statuses in real-time, with a maximum latency of 1 second.
- Rationale: Real-time bid processing ensures a responsive and dynamic auction experience for users. Delays in processing bids could lead to frustration among users and impact the fairness of the auction process.

2. Scalability of Ticket Listing Management:

- Requirement: The system should be able to handle concurrent requests for adding, updating, or removing ticket listings from multiple sellers without performance degradation.
- Rationale: As TicketBuddy may experience high traffic during peak times or when popular events are listed, the system must scale horizontally to accommodate increased demand without compromising performance.

3. High Availability for Ticket Purchasing:

- Requirement: The system should maintain at least 99.9% uptime for ticket purchasing functionality.
- Rationale: Users expect to access TicketBuddy at any time to purchase tickets, especially during critical periods such as when auction mode is active. Ensuring high availability minimizes service disruptions and maximizes user satisfaction.

4. Optimized Bid Matching Algorithm:

- Requirement: The system should employ an efficient bid matching algorithm capable of handling large volumes of bids within a short timeframe.
- Rationale: During auction mode, the system must quickly identify the highest bidders for each ticket listing to finalize allocations promptly. An optimized

bid matching algorithm reduces processing time and improves overall system performance.

5. Fast Ticket Return Processing:

- Requirement: Ticket returns should be processed within 30 seconds of initiation.
- Rationale: Users may need to return tickets for various reasons, and delays in processing returns could impact their experience negatively. Fast ticket return processing ensures prompt resolution and maintains user trust in the platform.

6. Low Latency Notifications:

- Requirement: Notifications regarding bid outcomes, ticket updates, or system events should be delivered to users within 5 seconds of the triggering event.
- Rationale: Timely notifications keep users informed about the status of their bids, ticket listings, and other relevant activities. Low latency ensures that users receive updates promptly, enhancing their engagement with the platform.

2.5.2 Safety Requirements

1. Data Security and Privacy:

- Requirement: The system must adhere to industry-standard encryption protocols to protect sensitive user information, including personal and financial data.
- Safeguard: Implement secure data transmission and storage practices to prevent unauthorized access or data breaches.
- External Policy/Regulation: Compliance with data protection regulations.
- Safety Certification: The system must obtain relevant certifications such as ISO 27001 for information security management to demonstrate compliance with industry best practices.

2. Prevention of Unauthorized Access:

- Requirement: The system should implement robust authentication and authorization mechanisms to prevent unauthorized access to user accounts and administrative functions.
- Safeguard: Enforce strong password policies, multi-factor authentication, and session management controls to safeguard user accounts.
- External Policy/Regulation: Adherence to cybersecurity standards and regulations such as NIST SP 800-63 for digital identity guidelines ensures protection against unauthorized access.
- Safety Certification: Compliance with cybersecurity certifications like SOC 2 demonstrates the system's commitment to safeguarding user access.

3. Financial Transaction Security:

- Requirement: All financial transactions, including ticket purchases and refunds, must be conducted securely using PCI DSS-compliant payment processing systems.
- Safeguard: Employ tokenization and encryption techniques to secure payment data during transmission and storage.
- External Policy/Regulation: Compliance with the Payment Card Industry Data Security Standard (PCI DSS) is essential for protecting payment card data and preventing fraud.
- Safety Certification: PCI DSS certification is necessary to ensure that the system meets industry standards for secure payment processing.

4. Resilience to Cyberattacks:

- Requirement: The system should have mechanisms in place to detect, mitigate, and respond to cybersecurity threats such as DDoS attacks, malware, and phishing attempts.
- Safeguard: Implement intrusion detection systems, firewalls, and continuous monitoring tools to identify and mitigate security breaches promptly.

- External Policy/Regulation: Compliance with cybersecurity frameworks such as NIST Cybersecurity Framework or ISO 27001 helps mitigate risks associated with cyberattacks.
- Safety Certification: Certification under cybersecurity standards like SOC 2 or ISO 27001 demonstrates the system's resilience to cyber threats and adherence to best practices.

2.5.3 Security Requirements

1. User Identity Authentication:

- Requirement: The system must implement robust user identity authentication mechanisms, including password-based authentication and optional multi-factor authentication.
- External Policy/Regulation: Compliance with regulatory frameworks such as GDPR, and CCPA mandates strong authentication controls to protect user privacy.
- Security Certification: Obtaining certifications like ISO 27001 or SOC 2 Type II requires demonstrating effective user authentication mechanisms to ensure the security of user identities and data.

2. Data Encryption in Transit and at Rest:

- Requirement: All sensitive data transmitted between the client and server, as well as data stored on servers, must be encrypted using industry-standard encryption protocols.
- External Policy/Regulation: Compliance with data protection regulations such as GDPR requires encryption of sensitive data to prevent unauthorized access and ensure privacy.
- Security Certification: Achieving certifications like ISO 27001 or SOC 2 Type II necessitates the implementation of encryption protocols to protect data confidentiality.

3. Access Control and Authorization:

- Requirement: The system should enforce strict access control policies to limit user access to data and functionality based on their roles and permissions.
- External Policy/Regulation: Adherence to regulatory standards like GDPR and CCPA requires implementing access controls to protect user privacy and prevent unauthorized data access.
- Security Certification: Compliance with certifications such as ISO 27001 or SOC 2 Type II necessitates the implementation of access control mechanisms to ensure data security and privacy.

4. Secure Payment Processing:

- Requirement: Payment processing systems integrated with the platform must comply with PCI DSS standards for handling payment card data securely.
- External Policy/Regulation: Compliance with PCI DSS regulations is mandatory to protect payment card data and prevent fraud during payment transactions.
- Security Certification: Obtaining PCI DSS certification for payment processing systems ensures adherence to industry standards for secure handling of financial data.

5. Data Breach Response Plan:

- Requirement: The system should have a documented data breach response plan outlining procedures for detecting, responding to, and mitigating data breaches.
- External Policy/Regulation: Compliance with regulations such as GDPR and CCPA mandates having a data breach response plan to minimize the impact of data breaches on user privacy.
- Security Certification: Certification under ISO 27001 or SOC 2 Type II requires having a data breach response plan as part of the organization's information security management system.

2.5.4 Software Quality Attributes

1. Usability:

- Requirement: The system interface must be intuitive and user-friendly, with an average user satisfaction rating of at least 4.5 out of 5.
- Verification: Conduct usability testing with representative users and measure satisfaction scores through surveys or feedback forms.

2. Reliability:

- Requirement: The system must achieve a minimum uptime of 99.9% over a 12-month period, with no more than 0.1% downtime for maintenance or unplanned outages.
- Verification: Monitor system uptime and downtime using logging and monitoring tools, and track service level agreements (SLAs) with hosting providers.

3. Maintainability:

- Requirement: The system codebase must adhere to coding standards and best practices, with at least 80% code coverage in unit tests.
- Verification: Conduct code reviews and static code analysis to ensure adherence to coding standards, and use code coverage tools to measure test coverage.

4. Scalability:

- Requirement: The system should scale to accommodate a 200% increase in concurrent user traffic without degradation in performance.
- Verification: Perform load testing using simulated traffic to evaluate system performance under increased loads, and monitor resource utilization during peak usage periods.

5. Robustness:

- Requirement: The system should gracefully handle errors and exceptions, with no more than 0.5% of user interactions resulting in critical failures.

- Verification: Conduct stress testing and error injection tests to assess the system's resilience to failures, and monitor error rates in production environments.

6. Interoperability:

- Requirement: The system must support integration with third-party services and APIs using industry-standard protocols such as REST or GraphQL.
- Verification: Validate interoperability through testing with various third-party services and APIs, ensuring compatibility and seamless data exchange.

7. Testability:

- Requirement: The system must have automated test suites covering at least 80% of code functionality, including unit tests, integration tests, and end-to-end tests.
- Verification: Measure test coverage using code coverage tools, and ensure that test suites provide comprehensive coverage of system functionality and use cases.

8. Adaptability:

- Requirement: The system architecture must support easy adaptation to changing business requirements and technological advancements, with a minimum of 80% of features adaptable within a 3-month timeframe.
- Verification: Evaluate system architecture and design patterns for flexibility and extensibility, and track the time and effort required to implement changes or new features.

Chapter 3

System Architecture and Design

3.1 System Overview

TicketBuddy is designed as a comprehensive platform to streamline the buying, selling, and management of event tickets, aiming to create a harmonious ecosystem for both sellers and buyers. Beyond simplifying ticket transactions, TicketBuddy targets dual profitability, offering sellers an avenue to maximize profits while providing buyers with fair deals. This dual-profitability model underscores TicketBuddy's commitment to creating a balanced marketplace where both parties can thrive. The platform caters to individual ticket buyers, professional sellers, and event organizers, ensuring a multifaceted solution to meet diverse needs.

TicketBuddy's architecture is built on a foundation of reliability and scalability. The backend operations utilize Python (version 3.9) with Flask, while the frontend is powered by HTML, CSS, and JavaScript with React.js. This technology stack ensures a responsive user interface and efficient data processing. Data management is entrusted to PostgreSQL (version 13), a robust and scalable database system. The architecture is designed to be modular and extensible, enabling seamless integration with third-party services and adaptability to future technologies.

