C.V. RAMAN GLOBAL UNIVERSITY BHUBANESWAR, ODISHA

Department of Computer Science & Engineering.



**SUBMITTED BY: -**

|  |  |  |
| --- | --- | --- |
| **MEMBER 1** | **NAME** | **OMM PRAKASH MOHANTY** |
| **REGD. NO** | **CL20250106019409113** |
| **BRANCH** | **CSE** |
| **EMAIL** | **2201020019@cgu-odisha.ac.in** |
| **MEMBER 2** | **NAME** | **BADITHI SANAMUKH** |
| **REGD. NO** | **CL2025010601952564** |
| **BRANCH** | **CSIT** |
| **EMAIL** | **2201020193@cgu-odisha.ac.in** |

**ACKNOWLEDGEMENT**

It is with immense gratitude and appreciation that we, the team, extend our heartfelt thanks to Mr. Pavan Kumar for his exceptional guidance, unwavering support, and invaluable encouragement throughout the development of the “Train Reservation System using JDBC”. His profoundexpertise, insightful suggestions, and constructive feedback have been instrumental in shaping the direction of this project and helping us overcome numerous challenges. His patience, dedication, and mentorship have been a constant source of inspiration and motivation, enabling us to deepen our understanding of Java programming, database connectivity, and software development.

We are deeply thankful to the faculty members and staff of Cranes for providing a conducive academic environment, access to essential resources, and their unwavering support, which laid the foundation for the successful completion of this project. Their encouragement and guidance at various stages of this journey have been invaluable.

We would also like to express our gratitude to our peers and colleagues for their enthusiastic collaboration, insightful discussions, and timely feedback, which played a significant role in refining the scope and execution of this project. Their camaraderie and shared passion for learning have made this experience truly enriching.

Lastly, we extend our heartfelt appreciation to our families and friends for their unconditional love, encouragement, and understanding throughout the duration of this project. Their unwavering support has been a pillar of strength, allowing us to dedicate ourselves fully to this endeavor.

This project has been an incredibly fulfilling learning journey, and we are deeply grateful to everyone who contributed to its successful realization. Thank you all for being a part of this experience.

**ABSTRACT**

The Train Reservation System using JDBC is a robust and efficient application designed to streamline train ticket booking, cancellation, and reservation management. Addressing the increasing demand for digital solutions in railway booking, this system utilizes Java Database Connectivity (JDBC) to facilitate smooth communication between the application and an Oracle database. The primary goal of the project is to provide users with a convenient platform for managing train reservations through a Command Line Interface (CLI).

Developed using Java as the core programming language, the system incorporates SQL to handle database operations related to train schedules, ticket bookings, cancellations, and seat availability. The Oracle database ensures data persistence and integrity, while JDBC enables seamless interaction between the application and the database. The development process is carried out using integrated development environments such as Eclipse or IntelliJ IDE, enabling efficient coding and debugging.

The system offers essential features such as train search and availability checking, ticket booking and cancellation, and viewing booking history and status. It employs robust transaction management to maintain data consistency and accuracy during booking and cancellation processes. The modular and scalable architecture of the system allows for easy maintenance and future enhancements, including the potential integration of a Graphical User Interface (GUI) or online booking functionality.

Challenges such as data consistency, transaction handling, and real-time updates are addressed through efficient SQL query optimization and error handling mechanisms. The system has been thoroughly tested for performance and reliability, delivering a practical and user-friendly solution for train reservation management. In the future, features like secure payment integration, multi-user access control, and real-time seat tracking can further enhance the system’s functionality and usability.

