VISHWESHVARAN M
20EUIT167

COVID VACCINATION BOOKING

CONSOLE APPLICATION DOCUMENTATION

DESCRIPTION:

The Covid Vaccination Booking application is a console application made using Java (JDBC) and MySql. Eclipse IDE is used to create this application. It provides functionalities to book a slot for vaccination, view the bookings made by the user etc.

- This application automatically creates a database named "covidvaccinationbooking" in the local host having a default port number 3306 and uses that database.
- The program also creates 3 tables in that database named Vaccine details, User, and User Bookings
- The default admin username is "admin" and password is "admin" - case sensitive.

The console application has two users:

Type of Users:

- Admin
- Users

Admin Use Cases:

- Login
- Add Vaccination Center
- Remove Vaccination Center
- Get Dosage Details
- Logout

User Use Cases:

- User Sign-up
- User Login
- Get vaccination center details (Center name and working hours)
- Apply for a vaccination slot
- View user bookings
- Logout

How to run this application?

- Install Java and MySql
- Install a Java IDE (Eclipse, NetBeans etc)
- Install mysql-connector-java.jar file to connect Java with MySql
- Create a Java project with a class named “covidVaccinationBooking”.
- Add mysql-connector-java.jar as referenced library to the Java project
- Copy this code to the main file
- Run java code

ADMIN USE CASES:

- The admin must first login using default username and password (username and password are case sensitive).

- Then it will show a list of actions that an admin can do.

  1) Add Vaccination Center
     a. The admin can add new vaccination centers by providing the center name, starting and ending time of the center (working hours), dosage details includes the name of the vaccine present in the center, and the number of slots available as inputs. The details are stores in the “vaccine details” table.

  2) Remove Vaccination Center
     a. The admin can remove the existing vaccination center by providing the center name as input. The details of the center are removed from the “vaccine details” table.

  3) Get Dosage Details
     a. The admin can get the dosage details, i.e., the names of the vaccine available in the center. The required output is displayed from the “vaccine details” table.

USER USE CASES:

User Sign-up:

- The user can sign-up by providing the username and the password as input. It is stored in the user table. And the user is requested to login.

User Login:

- The user can login by using their username and password. If the username is present in the user table, then the password is validated and the user is logged in.

- Then it will show the list of actions that a user can do:

  1) Get vaccination center details (Center name and working hours)
     a. The user can get the working hours of the vaccination center by providing the center’s name as input. The output is fetched from the vaccine details table.

  2) Apply for a vaccination slot:
     a. The user can apply for a slot in the available centers by selecting the required center and the vaccine name. The slot is booked and the username and details is inserted in the userbookings table, if the slot is available.

  3) View user bookings:
     a. The user can view their vaccination slot booking made by them. The data is fetched from the userbookings table and displayed.