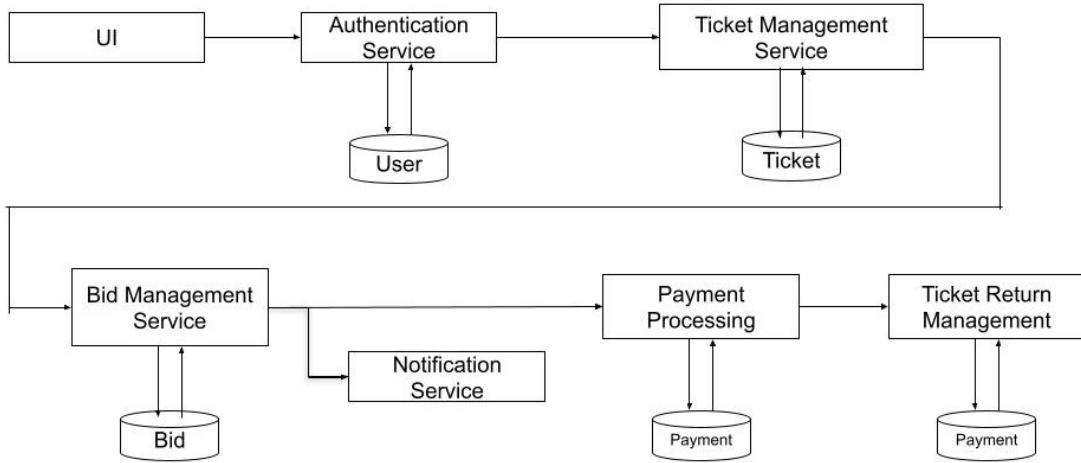


Figure 3.1: System Architecture Diagram

- Authentication Service: Ensures secure user access through robust mechanisms.
- Ticket Management Service: Enables real-time ticket listing and updates.
- Bid Management Service: Facilitates fair bidding with real-time processing.
- Notification Service: Delivers timely updates on bidding outcomes and ticket availability.
- Payment Processing: Ensures secure and compliant transactions.
- Ticket Return Management: Streamlines ticket returns and refunds.
- User Interface: Provides an intuitive and accessible design.

Security is a top priority for TicketBuddy, employing robust encryption protocols and compliance measures to protect user data and platform integrity. The platform relies on PostgreSQL for efficient data storage and management, reinforcing data security, integrity, and reliability. Backup and recovery procedures are also in place to ensure business continuity.

TicketBuddy is engineered for high performance and scalability. Features like real-time bid processing and low-latency notifications enhance user satisfaction. The architecture's

scalability ensures adaptability to growing demands, maintaining optimal performance under varying workloads.

3.2 Architectural Design

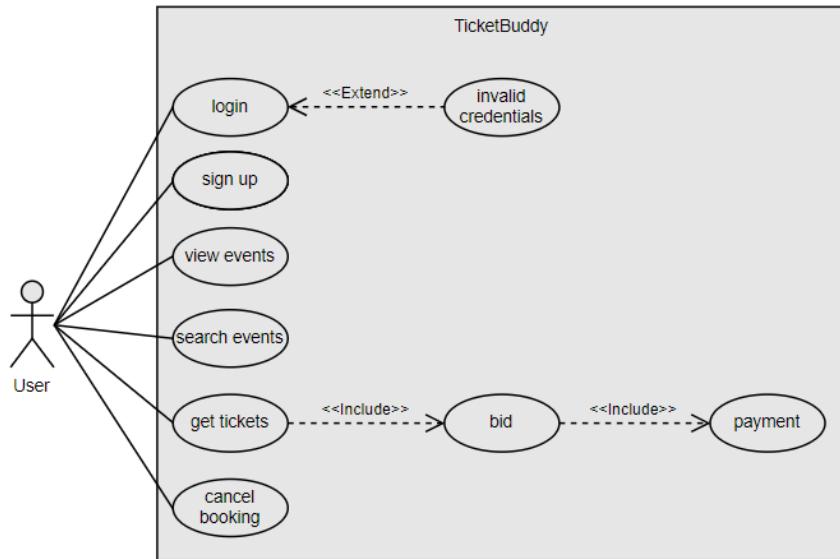


Figure 3.2: Use Case Diagram

3.3 Proposed Methodology/Algorithms

ALGORITHM: User Authentication

- STEP 1:** Validate user credentials during login.
- STEP 2:** Generate and verify tokens for session management.
- STEP 3:** Implement multi-factor authentication for enhanced security.

ALGORITHM: Ticket Listing Management

- STEP 1:** Seller initiates contact to list tickets.
- STEP 2:** System prompts for event details and contact information.
- STEP 3:** Validate and store the ticket listing in the Ticket table.
- STEP 4:** Update the platform interface in real-time to reflect the new listing.
- STEP 5:** Generate notifications for users about the new listing.

ALGORITHM: Seat Selection

- STEP 1:** Display available seats based on selected event and venue.
- STEP 2:** Allow users to select seats or sections.
- STEP 3:** Update seat and section statuses upon selection.
- STEP 4:** Calculate and display ticket price based on seat selection and quantity.

ALGORITHM: Bid Management

- STEP 1:** User places a bid on a ticket listing.
- STEP 2:** Record the bid details in the Bid table.
- STEP 3:** Update bid status and waitlist position based on bid amount.
- STEP 4:** Activate auction mode if all tickets are waitlisted.
- STEP 5:** Finalize ticket allocation when bid time limit expires.

ALGORITHM: Real-Time Bid Processing

- STEP 1:** Capture and record bids in real-time from users.
- STEP 2:** Update bid statuses and waitlist positions immediately.
- STEP 3:** Activate auction mode when necessary.
- STEP 4:** Finalize ticket allocations promptly based on highest bids.

ALGORITHM: Ticket Return Management

- STEP 1:** User requests to return a purchased ticket.
- STEP 2:** Initiate return process, collect cancellation fee, and update waitlist.
- STEP 3:** Upgrade secondary waitlist customers based on primary waitlist returns.
- STEP 4:** Remove user from waitlists if they fail to reclaim tickets by deadline.
- STEP 5:** Issue refund to the user within a specified timeframe.