By combining the power of Java, JDBC, and SQL, this project effectively demonstrates the development of a practical and scalable solution for train reservation management, meeting the needs of both users and administrators.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| Sl. No. | Content | Page |
| 1. | Introduction | 1 |
| 2. | **Objectives** | 2 |
| 3. | Functionality | 3 |
| 4. | Technologies and Tools | 4 |
| 5. | Implementation Details | 5 |
| 6. | Code and Output | 6-9 |
| 7. | Challenges Faced | 10 |
| 8. | Future Enhancements | 11 |
| 9. | Conclusion | 12 |
| 10. | References | 13 |

**INTRODUCTION**

The Train Reservation System using JDBC is a Java-based application that streamlines the process of booking train tickets, managing reservations, and checking availability. This application provides a user-friendly Command Line Interface (CLI) for seamless interaction and efficient train ticket management. The system is designed to cater to railway booking requirements with features such as viewing available trains, booking tickets, canceling reservations, and checking existing bookings.

This system leverages Java Database Connectivity (JDBC) to connect with an Oracle database, ensuring smooth communication between the application and the database. By using SQL queries and implementing robust transaction management, the system maintains data consistency and accuracy, minimizing the risk of booking conflicts and data loss. The use of Eclipse/IntelliJ IDEs during development ensures efficient coding and debugging practices.

The modular architecture allows for scalability and easy maintenance, making it adaptable to changing requirements or future enhancements. Potential upgrades include integrating a graphical user interface (GUI) for a more intuitive experience or enabling online booking capabilities through web-based or mobile applications. Additionally, implementing security measures, such as user authentication and secure transactions, can further enhance the system’s reliability and safety.

By combining the power of Java, JDBC, and Oracle, this Train Reservation System offers a solid foundation for efficient ticket management while allowing ample room for future improvements and integration.

**OBJECTIVES**

The primary objective of the Train Reservation System using JDBC is to develop a database-driven application that allows users to efficiently manage train ticket bookings. The system is designed to ensure accuracy, reliability, and ease of use in handling train reservations. The key objectives of this project are:

1. **To Develop a User-Friendly Train Reservation System**
   * Provide a console-based interface for passengers to book, cancel, and view their tickets.
   * Ensure a seamless and intuitive interaction for users.
2. **To Implement Efficient Train and Ticket Management**
   * Maintain a structured database to store train schedules, seat availability, and passenger bookings.
3. **To Enable Secure and Reliable Ticket Booking**
   * Prevent overbooking by dynamically updating available seats after each booking.
   * Assign seat numbers in a systematic manner for each reservation.
4. **To Allow Ticket Cancellation with Seat Restoration**
   * Provide an option to cancel tickets while ensuring seat availability is updated accordingly.
5. **To Use JDBC for Database Connectivity**
   * Establish a connection between the Java application and Oracle database using JDBC.
   * Utilize Prepared Statements to prevent SQL injection and ensure secure transactions.
6. **To Ensure Data Integrity and Transaction Management**
   * Use commit and rollback mechanisms to maintain database consistency in case of errors.

**FUNCTIONALITY**

The Train Reservation System using JDBC provides the following key functionalities:

1. **View Available Trains:**  
   This feature allows users to view the list of available trains, displaying comprehensive details such as train number, train name, source, destination, available seats, and ticket price. Users can easily find relevant train options based on their travel requirements. The data is fetched dynamically from the Oracle database using SQL queries through JDBC, ensuring accurate and up-to-date information.
2. **Book a Ticket:**  
   Users can book a train ticket by providing their name and selecting a train from the list of available options. The system automatically assigns a seat and updates the seat availability in the database. To ensure data integrity and prevent overbooking, the system employs transaction management using commit and rollback mechanisms. The booking details, including ticket ID, train number, passenger name, and seat number, are stored in the tickets table.
3. **Cancel a Ticket:**  
   This functionality enables users to cancel their booking by providing the ticket ID. Upon successful cancellation, the system updates the seat availability by releasing the booked seat. The ticket entry is removed from the database, maintaining accurate seat count and availability. The system ensures that cancellations are handled efficiently, with appropriate error messages if the ticket ID is invalid or not found.
4. **View My Tickets:**  
   Users can view their booked tickets, which display key information such as train number, seat number, passenger name, and fare. This feature provides a consolidated list of all reservations made by the user, retrieved from the database using optimized SQL queries. It helps users keep track of their bookings and verify travel details conveniently.