Console Application Code:

```
import java.util.*;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.JTableHeader;
import java.awt.Color;
import java.awt.Font;
import java.sql.*;

public class covidVaccinationBooking {
    private static Connection con;
    private static PreparedStatement pstmt;
    private static Statement stmt;
    private static ResultSet res;

    public static void main(String[] args) {
      System.out.println("\n-------- WELCOME TO COVID VACCINATION BOOKING SYSTEM --------");
      Scanner sc = new Scanner(System.in);

      try {
          con = DriverManager.getConnection("jdbc:mysql://localhost:3306/","root","Vishwa@222");
          stmt = con.createStatement();

          //creating database

          String str = "show databases";
          res = stmt.executeQuery(str);
          int flag = 1;
          while(res.next()) {
                if(res.getString("Database").equalsIgnoreCase("CovidVaccinationBooking")) {
                      flag = 0;
                }
          }
          if(flag == 1) {
                stmt.executeUpdate("create database CovidVaccinationBooking");
          }

          // creating tables

          stmt.executeUpdate("use CovidVaccinationBooking");

          // user table

          res = stmt.executeQuery("show tables");
          flag = 1;
          while(res.next()) {
                if(res.getString("Tables_in_CovidVaccinationBooking").equalsIgnoreCase("user")) {
                      flag  = 0;;
                }
          }
          if(flag == 1) {
                stmt.executeUpdate("create table user(username varchar(10) unique, password varchar(10))");
          }

          // vaccine_details table

          res = stmt.executeQuery("show tables");
          flag = 1;
```

```
        while(res.next()) {
              if(res.getString("Tables_in_CovidVaccinationBooking").equalsIgnoreCase("vaccine_details")) {
                    flag  = 0;;
              }
        }
        if(flag == 1) {
              stmt.executeUpdate("create table vaccine_details(center_name varchar(50), start_time time, end_time time, 
dosage_details varchar(30), availability_of_slots int)");
        }

        // user booking table :

        res = stmt.executeQuery("show tables");
        flag = 1;
        while(res.next()) {
              if(res.getString("Tables_in_CovidVaccinationBooking").equalsIgnoreCase("userBookings")) {
                    flag  = 0;;
              }
        }
        if(flag == 1) {
              stmt.executeUpdate("create table userBookings(user_name varchar(50), center_name varchar(50), 
dosage_details varchar(30))");
        }

        // Console Application

        while(true) {
           System.out.println("\n-------- MAIN MENU --------");
           System.out.println("1) Admin Login");
           System.out.println("2) User Sign-up");
           System.out.println("3) User Login");
           System.out.println("4) Exit");
           System.out.print("\nEnter your choice : ");

           int loginChoice = sc.nextInt();


           // Admin Login

           if(loginChoice == 1) {

              String userName, password;

              System.out.print("\nEnter the username : ");
              userName = sc.next();

              System.out.print("Enter the password : ");
              password = sc.next();

              if(userName.equals("admin") && password.equals("admin")) {
                 flag = 1;
              }

              if(flag == 1) {
                 System.out.println("Successful login!");
              }
              if(flag == 0) {
                 System.out.println("\nUser Not Found!");
                 continue;
              }
```

```
while(true) {
   System.out.println("\n-------- ADMIN MENU --------");
   System.out.println("1) Add Vaccination Centers");
   System.out.println("2) Remove Vaccination Centers");
   System.out.println("3) Get Dosage Details");
   System.out.println("4) Logout");
   System.out.print("Enter your choice : ");

   int adminChoice = sc.nextInt();


   // Add Vaccination Center:

   if(adminChoice == 1) {
      String center, startTime, endTime, dosage;
      int slots;

      System.out.println("\nEnter details of new Vaccination center :");

      sc.nextLine();

      pstmt = con.prepareStatement("insert into vaccine_details values(?,?,?,?,?)");
      System.out.print("\n-> Enter the center name : ");
      center = sc.nextLine();

      System.out.print("-> Enter the start time (HH:MM:SS) : ");
      startTime = sc.next();
      System.out.print("-> Enter the end time (HH:MM:SS) : ");
      endTime = sc.next();
      sc.nextLine();
      System.out.print("-> Enter the dosage_details (vaccine name) : ");
      dosage = sc.nextLine();
      System.out.print("-> Enter the availability_of_slots : ");
      slots = sc.nextInt();

      pstmt.setString(1, center);
      pstmt.setString(2, startTime);
      pstmt.setString(3, endTime);
      pstmt.setString(4, dosage);
      pstmt.setInt(5, slots);
      pstmt.executeUpdate();

      System.out.println("\nNew center successfully created!");
   }


   // Remove Vaccination Center:

   else if(adminChoice == 2) {
      String center;

      displayCenter();

      ArrayList<String> list = new ArrayList<>();
      res = pstmt.executeQuery("select distinct center_name from vaccine_details");

      while(res.next()) {
            list.add(res.getString(1));
      }
```

```
          pstmt = con.prepareStatement("delete from vaccine_details where center_name = ?");

          System.out.print("\n-> Enter the center name to remove :");
          sc.nextLine();
          center = sc.nextLine();

          pstmt.setString(1, center);

          if(list.contains(center)) {
              pstmt.executeUpdate();
              System.out.println("\nCenter successfully removed!");
          }
          else {
              System.out.println("\nCenter not found!");
              continue;
          }

        }


        // Get Dosage Details :

        else if(adminChoice == 3) {