3.4 User Interface Design

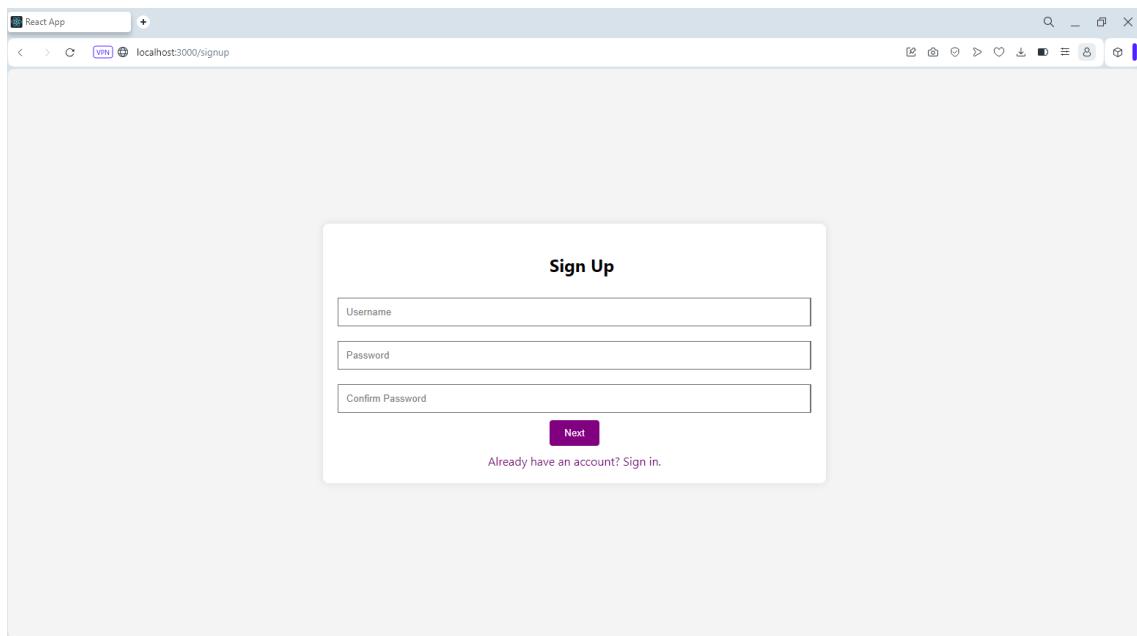


Figure 3.3: Sign Up Page

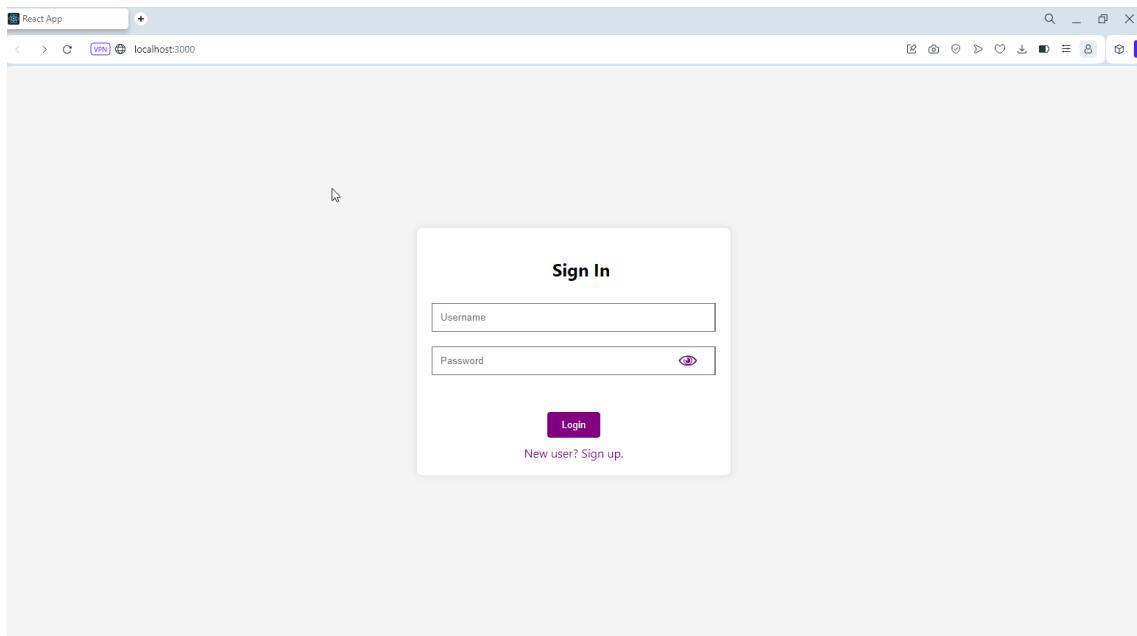


Figure 3.4: Sign In Page

A screenshot of the TicketBuddy website. The header features a dark blue navigation bar with the "TicketBuddy" logo, a search bar, and a green "Find Tickets" button. Below the header, the page is titled "Featured Events" and displays four event cards. The first three cards are fully visible, while the fourth card is partially visible at the bottom. Each card includes a thumbnail image, the title "Example Event", the genre "Example Genre", and a "Get Tickets" button. The thumbnails show various performers: a man in sunglasses, a man singing into a microphone, a man in a turban, and a man in a purple suit.

Figure 3.5: Home Page

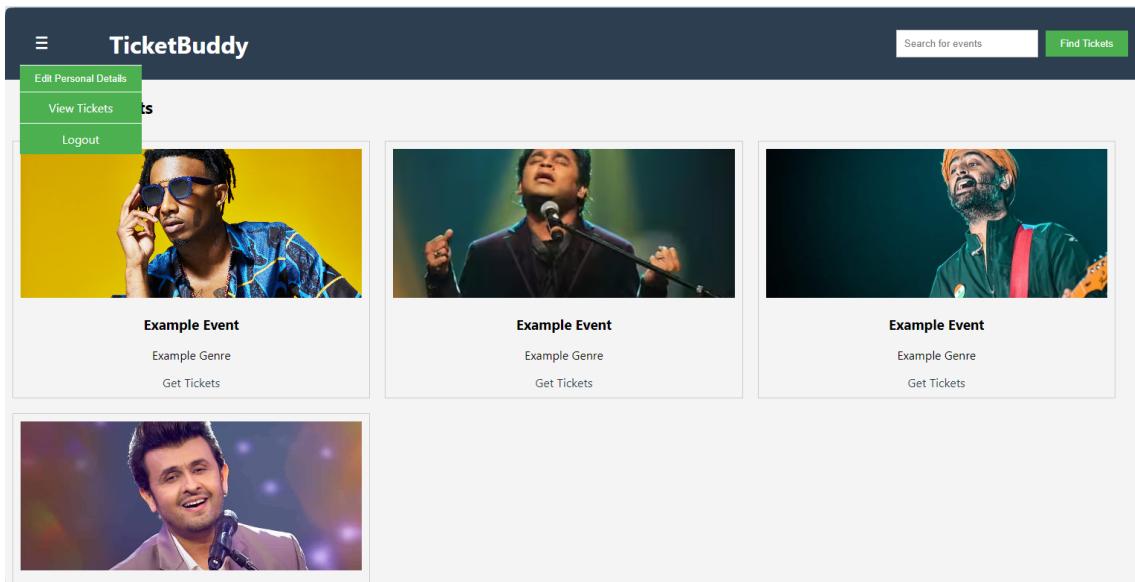


Figure 3.6: Home Page Dropdown Menu

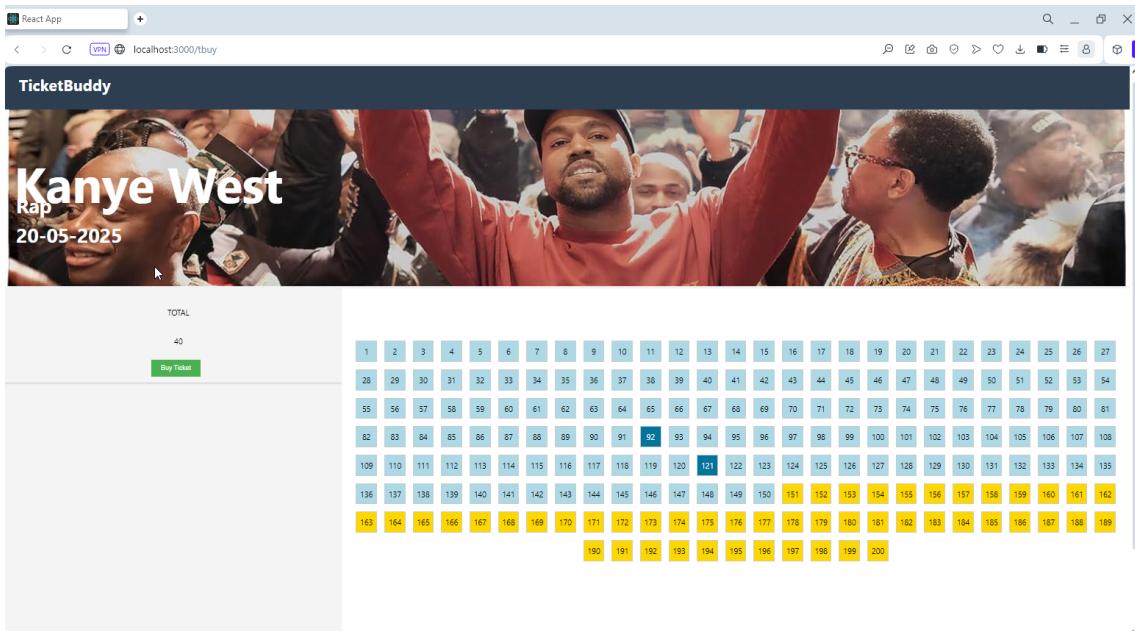


Figure 3.7: Booking Page

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/tbuy". The main content is a dark-themed interface for a ticket bidding application. At the top, it displays "TicketBuddy" and a large image of a crowd at a concert. Overlaid on the image is the text "Kanye West" in large white letters, with "Rap" written below it, and the date "20-05-2025". To the left of the image is a table showing current bids:

| Position | Bid Amount | User ID |
|----------|------------|---------|
| 1 | 100 | 1 |
| 2 | 48 | 3 |
| 3 | 10 | 2 |

To the right of the bids is a "MAKE BID" form with a "Bid Amount" input field and a green "Confirm" button.

Figure 3.8: Bidding Page

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/view_tickets". The main content is a dark-themed interface for viewing tickets. It displays "TicketBuddy" at the top and "YOUR TICKETS" below it. A list of three tickets is shown, each with details and a "Cancel" button:

| | | | |
|---------------|------------|----------|--------|
| Playboi Carti | 12-06-2024 | #A0034B1 | Cancel |
| Playboi Carti | 12-06-2024 | #A0034B1 | Cancel |
| Playboi Carti | 12-06-2024 | #A0034B1 | Cancel |