**TECHNOLOGY USED**

1. **Java** – The core programming language used for implementing the system's logic and user interactions.
2. **JDBC (Java Database Connectivity)** – Enables communication between the Java application and the Oracle database.
3. **Oracle Database** – Stores train details, ticket bookings, and seat availability, ensuring data persistence.
4. **SQL (Structured Query Language)** – Used for querying and managing database records, including train schedules and reservations.
5. **Eclipse/IntelliJ (IDE)** – Recommended development environments for writing and debugging Java code.
6. **Command Line Interface (CLI)** – Provides a simple, text-based user interaction for booking and managing tickets.

**IMPLEMENTATION**

The Train Reservation System using JDBC is a console-based Java application designed to simplify train ticket booking, cancellation, and reservation management. Utilizing Java Database Connectivity (JDBC), the system establishes a secure connection with an Oracle database, allowing for efficient data handling and real-time updates. The application provides a simple and user-friendly Command Line Interface (CLI), making it accessible and easy to use.

The system's database setup involves two primary tables:

1. **Trains Table:** Stores details about available trains, including train number, name, source, destination, schedule, and seat count.
2. **Tickets Table:** Tracks booking details such as ticket ID, train number, passenger information, travel date, and booking status.

JDBC connectivity is achieved by dynamically loading the Oracle JDBC driver and using Prepared Statements to execute SQL queries securely, protecting against SQL injection attacks. The system follows a structured approach to manage connections efficiently, ensuring that all resources are properly closed to prevent leaks.

