DRUG PERFORMANCE ANALYSIS

A
Mini Project Report

Submitted in the partial fulfilment of the

Requirements for the Degree of **BACHELOR**

OF ENGINEERING IN

INFORMATION TECHNOLOGY

BY

IMRAN – 1602-21-737-097

MIRZA – 1602-21-737-095

CHAITANYA– 1602-21-737-078

ABHIRAM – 1602-21-737-066

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahim Bagh, Hyderabad-31

2021

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahim Bagh, Hyderabad-31

2023

DECLARATION BY CANDIDATE

We, IMRAN, MIRZA, CHAITANYA, ABHIRAM bearing hall ticker numbers, 1602-21-737-097, 1602-21-737-095, 1602-21-737-078, 1602-21-737-066, hereby declare that the project report titled for the award of the degree of Bachelor of Engineering in Information Technology.

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

IMRAN

1602-21-737-097

MIRZA

1602-21-737-095

CHAITANYA

1602-21-737-078

ABHIRAM

1602-21-737-066

(Faculty in-charge) (Head Dept. IT)

ACKNOWLEDGEMENTS

We are overwhelmed in all humbleness and gratefulness to acknowledge our debt to all those who have helped us put these ideas, well above the level of simplicity and into something concrete. We are extremely thankful to our college, Vasavi College of Engineering, Hyderabad for providing the opportunity to implement our project, "Drug Performance Analysis" which helped us in learning many new things.

We would like to express our gratitude to Ms. Sidam Aruna, Associate Professor, Vasavi College of Engineering, for the esteemed guidance, moral support and invaluable advice provided by her for the success of the project.

We are also thankful for our friends and peers for their help and encouragement throughout the project.