Figure 3.9: Your Tickets Page

3.5 Database Design

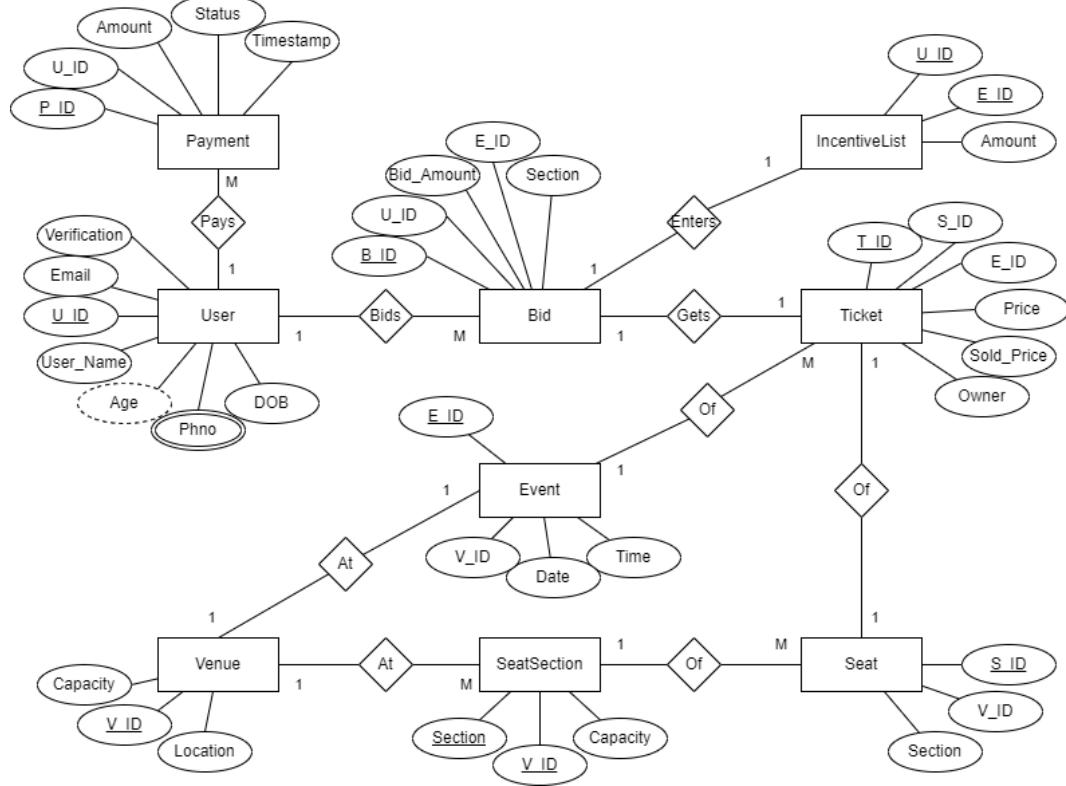


Figure 3.10: ER Diagram

- **User Table:** The User table manages registered users with fields like user ID, user name, email, phone no., and more, facilitating user management, personalized experiences, and secure authentication.
- **Event Table:** The Event table stores event details including event ID, name, date, and venue, facilitating event management, linking tickets to events, and showcasing events to users.
- **Venue Table:** The Venue table provides venue details such as venue ID, name, address, and seating capacity, offering location information for events and supporting event planning.
- **Seat Section Table:** The Seat Section table categorizes venue seats into sections

with section ID, venue, section name, and capacity, organizing venue layouts and providing categorized seating options.

- Seat Table: The Seat table tracks seats or sections within venues with seat ID, venue, section, and status, enabling seat selection during ticket purchase and managing venue capacity.
- Ticket Table: The Ticket table displays tickets for sale with ticket ID, event ID, seller ID, price, and status, managing ticket listings and enabling users to browse and purchase tickets.
- Bid Table: The Bid table logs bids on ticket listings with bid ID, user, ticket, bid amount, and timestamp, supporting the auction process and ensuring transparency in determining highest bidders.
- Payment Table: The Payment table captures financial transactions on TicketBuddy, detailing transaction ID, user, amount, payment method, and timestamp, ensuring financial accountability and supporting refunds.

This database design ensures efficient data storage, retrieval, and management for TicketBuddy, supporting its core functionalities of ticket listing, bid management, and ticket returns. The normalization of tables reduces data redundancy and ensures data integrity, while the defined relationships between tables facilitate complex queries and operations.

3.6 Description of Implementation Strategies

In the frontend development using React.js, we utilize several key libraries to enhance user interface and interaction. Styled Components allows us to craft styled React components with CSS-in-JS, providing a maintainable and scalable approach to styling. Axios serves as our HTTP client, simplifying API calls with its promise-based interface. This ensures streamlined request and response handling, error management, and data transformations. To design responsive and aesthetically pleasing interfaces, we integrate React Bootstrap, which offers pre-designed Bootstrap components as React components. React Icons further enriches the user experience by providing popular icon packs as React components, streamlining the integration of icons into our application. Meanwhile, Material-UI,

implementing Google’s Material Design, offers a set of customizable and accessible components, providing a consistent and visually appealing UI. Additionally, Next.js Commerce extends Next.js with headless commerce capabilities, offering pre-built components and integrations with popular eCommerce platforms like Shopify and BigCommerce.

On the backend, Flask serves as the foundation for our web application, providing a lightweight and flexible micro-framework for building web services in Python. We integrate Flask-SQLAlchemy to simplify database interactions, benefiting from SQLAlchemy’s powerful ORM for efficient data management and querying. PostgreSQL is our chosen database, offering robustness, reliability, and ACID compliance for secure and scalable data storage. To handle Cross-Origin Resource Sharing (CORS) and facilitate secure communication between frontend and backend, we employ Flask-CORS. For PostgreSQL database access, we utilize Psycopg2, a PostgreSQL adapter for Python, enabling efficient and secure database interactions. Lastly, Python-dotenv allows us to manage configuration settings and sensitive information like database credentials and API keys without hardcoding them in the application code.

With these technologies and tools, TicketBuddy is poised to be a scalable, secure, and user-friendly platform. Styled Components, Axios, and React Bootstrap enhance frontend development, while Flask, Flask-SQLAlchemy, and PostgreSQL form the robust backend foundation. Integration of Flask-CORS and Psycopg2 ensures secure and efficient communication between frontend, backend, and database. Together, these components and strategies pave the way for the development of a feature-rich and reliable ticketing platform.

3.7 Module Division

TicketBuddy integrates several core modules to deliver a seamless and efficient platform:

- Authentication Service: This module ensures secure user access through robust mechanisms, incorporating strong password policies and multi-factor authentication to safeguard user accounts against unauthorized access.
- Ticket Management Service: This service enables real-time listing and updates of

ticket inventory. It supports sellers in managing their listings efficiently while providing buyers with up-to-date information on available tickets and event details.

- Bid Management Service: This module facilitates fair bidding with real-time processing. It records bids with timestamps, automatically allocates tickets to winning bids, and ensures transparency throughout the bidding process.
- Notification Service: The Notification Service delivers timely updates on bidding outcomes, ticket availability, and other important events. It leverages low-latency communication to keep users informed and engaged.
- Payment Processing: This module ensures secure and compliant transactions by integrating with trusted financial systems. It encrypts payment data and adheres to industry standards to protect user financial information.
- Ticket Return Management: This service streamlines the ticket return and refund process, offering fast processing times and seamless integration with the ticket management system. It enables users to initiate returns and refunds with ease, enhancing customer satisfaction.
- User Interface: The User Interface provides an intuitive and accessible design, ensuring an exceptional user experience across devices. It features responsive layouts and accessibility options to cater to a diverse user base, making navigation and interaction effortless.

The development plan follows Nithin John and Thomas T Thayyil for frontend design and development, with Paul Allen Kadayaparambil and Ryan Thomas managing the same for the backend.

3.8 Work Schedule - Gantt Chart



Figure 3.11: Gantt Chart

Chapter 4

Results and Discussions

4.1 Overview

The results of our development efforts culminate in a comprehensive platform known as TicketBuddy. Through meticulous design and implementation, we've achieved a system that seamlessly manages ticket listings, facilitates a clever bidding mechanism, and streamlines ticket returns. With a focus on user experience and robust backend infrastructure, TicketBuddy ensures high performance, scalability, and security.

