

Project Presentation

PRESENTED BY PRINCE KUMAR- 0111AL211125

— PRINCE RAJ - 0111AL211126

— MD SAQUIB BASHAR-0111AL211095

— YASHASVEE SISODIYA-0111AL211185

Online Railway Reservation System with Java and SQL

An in-depth exploration of the design and implementation of an online railway reservation system using Java and SQL. Created by Prince kumar and helper name Prince raj, MD SAQUIB BASHAR, and Yashasvee Sisodiya.

System Overview

Introduction

Discover the concept and purpose of the online railway reservation system and its benefits to both users and the railway industry.

Key Features

Explore the essential features and functionalities that make the system efficient, user-friendly, and reliable.

Technologies Used

Learn about the Java programming language and SQL for efficient database management, and how they are integrated into the system for seamless operations.



Implementation Process

1 System Design

Understand the architectural design of the online railway reservation system, including the backend and frontend components.

2 Development and Coding

Dive into the coding process and see how Java and SQL are utilized to create the system's functionalities.

3 Testing and Quality Assurance

Learn about the rigorous testing and quality assurance procedures that ensure a stable and error-free online railway reservation system.

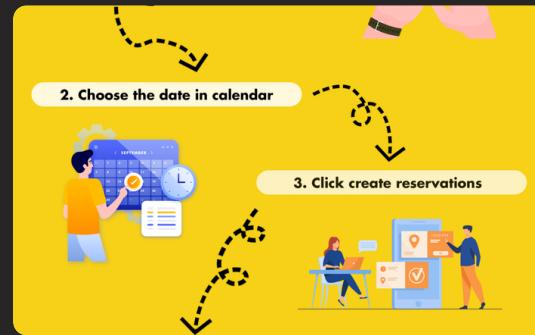


User Interface and Functionality



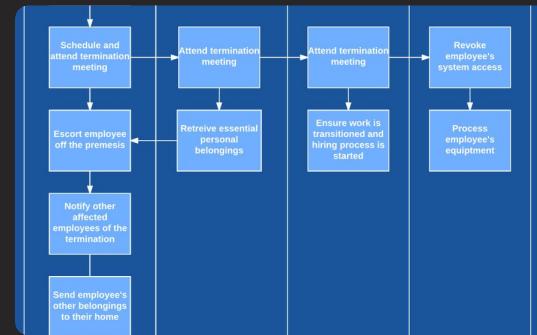
Overview

Get an overview of the user interface, designed to be intuitive and easy to navigate, ensuring a seamless booking experience for users.



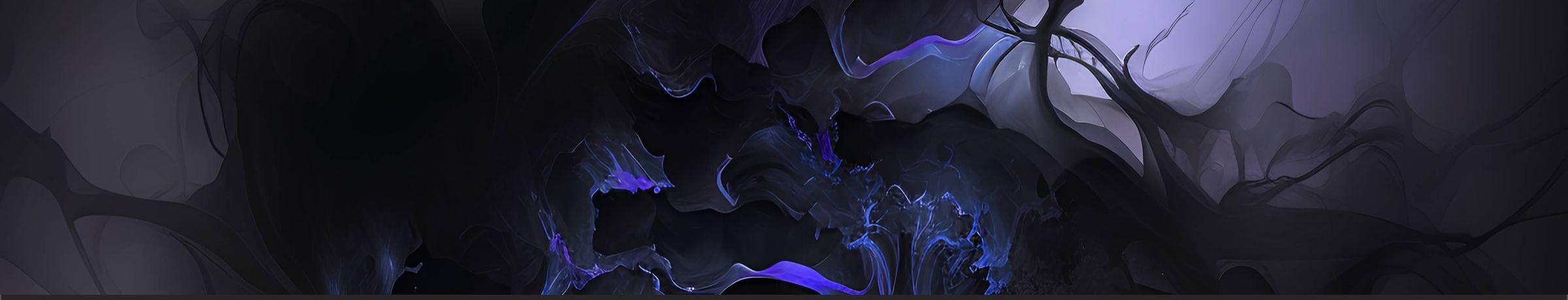
Booking Process

Explore the step-by-step ticket reservation process, including search, selection, payment, and ticket confirmation.



Cancellation

Understand the straightforward cancellation process to provide users with flexibility and peace of mind.



Security and Authentication

1 Importance of Security

Delve into the significance of robust security measures to protect user data, transactions, and avoid unauthorized access.

2 Authentication Techniques

Learn about the authentication methods used to verify user identities, ensuring secure access to the railway reservation system.

3 Data Protection Measures

Discover the various data protection measures implemented to safeguard sensitive information and prevent security breaches.