IMRAN

1602-21-737-097

MIRZA

1602-21-737-095

CHAITANYA

1602-21-737-078

ABHIRAM

1602-21-737-066

Date: 04-07-2023

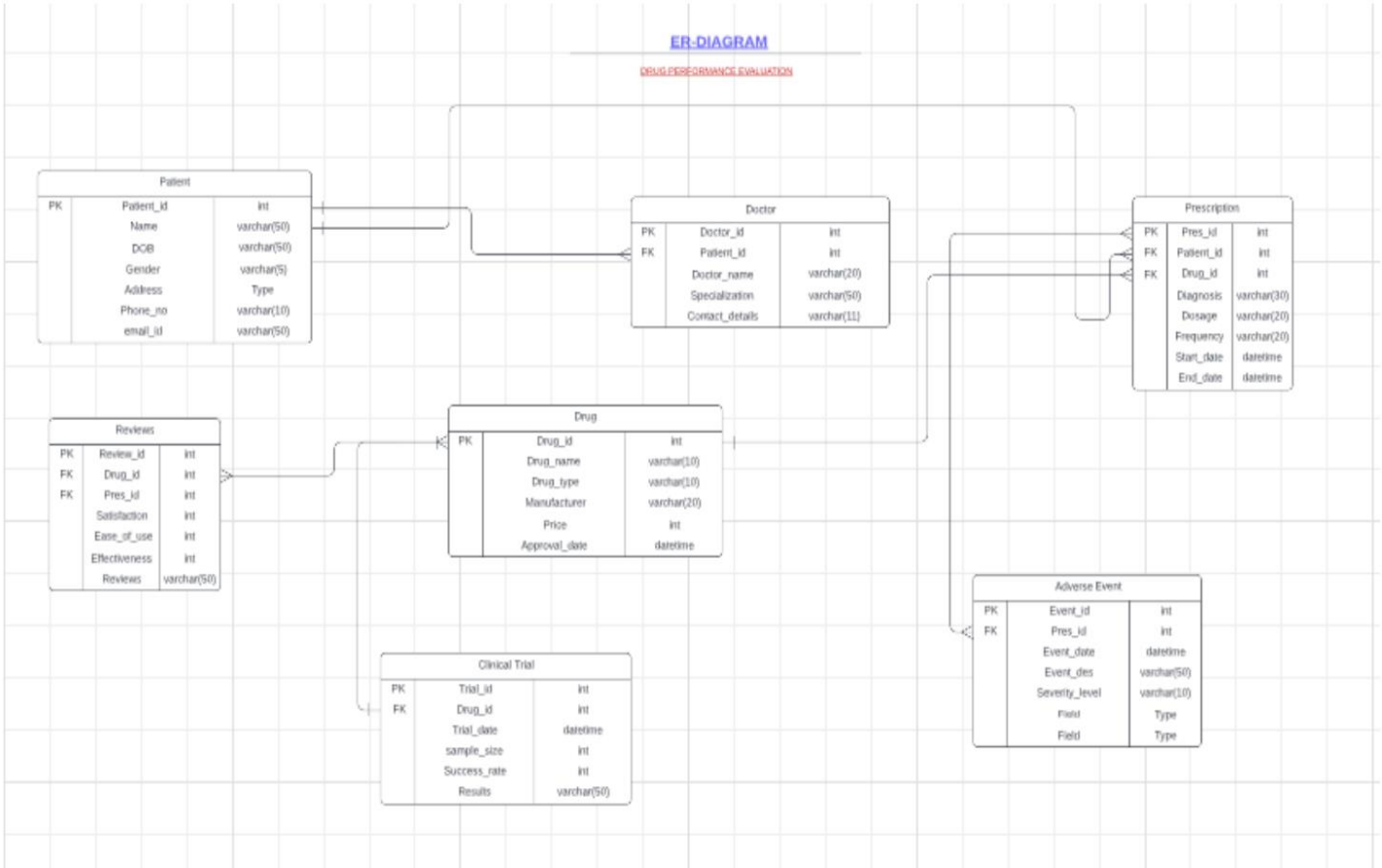
ABSTRACT

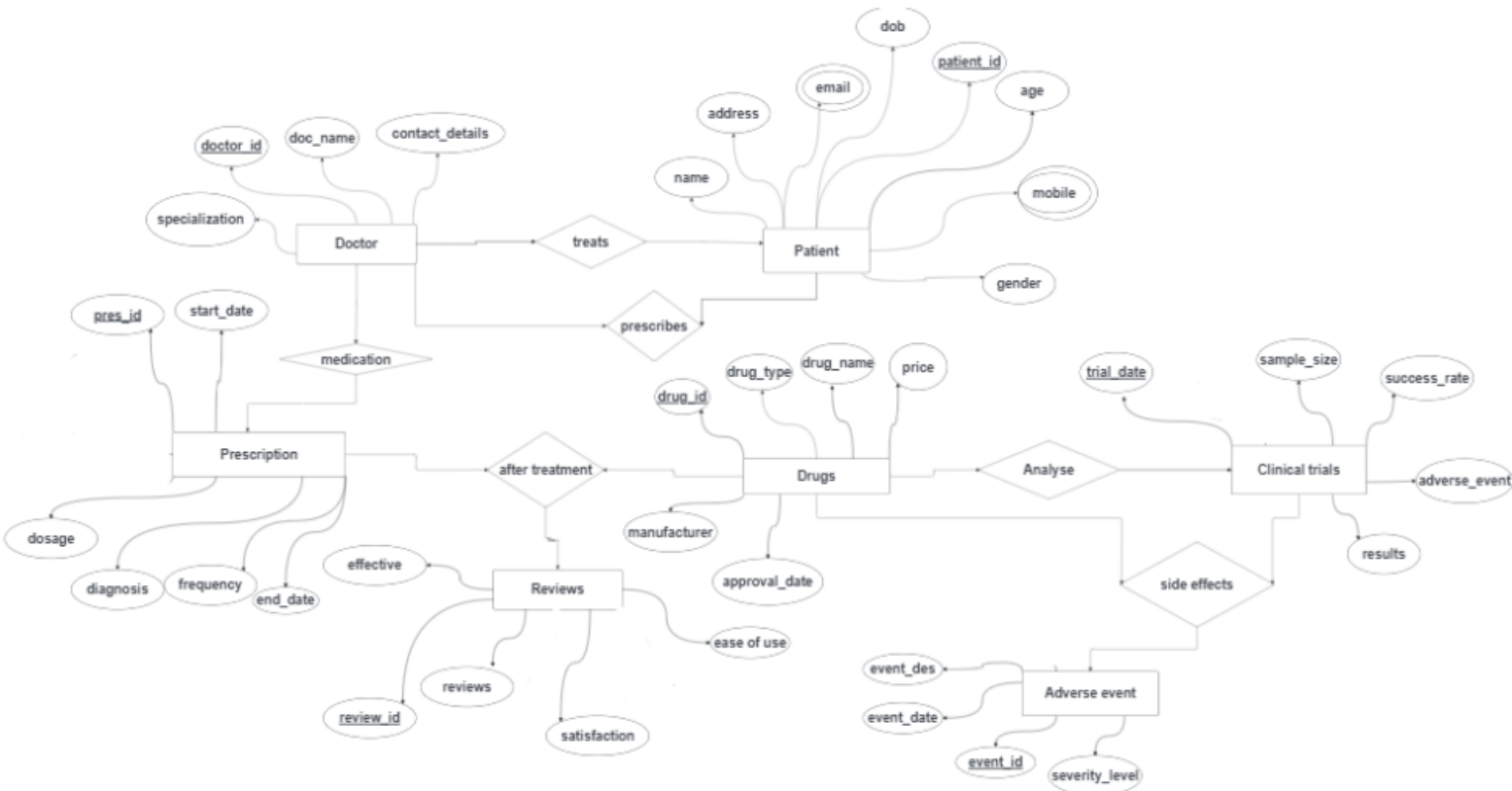
The effective evaluation of drug performance plays a pivotal role in optimizing therapeutic outcomes and improving patient care. This project aims to utilize data-driven analysis techniques to assess and analyze the performance of pharmaceutical drugs in various therapeutic areas. By leveraging large-scale data sources, including clinical trials, electronic health records, and pharmacovigilance databases, we aim to identify patterns, trends, and factors influencing drug performance