4.2 Testing

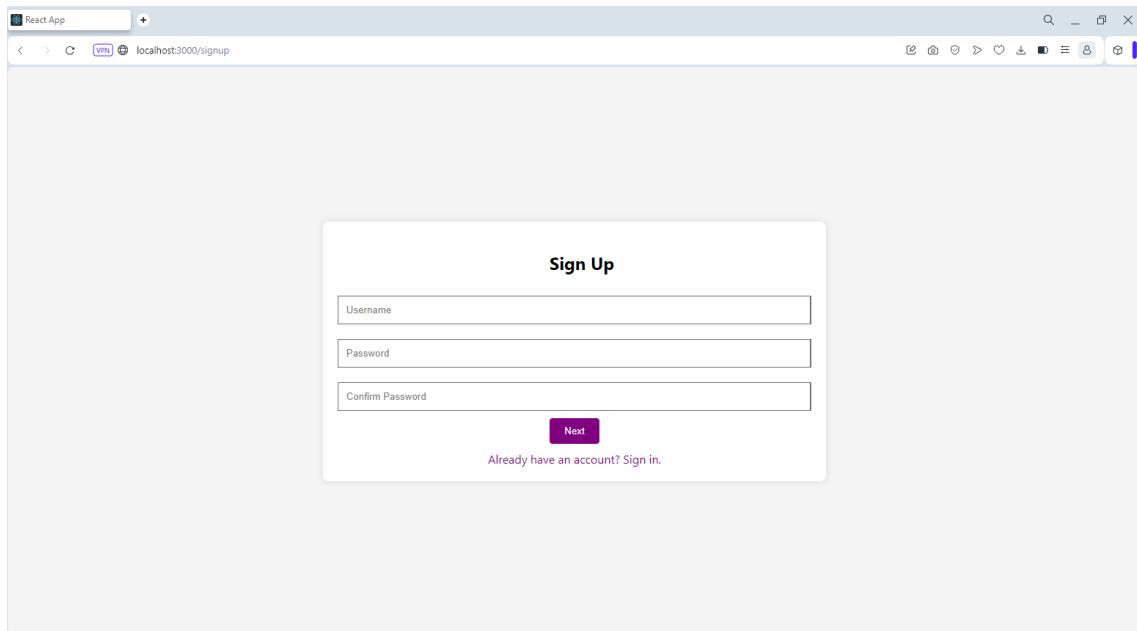


Figure 4.1: Sign Up Page

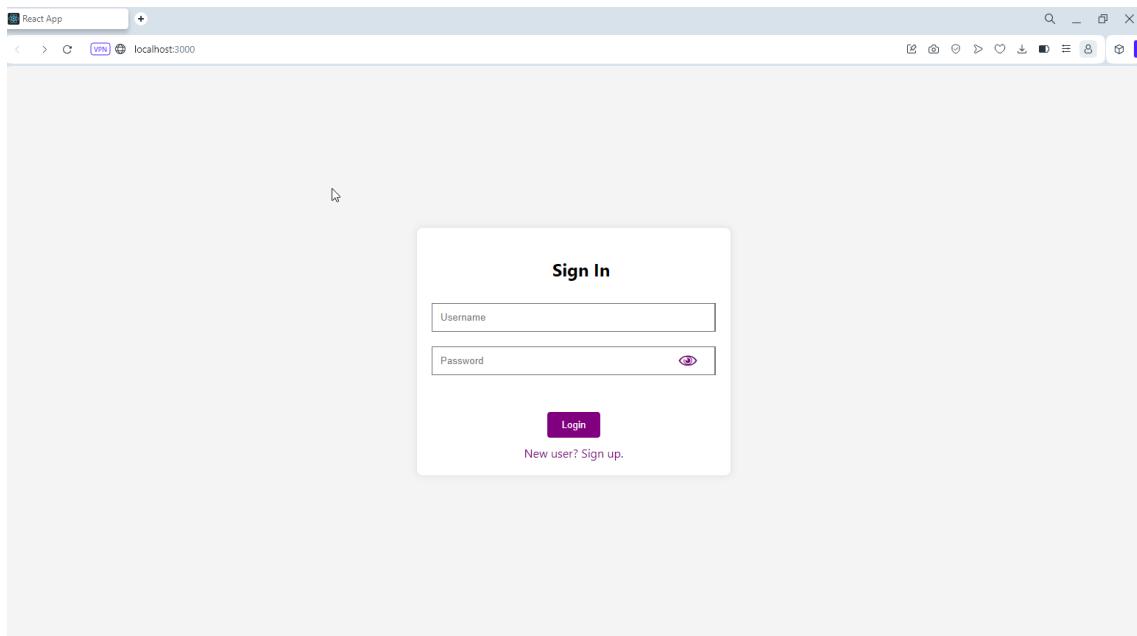


Figure 4.2: Sign In Page

A screenshot of the TicketBuddy website. The header features a dark blue navigation bar with the "TicketBuddy" logo, a search bar, and a green "Find Tickets" button. Below the header, the "Featured Events" section is displayed. It includes four event cards, each with a thumbnail image, the title "Example Event", the genre "Example Genre", and a "Get Tickets" button. The first three cards are fully visible, while the fourth card is partially visible below them.

Figure 4.3: Home Page

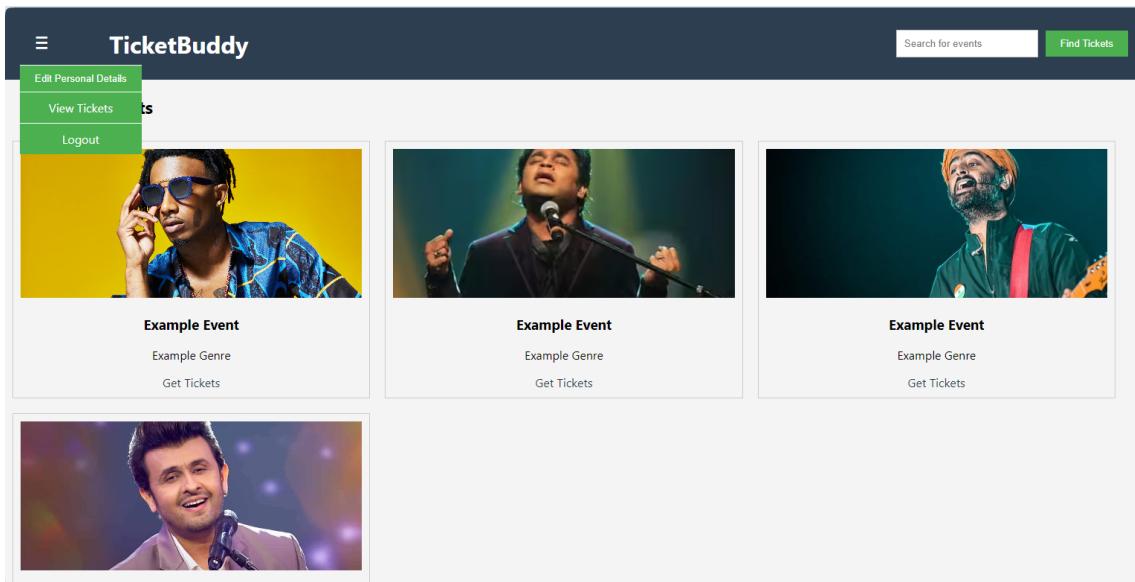


Figure 4.4: Home Page Dropdown Menu

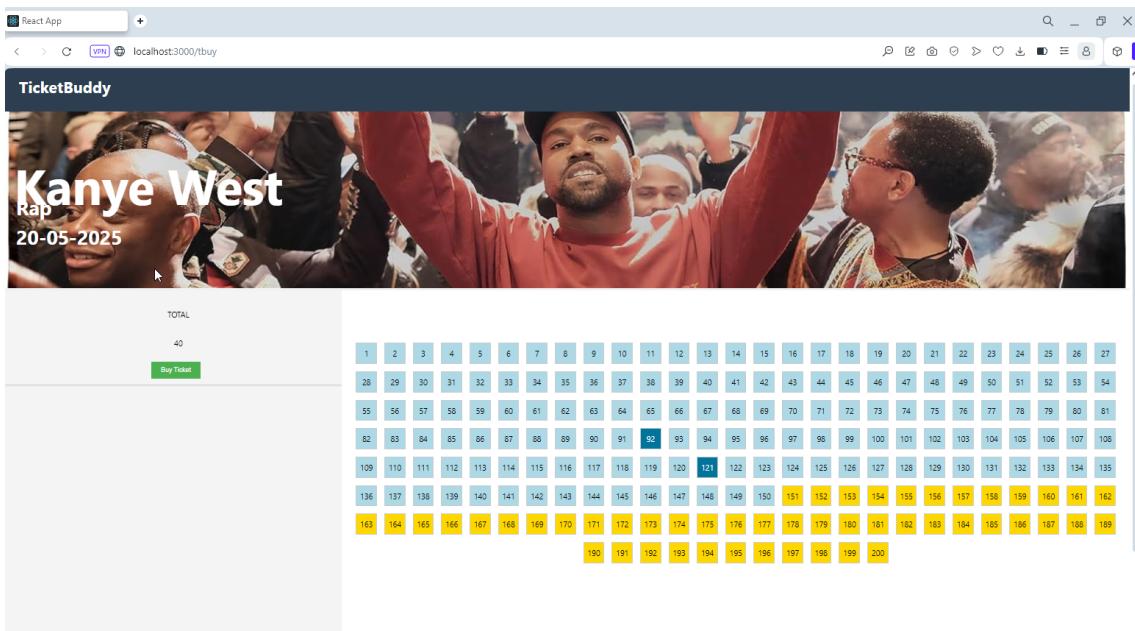


Figure 4.5: Booking Page

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/tbuy". The main content is a dark-themed interface for a ticket bidding application called "TicketBuddy". At the top, there is a large image of a crowd at a concert, with the text "Kanye West" overlaid in large white letters and "Rap" below it. Below the image, the date "20-05-2025" is displayed. To the left of the image, there is a table showing current bids:

| Position | Bid Amount | User ID |
|----------|------------|---------|
| 1 | 100 | 1 |
| 2 | 48 | 3 |
| 3 | 10 | 2 |

To the right of the bids, there is a "MAKE BID" form with a "Bid Amount" input field and a green "Confirm" button.

Figure 4.6: Bidding Page

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/view_tickets". The main content is a dark-themed interface for a ticket management application called "TicketBuddy". At the top, there is a header "TicketBuddy" and a section titled "YOUR TICKETS". Below the section title, there is a table listing three tickets:

| | | | |
|---------------|------------|----------|--------|
| Playboi Carti | 12-06-2024 | #A0034B1 | Cancel |
| Playboi Carti | 12-06-2024 | #A0034B1 | Cancel |
| Playboi Carti | 12-06-2024 | #A0034B1 | Cancel |

Figure 4.7: Your Tickets Page

4.3 Discussion

TicketBuddy has demonstrated significant success in effectively managing ticket listings, facilitating fair bidding processes, and streamlining ticket returns. The platform's intuitive user interface, designed based on best practices and user-centered principles, likely contributes to its positive reception. Emphasizing a good user interface underscores TicketBuddy's commitment to enhancing user experience and ensuring ease of navigation.