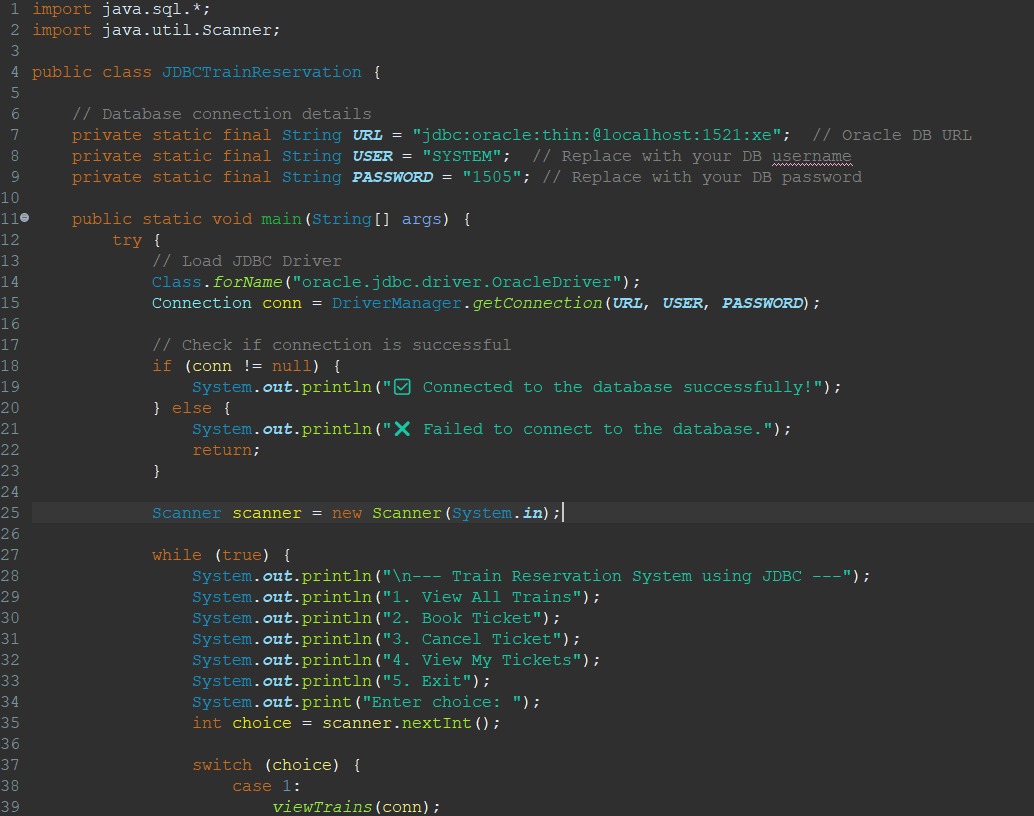
**Core Functions:**

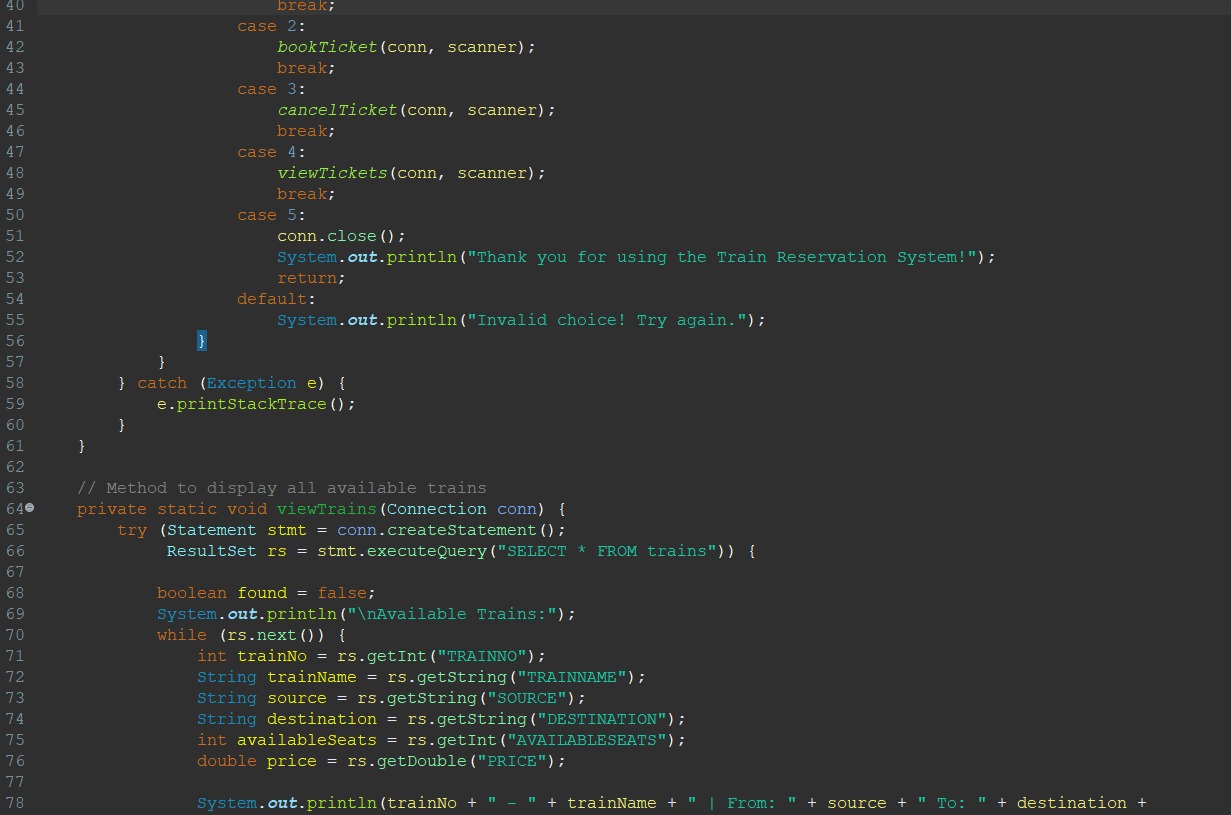
1. **View Trains:** Displays available trains with complete details.
2. **Book Ticket:** Reserves seats after checking availability, generating a unique ticket ID upon success.
3. **Cancel Ticket:** Cancels a reservation by ticket ID, updating availability.
4. **View My Tickets:** Lists all bookings with travel details and status.