The project begins with a comprehensive data collection phase, where relevant datasets are obtained and curated. These datasets encompass a wide range of variables, including patient demographics, treatment protocols, drug dosages, adverse events, and treatment responses. Following data preprocessing and cleaning, advanced statistical and machine learning algorithms are applied to extract meaningful insights and patterns.

Through our analysis, we aim to identify drugs that demonstrate superior efficacy, safety profiles, and tolerability. We examine factors such as drug-drug interactions, patient characteristics, and treatment adherence, which influence the overall therapeutic outcome. Moreover, we investigate the impact of various interventions, such as combination therapies and dose adjustments, on drug performance.

ER DIAGRAM


ER-DIAGRAM
DRUG PERFORMANCE EVALUATION
Patient
PK Patient_id int
Name varchar(50)
DOB varchar(50)
Gender varchar(5)
Address Type
Phone_no varchar(10)
email_id varchar(50)
Doctor
PK Doctor_id int
FK Patient_id int
Doctor_name varchar(20)
Specialization varchar(50)
Contact_details varchar(11)
Prescription
PK Pres_id int
FK Patient_id int
FK Drug_id int
Diagnosis varchar(30)
Dosage varchar(20)
Frequency varchar(20)
Start_date datetime
End_date datetime
Reviews
PK Review_id int
FK Drug_id int
FK Pres_id int
Satisfaction int
Ease_of_use int
Effectiveness int
Reviews varchar(50)
Drug
PK Drug_id int
Drug_name varchar(10)
Drug_type varchar(10)
Manufacturer varchar(20)
Price int
Approval_date datetime
Adverse Event
PK Event_id int
FK Pres_id int
Event_date datetime
Event_des varchar(50)
Severity_level varchar(10)
Field Type
Field Type
Clinical Trial
PK Trial_id int
FK Drug_id int
Trial_date datetime
sample_size int
Success_rate int
Results varchar(50)


dob
patient_id
email
age
address
doctor_id
doc_name
contact_details
name
mobile
specialization
Doctor
treats
Patient
gender
pres_id
start_date
prescribes
medication
drug_type
drug_name
price
trial_date
sample_size
success_rate
drug_id
Prescription
after treatment
Drugs
Analyse
Clinical trials
adverse_event
dosage
manufacturer
results
effective
Reviews
approval_date
side effects
diagnosis
frequency
end_date
ease of use
event_des
reviews
Adverse event
review_id
event_date
satisfaction
event_id
severity_level

CARDINALITY

Mapping cardinalities:

- One doctor can have zero to many patients. (1:N)
- One patient can have zero to many prescriptions. (1:N)
- One drug can have zero to many prescriptions. (1:N)
- One drug can have zero to many clinical trials. (1:N)
- One drug can have zero to many reviews. (1:N)
- One prescription can have zero to many adverse events (1:N)

TABLES USED IN DATABASE

PRESCRIPTION TABLE :

```
SQL> desc prescription
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRES_ID                                   NOT NULL NUMBER(3)
 USERNAME                                           VARCHAR2(30)
 DOC_ID                                             NUMBER(3)
 DRUG_ID                                            NUMBER(3)
 DIAGNOSIS                                          VARCHAR2(500)
 DATE_ADDED                                         DATE
```