          String center;

          displayCenter();

          PreparedStatement pstmt = con.prepareStatement("select center_name, dosage_details from
vaccine_details where center_name = ?");

          System.out.print("\n-> Enter the center name to get dosage details : ");
          sc.nextLine();
          center = sc.nextLine();

          pstmt.setString(1, center);
          res = pstmt.executeQuery();

          ArrayList<ArrayList<String>> list = new ArrayList<>();
          int index = 0;
          while(res.next()) {
              list.add(new ArrayList<String>());
              list.get(index).add(res.getString(1));
              list.get(index).add(res.getString(2));
              index++;
          }

          if(list.isEmpty()) {
              System.out.println("Center not available!");
              continue;
          }

          String data[][]= new String[list.size()][list.get(0).size()];
          for(int i = 0; i < list.size(); i++) {
            for(int j = 0; j < list.get(i).size(); j++) {
                data[i][j] = list.get(i).get(j);
            }
          }

          String column[]={"Center Name" ,"Dosage Details"};
```

```
            JFrame f = new JFrame("Dosage details");
                f.setBounds(450,150,500,700);

            JTable table = new JTable(data,column);

            table.setRowHeight(35);
            table.setBounds(100,100,500,700);
            table.setBackground(new Color(255, 135, 135));

            JTableHeader tableHeader = table.getTableHeader();
            tableHeader.setBackground(Color.black);
            tableHeader.setForeground(Color.white);
            Font headerFont = new Font("Verdana", Font.BOLD, 14);
            tableHeader.setFont(headerFont);

            table.setFont(new Font("Times New Roman", Font.BOLD,15));

            JScrollPane sp=new JScrollPane(table);

            f.add(sp);
            f.setSize(600,400);
            f.setVisible(true);
        }


        // Admin Logout:

        else if(adminChoice == 4) {
            System.out.println("\n" + userName + " Logged Out");
            break;
        }

        else {
            System.out.println("Incorrect Option! Start from begining.");
            continue;
        }
    }
}


// User Sign-up:

else if(loginChoice == 2) {
    String userName, password;
    pstmt = con.prepareStatement("insert into user values(?,?)");

    System.out.print("\nEnter the new username to create : ");
    userName = sc.next();

    System.out.print("Enter the password : ");
    password = sc.next();

    pstmt.setString(1, userName);
    pstmt.setString(2, password);
    pstmt.executeUpdate();

    System.out.println("\nUser Successfully Created!");
    System.out.println("Please login!");

}
```

```
// User Login:

else if(loginChoice == 3) {
    String userName, password;

    System.out.print("\nEnter the username : ");
    userName = sc.next();
    System.out.print("Enter the password : ");
    password = sc.next();

    res = stmt.executeQuery("select * from user");
    flag = loginValidation(res, userName, password);

    if(flag == 1) {
            System.out.println("Successful login!");
    }
    if(flag == 0) {
            System.out.println("\nUser Not Found!");
            continue;
    }


    while(true) {
        System.out.println("\n-------- User MENU --------");
        System.out.println("1) Get vaccination center details");
        System.out.println("2) Apply for a vaccination slot");
        System.out.println("3) View your bookings");
        System.out.println("4) Logout");
        System.out.print("Enter your choice : ");

        int userChoice = sc.nextInt();


        // Get vaccination center details:

        if(userChoice == 1) {
            String center;

            displayCenter();

            pstmt = con.prepareStatement("select distinct center_name, start_time, end_time from vaccine_details where center_name = ?");
            System.out.print("\n-> Enter the center name to get details : ");
            sc.nextLine();
            center = sc.nextLine();

            pstmt.setString(1, center);
            res = pstmt.executeQuery();

            ArrayList<ArrayList<String>> list = new ArrayList<>();
            int index = 0;
            while(res.next()) {
                    list.add(new ArrayList<String>());
                    list.get(index).add(res.getString(1));
                    list.get(index).add(res.getString(2));
                    list.get(index).add(res.getString(3));
                    index++;
            }
```

```
                if(list.isEmpty()) {
                        System.out.println("Center not available");
                        continue;
                }

                String data[][]= new String[list.size()][list.get(0).size()];
                for(int i = 0; i < list.size(); i++) {
                    for(int j = 0; j < list.get(i).size(); j++) {
                        data[i][j] = list.get(i).get(j);
                    }
                }

                String column[]={"Center Name" ,"Start Time", "End Time"};
                JFrame f = new JFrame("Center details");
                    f.setBounds(450,150,500,700);

                JTable table = new JTable(data,column);
                table.setRowHeight(35);
                table.setBounds(100,100,500,700);
                table.setBackground(new Color(255, 135, 135));
                JTableHeader tableHeader = table.getTableHeader();
                tableHeader.setBackground(Color.black);
                tableHeader.setForeground(Color.white);
                Font headerFont = new Font("Verdana", Font.BOLD, 14);
                tableHeader.setFont(headerFont);
                table.setFont(new Font("Times New Roman", Font.BOLD,15));
                JScrollPane sp=new JScrollPane(table);

                f.add(sp);
                f.setSize(600,400);
                f.setVisible(true);
            }

            // Apply for a vaccination slot

            else if(userChoice == 2) {