A key contributing factor to TicketBuddy's success is its robust backend infrastructure, which ensures high performance, scalability, and security. Leveraging modern technologies and best practices in system architecture and implementation has enabled TicketBuddy to efficiently handle diverse user interactions and data processing requirements. Additionally, the integration of real-time bid processing and low-latency notifications has further enriched the user experience, enabling timely updates and interactions.

While TicketBuddy has shown promising success, it's crucial to recognize potential challenges that may arise in real-life scenarios, particularly during periods of high volume usage. These challenges could include fluctuations in system uptime or delays in notification delivery, which might impact user satisfaction. However, without conducting real-life tests, it's challenging to definitively assess how TicketBuddy would perform under such conditions.

Moving forward, the insights gained from analyzing TicketBuddy's performance will guide future iterations and improvements. By identifying and addressing any areas of concern, such as optimizing system scalability or enhancing notification delivery mechanisms, TicketBuddy can evolve as a leading platform for ticket management and sales.

Chapter 5

Conclusion

5.1 Conclusion

In conclusion, TicketBuddy represents a significant achievement in ticket management systems, offering a versatile platform where users can seamlessly buy and sell tickets for various events. At its core, TicketBuddy's clever bidding mechanism allows sellers to maximize their profits by dynamic bidding, adjusting prices based on demand and other factors. This innovative approach not only benefits sellers but also offers customers the potential to profit through strategic bidding strategies and compensations. Additionally, TicketBuddy streamlines ticket listing management, bid processing, and ticket returns, ensuring a smooth and efficient experience for all users.

5.2 Future Scope

In the future, TicketBuddy could leverage advanced analytics and machine learning algorithms to provide sellers with insights into market trends and optimize pricing strategies in real-time. Expanding services to include features like event reminders, and personalized recommendations could enhance user experience. Additionally, implementing business strategies such as rewards programs, 24/7 live support, and chatbot assistance could foster customer loyalty and provide immediate assistance. Partnering with event organizers and venues would broaden TicketBuddy's reach, offering users access to a wider range of events. Finally, adopting blockchain technology for secure transactions could bolster trust and transparency within the platform ecosystem, positioning TicketBuddy as a reliable and innovative ticketing solution.

Bibliography

- [1] Website Features and their Individual Characteristics to the Context of the Influence of the Network Shopping Acceptance. X. Zhong. January 2021. Advances in Economics, Business and Management Research, volume 203 (pages 11-16)
- [2] A Study on the Conversion mode of Customer Satisfaction and Customer Loyalty in Online Shopping-Taking Taobao, Tmall and JD e-commerce sites as examples. Q. Wang, H. Guo, M. Liu. January 2023. International Journal of social science and humanities(pages 10-29)
- [3] “Future of e-commerce in India”, Chanana Nisha and Goele Sangeeta, March 2016, International Journal of Computing & Business Research(pages 231-246)
- [4] Bidder and Seller Strategies in Online Auctions: A Review, Tsinghua International School, Beijing, China, Audrey Elizabeth Wen, HoiYat Wong, January 2008, Proceedings of the 2021 3rd International Conference on Economic Management and Cultural Industry(pages 1992-1997)
- [5] A Review of Online Auction and It's Pros and Cons, Rashes G Chothani, Nainesh A Patel, Asagarali H Dekavadiya, Punit R Patel, February 2021, International Journal of Advance Engineering and Research Development(pages 8-11)
- [6] ECONOMIC INSIGHTS FROM INTERNET AUCTIONS: A SURVEY, Patrick Bajari and Ali Hortacsu, June 2004, Journal of Economic Literature(pages 457-481)
- [7] Flask Web Framework based News Summarizer: Web Application, Apeksha Kamble, Adnan GaganPooja Wagh, Umar Aziz, Pooja Wagh, December 2022, International Journal of Scientific Research and Engineering Development-- Volume 5 Issue 6(pages 953-956)

Appendix A: Presentation

TicketBuddy

Dr.Uma Narayanan

Nithin John

Paul Allen Kadayaparambil

Ryan Thomas

Thomas T Thayyil

5/17/2024

TicketBuddy

1

Contents

1. Introduction
2. Problem Definition
3. Objectives
4. Scope and Relevance
5. System Design
6. Work Division – Gantt Chart
7. Software/Hardware Requirements
8. Results
9. Conclusion
10. Future Enhancements
11. References

5/17/2024

TicketBuddy

2

Introduction

- Operating in the domain of event ticketing, addressing the complexities of ticket allocation and management while prioritizing enhanced user experience and equitable ticket distribution across various events.
- An innovative and inventive ticket management platform crafted to optimize the process of purchasing, vending, and refunding event tickets through a pioneering approach.

5/17/2024

TicketBuddy

3

Problem Definition

- To design, develop and implement TicketBuddy, a comprehensive event ticket management platform, incentivizing both buyers and sellers with a clever, innovative and truly novel concept.

5/17/2024

TicketBuddy

4

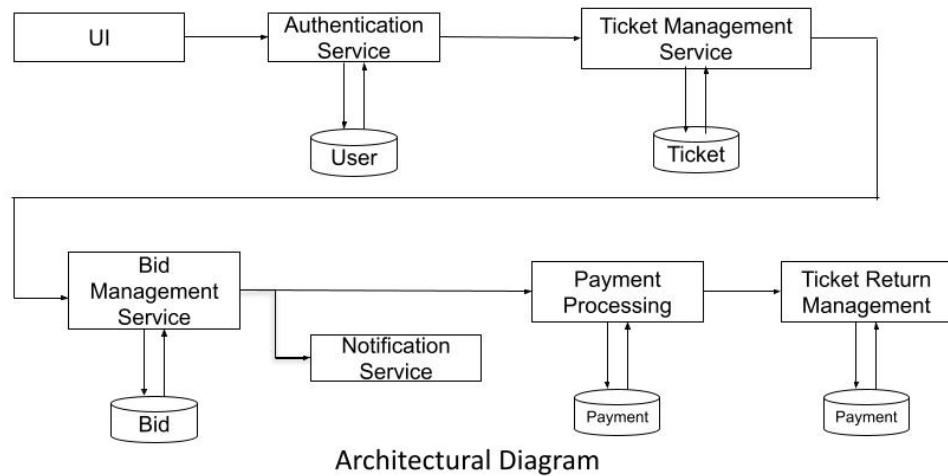
Objectives

- Develop a user-friendly interface platform for that allows entities to sell tickets to the venue(s) pertaining to the event(s) conducted by them.
- Streamline the user sign-up or login process to facilitate access to the customization platform.
- Allow sellers to update ticket details like price, quantity, and auction mode
- Update platform interface in real-time for users browsing tickets.

Scope and Relevance

- **Scope:** TicketBuddy provides a platform that allows entities to sell tickets to the venue(s) pertaining to the event(s) conducted by them, its compensations mechanism allows individuals to potentially profit off of its auctioning system.
- **Relevance :**A dynamic ticket management system facilitating event ticket allocation, bidding, and returns with real-time bid processing, efficient ticket return management, and seamless user experience.
- **Application:** Useful for performing musicians and artists and for fellow concertgoers or gig-goers, helping them find tickets suiting their needs

System Design



5/17/2024

TicketBuddy

7

System Design

1. Authentication Service - Handles user authentication and authorization securely
 - Robust authentication mechanisms.
 - Strong password policies.
2. Ticket Management Service - Manages ticket listings and updates in real-time
 - Real-time updates of ticket listings.
 - Scalable handling of concurrent requests.
 - Integration with other modules.

5/17/2024

TicketBuddy

8

System Design

3. Bid Management Service - Handles bidding process and bid outcomes
 - o Recording bids with timestamps.
 - o Real-time bid processing.
 - o Automated ticket allocation.
4. Notification Service - Generates and delivers notifications
 - o Low latency notifications.
 - o Integration with user interface.
 - o Real-time delivery of notifications.

5/17/2024

TicketBuddy

9

System Design

5. Payment Processing - Handles secure payment processing
 - o Integration with financial systems.
 - o Encryption of payment data.
 - o Compliance with industry standards.
6. Ticket Return Management - Manages ticket returns and refunds
 - o Fast ticket return processing.
 - o Refunds to original payment method.
 - o Integration with ticket management system.

5/17/2024

TicketBuddy

10

System Design

7. User Interface - Provides interface for user interaction with the system
 - Intuitive design.
 - Responsive layout.
 - Accessibility features.

System Design

Algorithm 1 - Bid Matching Algorithm

- Description: This algorithm determines the highest bidders for each ticket listing during auction mode. The sorting mechanism employs a combination of heap, merge and quick sort algorithms to sort the values.
- Steps:
 1. Retrieve all pending bids for each ticket listing.
 2. Sort bids in descending order based on bid amount.
 3. Allocate tickets to the highest bidders until all tickets are assigned or bids are exhausted.
- Complexity: $O(n \log n)$ due to sorting operation.