DRUG TABLE :

```
SQL> desc drug
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DRUG_ID                                   NOT NULL NUMBER(3)
 DRUG_NAME                                 NOT NULL VARCHAR2(100)
 DRUG_DESCRIPTION                          NOT NULL VARCHAR2(500)
 PRICE                                     NOT NULL NUMBER(10,2)
 CATEGORY                                  NOT NULL VARCHAR2(50)
 DATE_ADDED                                NOT NULL DATE
```

DOCTOR TABLE :

```
SQL> desc doctor
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DOC_ID                                    NOT NULL NUMBER(3)
 DOC_NAME                                  NOT NULL VARCHAR2(255)
 SPECIALIZATION                            NOT NULL VARCHAR2(255)
 CONTACT_DETAILS                                    VARCHAR2(255)
```

SIDEEFFECTS TABLE :

```
SQL> desc side_effects
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SIDE_EFFECT_ID                            NOT NULL NUMBER(38)
 DRUG_ID                                            NUMBER(38)
 USERNAME                                           VARCHAR2(30)
 DATE_ADDED                                         DATE
 SIDE_EFFECTS_DESCRIPTION                           VARCHAR2(500)
 SYMPTOM                                            VARCHAR2(200)
```

USER TABLE :

```
SQL> desc user_details
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERNAME                                  NOT NULL VARCHAR2(30)
 PASSWORD                                  NOT NULL VARCHAR2(225)
 FULLNAME                                  NOT NULL VARCHAR2(225)
 PHONE                                     NOT NULL VARCHAR2(10)
 LAST_LOGIN                                         DATE
 ISLOGGEDIN                                         VARCHAR2(1)
```

REVIEW TABLE :

```
SQL> desc review
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 REVIEW_ID                                 NOT NULL NUMBER(38)
 DRUG_ID                                            NUMBER(38)
 USERNAME                                           VARCHAR2(30)
 REVIEW_TEXT                                        VARCHAR2(1000)
 OVERALL_RATING                                     NUMBER(38)
 RECOMMEND                                          CHAR(1)
 REVIEW_DATE                                        DATE
```

DDL OPERATIONS :

```
SQL> CREATE TABLE Patient (
patient_id INT PRIMARY KEY,
name VARCHAR(50) NOT NULL,
date_of_birth DATE NOT NULL,
gender CHAR(1) NOT NULL,
address VARCHAR(100),
phone_number VARCHAR(20),
email VARCHAR(50) );

Table created.

SQL> CREATE TABLE Doctor (
doctor_id INT PRIMARY KEY,
doctor_name VARCHAR(50) NOT NULL,
specialization VARCHAR(50) NOT NULL,
contact_details VARCHAR(100) );

Table created.

SQL> CREATE TABLE Drug (
drug_id INT PRIMARY KEY,
drug_name VARCHAR(50) NOT NULL,
drug_type VARCHAR(50) NOT NULL,
manufacturer VARCHAR(50) NOT NULL,
approval_date DATE NOT NULL,
side_effects VARCHAR(200),
price DECIMAL(10,2),
form VARCHAR(50) NOT NULL,
type VARCHAR(50) NOT NULL );

Table created.
```

```
SQL>
CREATE TABLE Prescription (
prescription_id INT PRIMARY KEY,
patient_id INT ,
drug_id INT,
 diagnosis VARCHAR(200),
dosage VARCHAR(50),
frequency VARCHAR(50),
start_date DATE, end_date DATE, FOREIGN
KEY (patient_id) REFERENCES
Patient(patient_id), FOREIGN KEY (drug_id)
REFERENCES Drug(drug_id) );

Table created.


 SQL>
 CREATE TABLE Reviews (
 ReviewID INT PRIMARY KEY,
  DrugID INT,
  ConditionID INT,
  Satisfaction INT,
  EaseOfUse INT,
  Effective INT,
  Reviews VARCHAR(500),
  FOREIGN KEY (DrugID) REFERENCES Drug(drug_id) );

 Table created.
```

```
SQL>
CREATE TABLE Clinical_trial (
trial_id INT PRIMARY KEY,
drug_id INT,
trial_date DATE,
sample_size INT,
success_rate FLOAT,
adverse_events VARCHAR(500),
results VARCHAR(500),
FOREIGN KEY (drug_id) REFERENCES Drug(drug_id) );

Table created.


SQL>
CREATE TABLE Adverse_event (
event_id INT PRIMARY KEY,
prescription_id INT,
event_date DATE,
event_description VARCHAR(200),
severity_level VARCHAR(50),
FOREIGN KEY (prescription_id) REFERENCES
Prescription(prescription_id) );
Table created.
```

DML OPERATIONS :