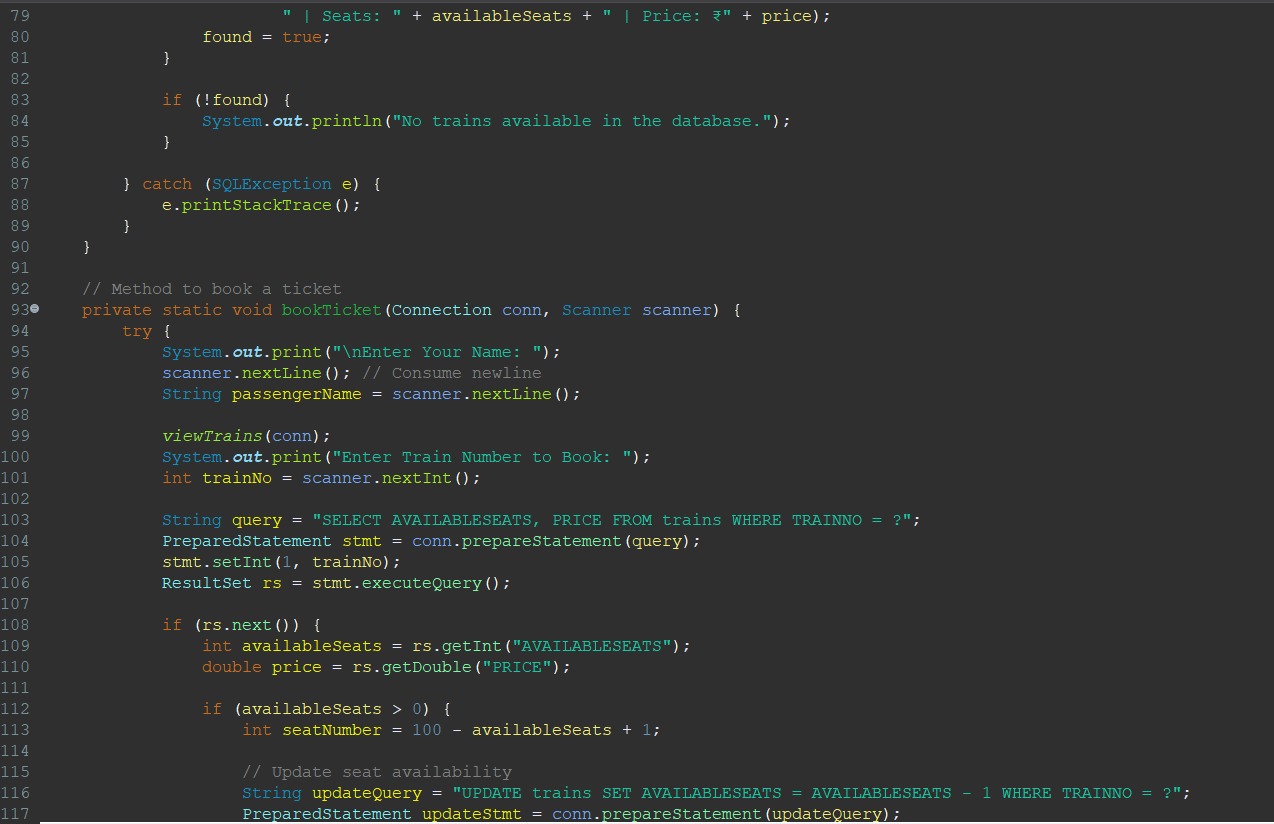
To ensure data consistency, the system employs transaction management using commit and rollback operations. In case of any error during booking or cancellation, the system reverts changes to maintain data integrity. Additionally, robust error handling addresses SQL exceptions, input validation issues, and connection failures.