System Design

Algorithm 2 - Ticket Allocation Algorithm

- Description: This algorithm finalizes ticket allocation based on bid outcomes.
- Steps:
 1. Check bid outcomes for each ticket listing.
 2. Allocate tickets to successful bidders.
 3. Notify successful bidders.
 4. Handle cases of no bids or insufficient bids.
- Complexity: $O(m)$, where m is the number of ticket listings.

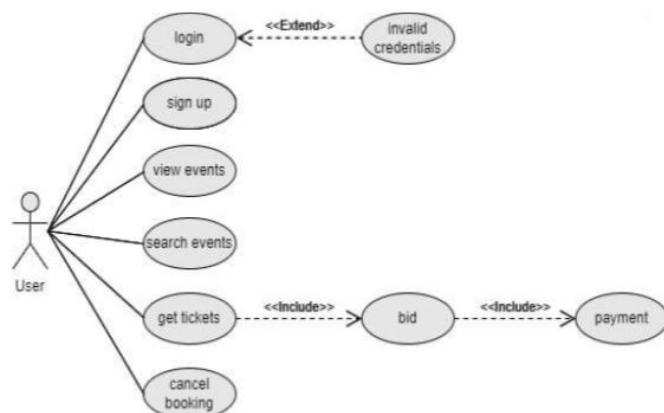
5/17/2024

TicketBuddy

13

System Design

Use Case Diagram



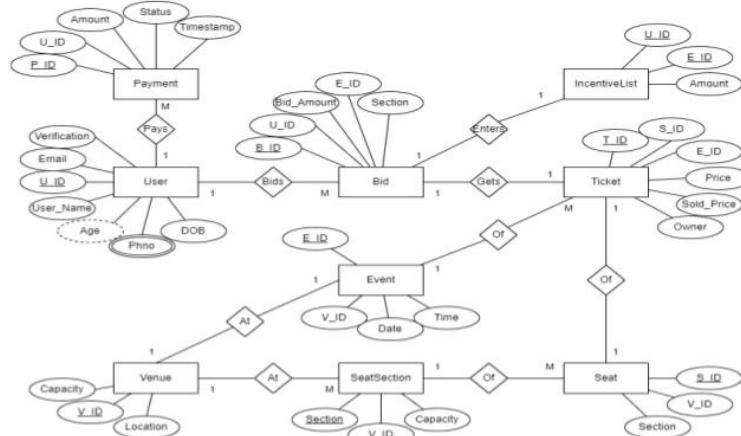
5/17/2024

TicketBuddy

14

System Design

Database Design



5/17/2024

TicketBuddy

15

Work Division

Gantt Chart



5/17/2024

TicketBuddy

16

Software/ Hardware Requirements

- Backend Framework: Python(version 3.9) with Flask
- Frontend Technology: HTML, CSS, JavaScript (React.js)
- Database Management System: PostgreSQL(version 13)

5/17/2024

TicketBuddy

17

Software/ Hardware Requirements

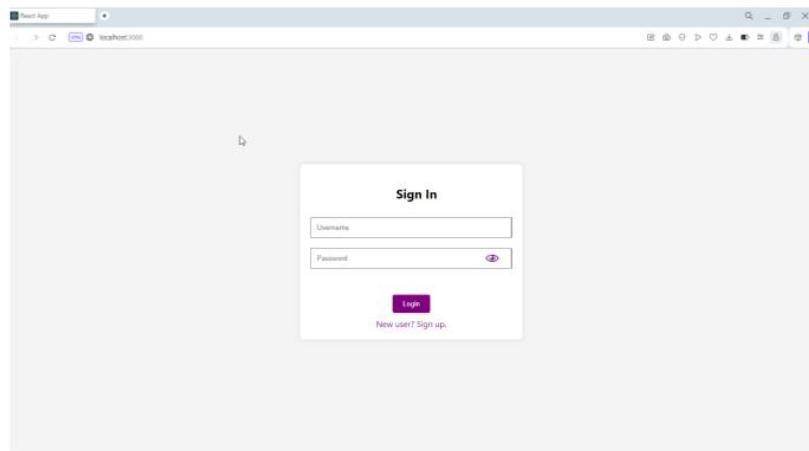
| | Windows requirements | Mac requirements | Linux requirements |
|--------------------------------|--------------------------------|-------------------------------|---|
| Operating system | Windows 10 or later | macOS Catalina 10.15 or later | 64-bit Ubuntu 18.04+, Debian 10+, openSUSE 15.5+, or Fedora Linux 38+ |
| Processor | Intel Pentium 4 or later | Intel | Intel Pentium 4 or later |
| Memory | 2 GB minimum, 4 GB recommended | | |
| Screen resolution | 1280x1024 or larger | | |
| Application window size | 1024x680 or larger | | |
| Internet connection | Required | | |

5/17/2024

TicketBuddy

18

Results

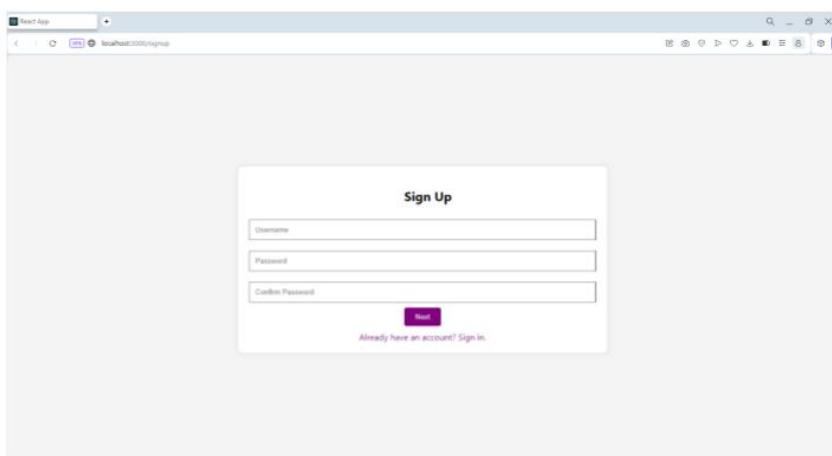


5/17/2024

TicketBuddy

19

Results



5/17/2024

TicketBuddy

20

Results

The screenshot shows the TicketBuddy website interface. At the top, there is a dark header bar with the 'TicketBuddy' logo. Below the header, a section titled 'Featured Events' displays four event cards. Each card contains a thumbnail image, the event name 'Example Event', a genre placeholder 'Example Genre', and a 'Get Tickets' button. The fourth card is partially visible below the others.

5/17/2024

TicketBuddy

21

Results

This screenshot shows the same TicketBuddy website as the previous one, but with a navigation bar at the top. The navigation bar includes links for 'Edit Personal Details', 'View Tickets', and 'Logout'. The rest of the page structure is identical to the first screenshot, displaying the 'Featured Events' section with four cards.

5/17/2024

TicketBuddy

22

Results

A screenshot of a web browser window titled "TicketBuddy". The page displays a section titled "YOUR TICKETS" containing three entries:

| Ticket Details | Action |
|---------------------------------------|-------------------------|
| Playboi Carti 12-06-2024 #A0034B1 | <button>Cancel</button> |
| Playboi Carti 12-06-2024 #A0034B1 | <button>Cancel</button> |
| Playboi Carti 12-06-2024 #A0034B1 | <button>Cancel</button> |

5/17/2024

TicketBuddy

23

Results

A screenshot of a web browser window titled "TicketBuddy". The page displays a search result for "Kanye West" on "20-05-2025". The results include a large image of a crowd and a seating chart.

Kanye West
20-05-2025

TOTAL
40

Buy Tickets

The seating chart shows a grid of 40 numbered seats, arranged in rows and columns, with some seats highlighted in yellow.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 |
| 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 |
| 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 |
| 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 |
| 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 |
| 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 |
| 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 |

5/17/2024

TicketBuddy

24

Results

The screenshot shows a web browser window titled "React App" with the URL "localhost:3001/ticket". The main content is a dark-themed interface for "TicketBuddy". At the top, there's a banner for a "Kanye West Rap" event on "20-05-2025". Below the banner is a photo of a crowd. To the left, there's a table titled "Position" with columns "Bid Amount" and "User ID". The table contains three rows:

| Position | Bid Amount | User ID |
|----------|------------|---------|
| 1 | 100 | 1 |
| 2 | 48 | 3 |
| 3 | 10 | 2 |

To the right of the table is a "MAKE BID" form with a "Bid Amount" input field and a "Confirm" button.

5/17/2024

TicketBuddy

25

Conclusion

- Our app TicketBuddy is a dynamic ticket management system facilitating event ticket allocation, bidding, and returns with real-time bid processing, efficient ticket return management, and seamless user experience, providing incentivize for both buyers and sellers.

5/17/2024

TicketBuddy

26

Future Enhancements

- **Rewards Program:** Implement a loyalty program where users can earn points for each ticket purchase or auction participation, which can be redeemed for discounts or exclusive access to events.
- **24/7 Live Support:** Introduce a 24/7 live chat support feature to assist users with any issues or inquiries in real-time.