                String center, dosage;
                displayCenter();

                pstmt = con.prepareStatement("select distinct center_name, dosage_details, availability_of_slots from vaccine_details where center_name = ?");
                System.out.print("\n-> Enter the center name to get details : ");
                sc.nextLine();
                center = sc.nextLine();
                pstmt.setString(1, center);
                res = pstmt.executeQuery();
                System.out.println();

                int temp = 0;
                while(res.next()) {
                    temp++;
                }

                if(temp == 0) {
                    System.out.println("Center not Available");
                    continue;
                }

                res = pstmt.executeQuery();
                System.out.print("Vaccines availbale at "+ center +" : \n");
                while(res.next()) {
                    System.out.println(res.getString("dosage_details"));
                }

                System.out.print("\nChoose the vaccine : ");

                pstmt = con.prepareStatement("select availability_of_slots from vaccine_details where dosage_details
    = ? and center_name = ?");
                dosage = sc.nextLine();
                pstmt.setString(1, dosage);
                pstmt.setString(2, center);
                res = pstmt.executeQuery();

                temp = 0;
                while(res.next()) {
                    temp = res.getInt("availability_of_slots");
                }

                if(temp>0) {
                    temp--;
                    System.out.println("Slot booked successfully!");
                    pstmt = con.prepareStatement("update vaccine_details set availability_of_slots = ? where
    dosage_details = ? and center_name = ?");

                    pstmt.setInt(1, temp);
                    pstmt.setString(2, dosage);
                    pstmt.setString(3, center);
                    pstmt.executeUpdate();

                    pstmt = con.prepareStatement("insert into userBookings values(?,?,?)");
                    pstmt.setString(1, userName);
                    pstmt.setString(2, center);
                    pstmt.setString(3, dosage);
                    pstmt.executeUpdate();

                }
                else {
                    System.out.println("Slots are full for "+center+"! Please try another vaccine/center");
                    continue;
                }

            }

            // View your bookings

            else if(userChoice == 3) {
                pstmt = con.prepareStatement("select * from userBookings where user_name = ?");
                pstmt.setString(1, userName);

                res = pstmt.executeQuery();
                System.out.println();

                ArrayList<ArrayList<String>> list = new ArrayList<>();
                int index = 0;
                while(res.next()) {
                    list.add(new ArrayList<String>());
                    list.get(index).add(res.getString(1));
                    list.get(index).add(res.getString(2));
                    list.get(index).add(res.getString(3));
                    index++;
                }

                if(list.isEmpty()) {
                    System.out.println("No bookings made by this user");
                    continue;
                }

                System.out.println();

                String data[][]= new String[list.size()][list.get(0).size()];
                for(int i = 0; i < list.size(); i++) {
                    for(int j = 0; j < list.get(i).size(); j++) {
                        data[i][j] = list.get(i).get(j);
                    }
                }

                String column[]={"User Name" ,"Center Name", "Dosage Details"};
                JFrame f = new JFrame("User Booking details");
                    f.setBounds(450,150,500,700);
                JTable table = new JTable(data,column);
                table.setRowHeight(35);
                table.setBounds(100,100,500,700);
                table.setBackground(new Color(255, 135, 135));
                JTableHeader tableHeader = table.getTableHeader();
                tableHeader.setBackground(Color.black);
                tableHeader.setForeground(Color.white);
                Font headerFont = new Font("Verdana", Font.BOLD, 14);
                tableHeader.setFont(headerFont);
                table.setFont(new Font("Times New Roman", Font.BOLD,15));

                JScrollPane sp=new JScrollPane(table);

                f.add(sp);
                f.setSize(600,400);
                f.setVisible(true);

            }

            // User Logout

            else if(userChoice == 4) {
                System.out.println("\n" + userName + " Logged Out");
                break;
            }

            else {
                System.out.println("Incorrect Option! Start from begining.");
                continue;
            }
          }

        }

        // Exit Application

        else if(loginChoice == 4) {
          System.out.println("Thank you!");
          break;
        }
```

```
            else {
               System.out.println("\nIncorrect Option!");
               continue;
            }
         }
      }

      catch(Exception e){
         System.out.println(e);
      }

      finally {
         sc.close();
      }
   }

   private static int loginValidation(ResultSet res, String userName, String password) throws SQLException {
      while(res.next()) {
            if(res.getString("username").equals(userName) && res.getString("password").equals(password)) {
                  return 1;
            }
      }
      return 0;
   }

   private static void displayCenter() throws SQLException {
      pstmt = con.prepareStatement("select distinct center_name from vaccine_details");
      res = pstmt.executeQuery();
      System.out.println("\nAvailable Centers : ");
      while(res.next()) {
            System.out.println(res.getString(1));
      }
   }
}
```