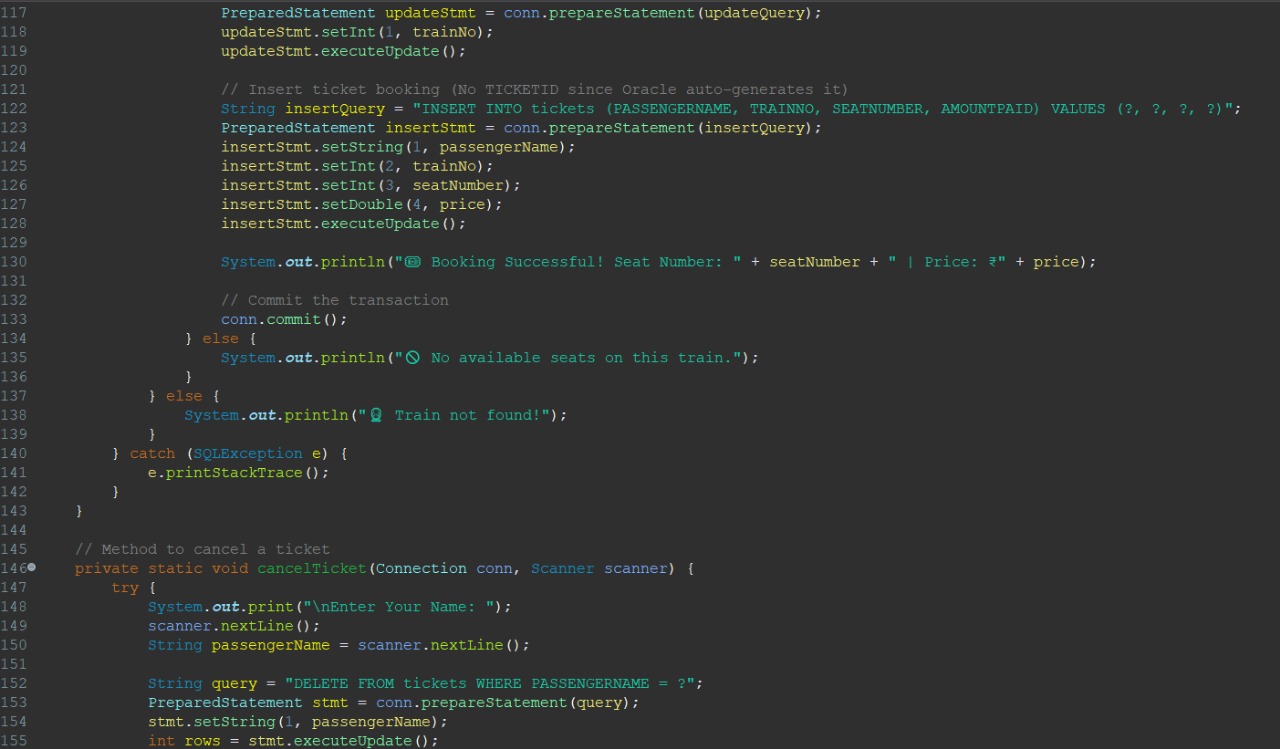
The application is developed using Eclipse/IntelliJ IDE, following a modular and scalable architecture that allows for future enhancements such as a Graphical User Interface (GUI), online booking integration, and secure payment processing. The system has been thoroughly tested for performance and reliability, providing an effective solution for train reservation management with a focus on accuracy, security, and user convenience.