References

- Website Features and their Individual Characteristics to the Context of the Influence of the Network Shopping Acceptance. X. Zhong. January 2021. Advances in Economics, Business and Management Research, volume 203 (pages 11-16)
- A Study on the Conversion mode of Customer Satisfaction and Customer Loyalty in Online Shopping-Taking Taobao, Tmall and JD e-commerce sites as examples. Q. Wang, H. Guo, M. Liu. January 2023. International Journal of social science and humanities(pages 10-29)
- “Future of e-commerce in India”, Chanana Nisha and Goele Sangeeta, March 2016, International Journal of Computing & Business Research(pages 231-246)

References

- Bidder and Seller Strategies in Online Auctions: A Review, Tsinghua International School, Beijing, China, Audrey Elizabeth Wen, HoiYat Wong, January 2008, Proceedings of the 2021 3rd International Conference on Economic Management and Cultural Industry(pages 1992-1997)
- A Review of Online Auction and It's Pros and Cons, Rashesh G Chothani, Nainesh A Patel, Asagarali H Dekavadiya, Punit R Patel, February 2021, International Journal of Advance Engineering and Research Development(pages 8-11)

References

- ECONOMIC INSIGHTS FROM INTERNET AUCTIONS: A SURVEY, Patrick Bajari and Ali Hortacsu, June 2004, Journal of Economic Literature(pages 457-481)
- Flask Web Framework based News Summarizer: Web Application, Apeksha Kamble, Adnan GaganPooja Wagh, Umar Aziz, Pooja Wagh, December 2022, International Journal of Scientific Research and Engineering Development—Volume 5 Issue 6(pages 953-956)

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039
(Affiliated to APJ Abdul Kalam Technological University)**



Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

- 1. Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and Team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills

The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills

The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills

The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes

After the completion of the course the student will be able to:

CO1:

Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)

CO2:

Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)

CO3:

Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)

CO4:

Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)

CO5:

Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

Appendix C: CO-PO-PSO Mapping

COURSE OUTCOMES:

After completion of the course the student will be able to

| SL. NO | DESCRIPTION | Blooms' Taxonomy Level |
|-------------------|---|---------------------------------------|
| CO1 | Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply) | Level 3: Apply |
| CO2 | Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply) | Level 3: Apply |
| CO3 | Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply) | Level 3: Apply |
| CO4 | Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply) | Level 3: Apply |
| CO5 | Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply) | Level 3: Apply |

CO-PO AND CO-PSO MAPPING

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 | PS O3 |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|----------|----------|----------|----------|----------|
| C O1 | 3 | 3 | 3 | 3 | | 2 | 2 | 3 | 2 | 2 | 2 | 3 | 2 | 2 | 2 |
| C O2 | 3 | 3 | 3 | 3 | 3 | 2 | | 3 | 2 | 3 | 2 | 3 | 2 | 2 | 2 |
| C O3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 3 | 2 | 2 | 2 | 3 | | | 2 |
| C O4 | 2 | 3 | 2 | 2 | 2 | | | 3 | 3 | 3 | 2 | 3 | 2 | 2 | 2 |
| C O5 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 3 | 2 | | 2 | 3 | 2 | 2 | 2 |

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

| MAPPING | LOW/ MEDIUM/ HIGH | JUSTIFICATION |
|--------------------------|-------------------------|---|
| 101003/CS6 22T.1-PO1 | HIGH | Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| 101003/CS6 22T.1-PO2 | HIGH | Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics. |
| 101003/CS6 22T.1-PO3 | HIGH | Design solutions for complex engineering problems by identifying technically and economically feasible problems. |
| 101003/CS6 22T.1-PO4 | HIGH | Identify technically and economically feasible problems by analysis and interpretation of data. |
| 101003/CS6 22T.1-PO6 | MEDIUM | Responsibilities relevant to the professional engineering practice by identifying the problem. |
| 101003/CS6 22T.1-PO7 | MEDIUM | Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions. |
| 101003/CS6 22T.1-PO8 | HIGH | Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems. |
| 101003/CS6 22T.1-PO9 | MEDIUM | Identify technically and economically feasible problems by working as a team. |
| 101003/CS6 22T.1-PO10 | MEDIUM | Communicate effectively with the engineering community by identifying technically and economically feasible problems. |
| 101003/CS6 22T.1-P011 | MEDIUM | Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems. |
| 101003/CS6 22T.1-PO12 | HIGH | Identify technically and economically feasible problems for long term learning. |
| 101003/CS6 22T.1-PSO1 | MEDIUM | Ability to identify, analyze and design solutions to identify technically and economically feasible problems. |
| 101003/CS6 22T.1-PSO2 | MEDIUM | By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems. |
| 101003/CS6 22T.1-PSO3 | MEDIUM | Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems. |
| 101003/CS6 22T.2-PO1 | HIGH | Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals. |

| | | |
|--------------------------|---------------|--|
| 101003/CS6 22T.2-PO2 | HIGH | Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes. |
| 101003/CS6 22T.2-PO3 | HIGH | Design solutions for complex engineering problems and design based on the relevant literature. |
| 101003/CS6 22T.2-PO4 | HIGH | Use research-based knowledge including design of experiments based on relevant literature. |
| 101003/CS6 22T.2-PO5 | HIGH | Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools. |
| 101003/CS6 22T.2-PO6 | MEDIUM | Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature. |
| 101003/CS6 22T.2-PO8 | HIGH | Apply ethical principles and commit to professional ethics based on the relevant literature. |
| 101003/CS6 22T.2-PO9 | MEDIUM | Identify and survey the relevant literature as a team. |
| 101003/CS6 22T.2-PO10 | HIGH | Identify and survey the relevant literature for a good communication to the engineering fraternity. |
| 101003/CS6 22T.2-PO11 | MEDIUM | Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles. |
| 101003/CS6 22T.2-PO12 | HIGH | Identify and survey the relevant literature for independent and lifelong learning. |
| 101003/CS6 22T.2-PSO1 | MEDIUM | Design solutions for complex engineering problems by Identifying and survey the relevant literature. |
| 101003/CS6 22T.2-PSO2 | MEDIUM | Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices. |
| 101003/CS6 22T.2-PSO3 | MEDIUM | Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research. |
| 101003/CS6 22T.3-PO1 | HIGH | Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals. |
| 101003/CS6 22T.3-PO2 | HIGH | Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions. |

| | | |
|--------------------------|---------------|--|
| 101003/CS6 22T.3-PO3 | HIGH | Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies. |
| 101003/CS6 22T.3-PO4 | HIGH | Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| 101003/CS6 22T.3-PO5 | HIGH | Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools. |
| 101003/CS6 22T.3-PO6 | MEDIUM | Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues. |
| 101003/CS6 22T.3-PO7 | MEDIUM | Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions. |
| 101003/CS6 22T.3-PO8 | HIGH | Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics. |
| 101003/CS6 22T.3-PO9 | MEDIUM | Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings. |
| 101003/CS6 22T.3-PO10 | MEDIUM | Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies. |
| 101003/CS6 22T.3-PO11 | MEDIUM | Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies. |
| 101003/CS6 22T.3-PO12 | HIGH | Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions. |
| 101003/CS6 22T.3-PSO3 | MEDIUM | The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies. |
| 101003/CS6 22T.4-PO1 | MEDIUM | Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| 101003/CS6 22T.4-PO2 | HIGH | Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation. |

| | | |
|--------------------------|---------------|---|
| 101003/CS6 22T.4-PO3 | MEDIUM | Prepare Design solutions for complex engineering problems and create technical report and deliver presentation. |
| 101003/CS6 22T.4-PO4 | MEDIUM | Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation. |
| 101003/CS6 22T.4-PO5 | MEDIUM | Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation. |
| 101003/CS6 22T.4-PO8 | HIGH | Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| 101003/CS6 22T.4-PO9 | HIGH | Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings. |
| 101003/CS6 22T.4-PO10 | HIGH | Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation. |
| 101003/CS6 22T.4-PO11 | MEDIUM | Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work by prepare technical report and deliver presentation. |
| 101003/CS6 22T.4-PO12 | HIGH | Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation. |
| 101003/CS6 22T.4-PSO1 | MEDIUM | Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. |
| 101003/CS6 22T.4-PSO2 | MEDIUM | To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation. |
| 101003/CS6 22T.4-PSO3 | MEDIUM | To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation. |
| 101003/CS6 22T.5-PO1 | HIGH | Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| 101003/CS6 22T.5-PO2 | HIGH | Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project. |

| | | |
|--------------------------|---------------|--|
| 101003/CS6 22T.5-PO3 | HIGH | Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs. |
| 101003/CS6 22T.5-PO4 | MEDIUM | Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| 101003/CS6 22T.5-PO5 | MEDIUM | Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project. |
| 101003/CS6 22T.5-PO6 | MEDIUM | Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project. |
| 101003/CS6 22T.5-PO7 | MEDIUM | Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project. |
| 101003/CS6 22T.5-PO8 | HIGH | Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project. |
| 101003/CS6 22T.5-PO9 | MEDIUM | Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project. |
| 101003/CS6 22T.5-PO11 | MEDIUM | Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project. |
| 101003/CS6 22T.5-PO12 | HIGH | Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project. |
| 101003/CS6 22T.5-PSO1 | MEDIUM | The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project. |

| | | |
|--------------------------|---------------|---|
| 101003/CS6 22T.5-PSO2 | MEDIUM | The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project. |
| 101003/CS6 22T.5-PSO3 | MEDIUM | The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project. |