Future Enhancements and Scalability

Potential Improvements

Explore future enhancements and additional features that can be incorporated to improve the user experience and optimize the system's performance.

Scalability Considerations

Understand the scalability options and considerations for the online railway reservation system to handle increased user demand and expansion.

Conclusion

1

Summary

Recap the key points discussed throughout the presentation, highlighting the benefits and capabilities of the online railway reservation system.

2

Q&A Session

Engage in a question and answer session to address any inquiries regarding the system's functionality, design, and implementation.



Made with Gamma

JAVA CODE:-

```
import java.sql.*;
import java.util.Scanner;
import java.util.Random;

public class DBConnect {
    private static final int min = 1000;
    private static final int max = 9999;

    public static class user {
        private String username;
        private String password;

        Scanner sc = new Scanner(System.in);

        public user() {
        }

        public String getUsername() {
            System.out.println("Enter Username: ");
            username = sc.nextLine();
            return username;
        }

        public String getPassword() {
            System.out.println("Enter Password: ");
            password = sc.nextLine();
            return password;
        }
    }

    public static class PnrRecord {
        private int pnrNumber;
        private String passengerName;
        private String trainNumber;
        private String classType;
        private String journeyDate;
        private String from;
        private String to;

        Scanner sc = new Scanner(System.in);

        public int getpnrNumber() {
            Random random = new Random();
            pnrNumber = random.nextInt(max) + min;
            return pnrNumber;
        }

        public String getPassengerName() {
            System.out.println("Enter the passenger name : ");
            passengerName = sc.nextLine();
            return passengerName;
        }

        public String gettrainNumber() {
            System.out.println("Enter the train number : ");
            trainNumber = sc.nextLine();
            return trainNumber;
        }

        public String getClassType() {
            System.out.println("Enter the class type : ");
            classType = sc.nextLine();
            return classType;
        }

        public String getjourneyDate() {
            System.out.println("Enter the Journey date as 'YYYY-MM-DD' format");
            journeyDate = sc.nextLine();
            return journeyDate;
        }

        public String getfrom() {
            System.out.println("Enter the starting place : ");
            from = sc.nextLine();
            return from;
        }

        public String getto() {
            System.out.println("Enter the destination place : ");
            to = sc.nextLine();
            return to;
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        user u1 = new user();
        String username = u1.getUsername();
        String password = u1.getPassword();

        String url = "jdbc:mysql://127.0.0.1:3306/?user=root"; // change the database as per your requirement
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
        }

        try (Connection connection = DriverManager.getConnection(url, username, password)) {
            System.out.println("User Connection Granted.\n");
            while (true) {
                String InsertQuery = "insert into `Prince` .reservation values (?, ?, ?, ?, ?, ?, ?)";
                String DeleteQuery = "DELETE FROM `Prince` .reservation WHERE pnr_number = ?";
                String ShowQuery = "Select * from `Prince` .reservation";

                System.out.println("Enter the choice : ");
                System.out.println("1. Insert Record.\n");
                System.out.println("2. Delete Record.\n");
                System.out.println("3. Show All Records.\n");
                System.out.println("4. Exit.\n");
                int choice = sc.nextInt();

                if (choice == 1) {

                    PnrRecord p1 = new PnrRecord();
                    int pnr_number = p1.getpnrNumber();
                    String passengerName = p1.getPassengerName();
                    String trainNumber = p1.gettrainNumber();
                    String classType = p1.getClassType();
                    String journeyDate = p1.getjourneyDate();
                    String getfrom = p1.getfrom();
                    String getto = p1.getto();

                    try (PreparedStatement preparedStatement = connection.prepareStatement(InsertQuery)) {
                        preparedStatement.setInt(1, pnr_number);
                        preparedStatement.setString(2, passengerName);
                        preparedStatement.setString(3, trainNumber);
                        preparedStatement.setString(4, classType);
                        preparedStatement.setString(5, journeyDate);
                        preparedStatement.setString(6, getfrom);
                        preparedStatement.setString(7, getto);

                        int rowsAffected = preparedStatement.executeUpdate();
                        if (rowsAffected > 0) {
                            System.out.println("Record added successfully");
                        } else {
                            System.out.println("No records were added.");
                        }
                    }

                    catch (SQLException e) {
                        System.out.println("SQLException: " + e.getMessage());
                    }
                }

                else if (choice == 2) {
                    System.out.println("Enter the PNR number to delete the record : ");
                    int pnrNumber = sc.nextInt();
                    try (PreparedStatement preparedStatement = connection.prepareStatement(DeleteQuery)) {
                        preparedStatement.setInt(1, pnrNumber);
                        int rowsAffected = preparedStatement.executeUpdate();

                        if (rowsAffected > 0) {
                            System.out.println("Record deleted successfully");
                        } else {
                            System.out.println("No records were deleted.");
                        }
                    }

                    catch (SQLException e) {
                        System.out.println("SQLException: " + e.getMessage());
                    }
                }

                else if (choice == 3) {
                    try (PreparedStatement preparedStatement = connection.prepareStatement>ShowQuery)) {
                        ResultSet resultSet = preparedStatement.executeQuery();
                        System.out.println("\nAll records printing.\n");
                        while (resultSet.next()) {
                            String pnrNumber = resultSet.getString("pnr_number");
                            String passengerName = resultSet.getString("passenger_name");
                            String trainNumber = resultSet.getString("train_number");
                            String classType = resultSet.getString("class_type");
                            String journeyDate = resultSet.getString("journey_date");
                            String fromLocation = resultSet.getString("from_location");
                            String toLocation = resultSet.getString("to_location");

                            System.out.println("PNR Number: " + pnrNumber);
                            System.out.println("Passenger Name: " + passengerName);
                            System.out.println("Train Number: " + trainNumber);
                            System.out.println("Class Type: " + classType);
                            System.out.println("Journey Date: " + journeyDate);
                            System.out.println("From Location: " + fromLocation);
                            System.out.println("To Location: " + toLocation);
                            System.out.println("-----");
                        }
                    }

                    catch (SQLException e) {
                        System.out.println("SQLException: " + e.getMessage());
                    }
                }

                else if (choice == 4) {
                    System.out.println("Exiting the program.\n");
                    break;
                }

                else {
                    System.out.println("Invalid Choice Entered.\n");
                }
            }
        }

        catch (SQLException e) {
            System.out.println("SQLException: " + e.getMessage());
        }

        sc.close();
    }
}
```