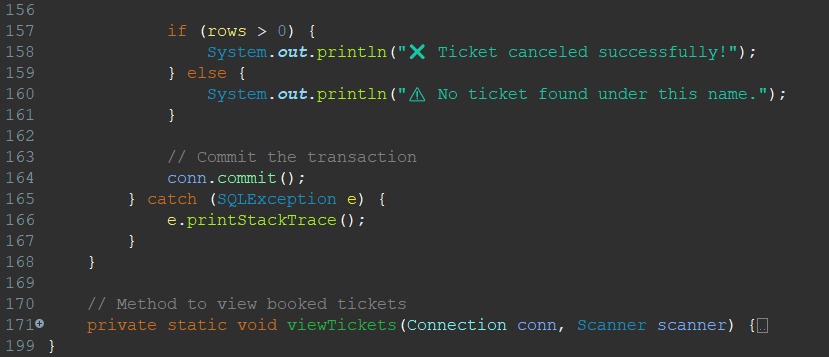
**CODE**



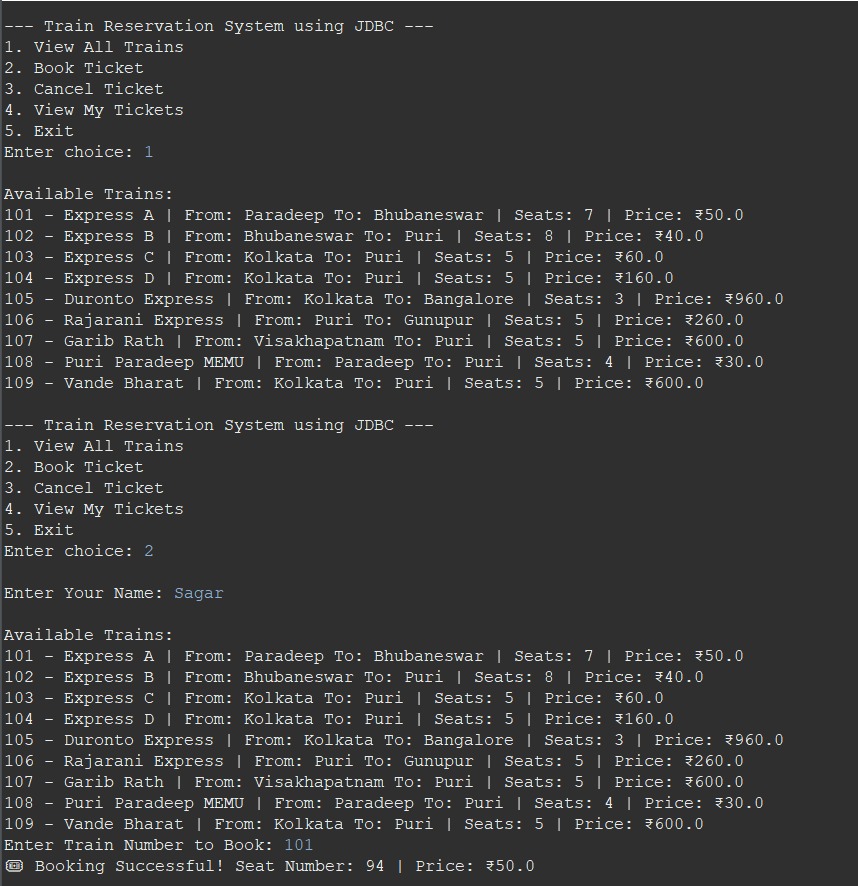


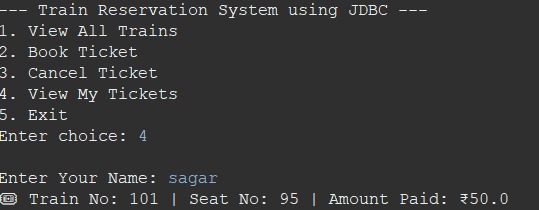






OUTPUT





**CHALLENGES FACED**

During the development of the Train Reservation System using JDBC, several challenges were encountered and addressed to ensure smooth functionality and performance.

1. **Database Connectivity**: Establishing a reliable connection between the Java application and the Oracle database using JDBC was challenging. Issues like driver compatibility, incorrect URLs, and connection failures needed careful troubleshooting. Proper configuration and dynamic loading of the Oracle JDBC driver ensured a stable connection, while implementing connection pooling optimized resource management.
2. **Seat Management**: Managing real-time seat availability while preventing overbooking was critical. Simultaneous booking attempts posed the risk of race conditions. To handle this, we implemented locking mechanisms and transaction isolation to synchronize seat availability checks and updates, ensuring accurate booking and preventing conflicts.
3. **SQL Optimization:** Writing efficient SQL queries was crucial to maintaining performance, especially with large datasets. Initially, unoptimized queries led to slow responses. We improved efficiency by using Prepared Statements, indexing frequently accessed columns (like train numbers), and employing batch processing to reduce execution time and server load.
4. **Transaction Handling:** Maintaining data consistency during booking and cancellation required robust transaction management. We implemented commit and rollback operations to ensure that incomplete transactions did not corrupt the database. This all-or-nothing approach ensured that bookings were fully processed or completely reverted in case of errors.
5. **Error Handling:** Managing unexpected errors and invalid inputs was essential for a smooth user experience. We handled SQL exceptions, input validation errors, and connection timeouts using a comprehensive error-handling framework. Meaningful error messages guided users and developers to quickly identify and resolve issues.

**FUTURE ENHANCEMENTS**

The Train Reservation System using JDBC can be significantly enhanced by incorporating additional features and modernizing its interface and functionality. Below are some potential improvements that can be implemented to increase usability, accessibility, and overall user experience:

1. **Graphical User Interface (GUI):**  
   Replacing the current Command-Line Interface (CLI) with a more intuitive Graphical User Interface (GUI) would greatly enhance user interaction. A GUI can be built using Java Swing, JavaFX, or even web technologies like HTML, CSS, and JavaScript for a more modern look.
2. **Online Booking System:**  
   To extend the system’s accessibility, an online booking system can be developed as a web or mobile application. Integrating web technologies like Spring Boot with JDBC would enable users to book tickets remotely from any device.
3. **User Authentication:**  
   Adding a user authentication system would enable personalized ticket management. Features like user registration, login, and password management could enhance security and user convenience.
4. **Payment Integration:**  
   Incorporating online payment options would make the booking process more seamless and efficient. Integrating payment gateways such as PayPal, Stripe, or Razorpay would allow users to complete transactions directly within the application.
5. **Waitlist & Notifications:**  
   Implementing a waitlist system would help manage bookings when trains are fully booked. Passengers could join a waitlist and be automatically upgraded to confirmed status if seats become available.

CONCLUSION

The Train Reservation System using JDBC successfully provides a simple and efficient way to manage train ticket bookings using a Java-based console application with Oracle database integration. The system allows users to view available trains, book tickets, cancel reservations, and check their bookings, ensuring real-time seat updates and data consistency.

Through JDBC connectivity, SQL queries, and transaction management, the system ensures reliability and security. Despite challenges like database setup, seat management, and error handling, structured coding and optimization helped in creating a functional system.

With potential future enhancements like a GUI, online booking, user authentication, and payment integration, this system can be further improved for better accessibility and user experience.

**REFERENCES**

1.**Oracle JDBC Documentation** – <https://docs.oracle.com/en/database/oracle/oracle-database/>

2. **Java SE Documentation** – <https://docs.oracle.com/en/java/javase/>

3. **SQL & Database Concepts** – https://www.w3schools.com/sql/

4. **JDBC Tutorial (GeeksforGeeks)** – https://www.geeksforgeeks.org/jdbc-tutorial/

5. **Java Programming (Oracle)** – <https://www.oracle.com/java/>