SQL QUERIES:-

First create a Queries for the reservation table for a varchar type:-

```
CREATE TABLE reservation ( pnr_number VARCHAR(30) UNIQUE, passenger_name VARCHAR(30) NOT NULL, train_number VARCHAR(30) NOT NULL, class_type VARCHAR(30) NOT NULL, journey_date VARCHAR(30) NOT NULL, from_location VARCHAR(30) NOT NULL, to_location VARCHAR(36) NOT NULL );
```

```
mysql> desc reservation;
```

```
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| pnr_number | varchar(30) | YES | UNI | NULL || |
| passenger_name | varchar(30) | NO | | NULL || |
| train_number | varchar(30) | NO | | NULL || |
| class_type | varchar(30) | NO | | NULL || |
| journey_date | varchar(30) | NO | | NULL || |
| from_location | varchar(30) | NO | | NULL || |
| to_location | varchar(36) | NO | | NULL || |
+-----+-----+-----+-----+-----+
```

```
7 rows in set (0.00 sec)
```

The screenshot shows a terminal window with the MySQL command-line interface. The session starts with a connection to a local MySQL server (version 8.0.26). It then creates a table named 'reservation' with columns: pnr_number (unique), passenger_name, train_number, class_type, journey_date, from_location, and to_location. The 'pnr_number' column is defined as VARCHAR(30) and has a unique constraint ('UNI'). The other columns are VARCHAR types with various constraints like NOT NULL or specific lengths. After creating the table, it runs a 'desc reservation' command to show the table structure, which lists all columns and their properties. Finally, it runs a 'select * from reservation' command to view the data, which returns a single row with values: pnr_number = 132A, passenger_name = Prince, train_number = 9086, class_type = BI, journey_date = 2000-12-16, from_location = Dhambad, and to_location = Bhagal.

```
Last login: Sat Dec 23 12:39:02 on ttys000
Last connection: Sun Dec 24 12:39:02 on ttys000 via MySQL monitor.  Commands end with ; or q.
Your MySQL connection id is 18
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use Prince;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> select * from reservation;
+-----+-----+-----+-----+-----+
| pnr_number | passenger_name | train_number | class_type | journey_date | from_location | to_location |
+-----+-----+-----+-----+-----+
| 132A | Prince | 9086 | BI | 2000-12-16 | Dhambad | Bhagal |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> desc reservation;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| pnr_number | varchar(30) | YES | UNI | NULL |
| passenger_name | varchar(30) | NO | | NULL |
| train_number | varchar(30) | NO | | NULL |
| class_type | varchar(30) | NO | | NULL |
| journey_date | varchar(30) | NO | | NULL |
| from_location | varchar(30) | NO | | NULL |
| to_location | varchar(36) | NO | | NULL |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> /
```