1.QUERY TO INSERT VALUES IN THE PRESCRIPTION TABLE.

```
SQL> INSERT INTO  PRESCRIPTION1 VALUES(789,66,402,'cold','250MG','2 PER DAY','01-dec-2002','17-dec-2002');


1 row created.


SQL> INSERT INTO  PRESCRIPTION1 VALUES(790,66,403,'headache','1000MG','1 PER DAY','10-nov-2007','17-nov-2007');


1 row created.
```

2.QUERY TO DISPLAY THE PRESCRIPTION TABLE.

```
SQL> select * from prescription1;
PRESCRIPTION_ID PATIENT_ID    DRUG_ID DIAGNOSIS
DOSAGE                              FREQUENCY                      START_DAT END_DATE

         788         66       401 FEVER
     500MG                              1 PER DAY                      01-MAY-23 07-MAY-23
         789         66       402 cold
     250MG                              2 PER DAY                      01-DEC-02 17-DEC-02
         790         66       403 headache
     1000MG                              1 PER DAY                      10-NOV-07 17-NOV-07
```

3.QUERY TO UPDATE THE PRESCRIPTION ID IN THE PRESCRIPTION TABLE WHERE DIAGNOSIS =’FEVER’.

```
SQL> update prescription1 set prescription_id=&prescription_id where diagnosis='FEVER';
Enter value for prescription_id: 787
old   1: update prescription1 set prescription_id=&prescription_id where diagnosis='FEVER'
new   1: update prescription1 set prescription_id=791 where diagnosis='FEVER'

1 rows updated.
```

4.QUERY TO DELETE THE ROW IN THE PRESCRIPTION TABLE WHERE PRESCRIPTION ID =789.

```
SQL> delete from prescription1 where prescription_id=789;

1 row deleted.
```

```
SQL> select * from prescription1;

PRESCRIPTION_ID PATIENT_ID    DRUG_ID DIAGNOSIS
    DOSAGE                              FREQUENCY                       START_DAT END_DATE
       787       66       401 FEVER
    500MG                               1 PER DAY                       01-MAY-23 07-MAY-23
       790       66       403 headache
    1000MG                              1 PER DAY                       10-NOV-07 17-NOV-07
```

5.QUERY TO ALTER THE FREQUENCY COLUMN BY REDUCING SIZE FROM VARCHAR2(50) TO VARCHAR2(20).

```
SQL>ALTER TABLE PRESCRIPTION MODIFY FREQUENCY VARCHAR2(20);
Table altered.
SQL> DESC PRESCRIPTION;
 Name                          Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRESCRIPTION_ID               NOT NULL NUMBER(38)
 PATIENT_ID                             NUMBER(20)
 DRUG_ID                                NUMBER(38)
 DIAGNOSIS                              VARCHAR2(200)
```

DOSAGE	VARCHAR2(50)
FREQUENCY	VARCHAR(50)
START_DATE	DATE
END_DATE	DATE

Code for Home Page:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class HomePage extends JFrame {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            HomePage homepage = new HomePage();
            homepage.createAndShowGUI();
        });
    }
    public void createAndShowGUI() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setTitle("Project Homepage");
    setSize(800, 600);
    setLocationRelativeTo(null);

    // Create the layout components
    JPanel welcomeInfoImagePanel = new JPanel();
    JPanel topPanel = new JPanel();
    JPanel infoImagePanel = new JPanel();
    JPanel infoPanel = new JPanel();
    JPanel imagePanel = new JPanel();
```

```
// Create the buttons
    JButton homeButton = new JButton("Home");
    JButton reviewButton = new JButton("Review");
    JButton profileButton = new JButton("Profile");
    JButton logoutButton = new JButton("Logout");
    JButton exitButton = new JButton("Exit");

    // Set the button colors
    Color buttonColor = new Color(17, 30, 108);
    Color buttonHoverColor = new Color(175, 238, 238);

    homeButton.setBackground(buttonColor);
    homeButton.setForeground(Color.WHITE);
    homeButton.setFocusPainted(false);
    homeButton.addMouseListener(new
ButtonHoverListener(homeButton, buttonColor, buttonHoverColor));
    reviewButton.setBackground(buttonColor);
    reviewButton.setForeground(Color.WHITE);
    reviewButton.setFocusPainted(false);
    reviewButton.addMouseListener(new
ButtonHoverListener(reviewButton, buttonColor,
buttonHoverColor));
    profileButton.setBackground(buttonColor);
    profileButton.setForeground(Color.WHITE);
    profileButton.setFocusPainted(false);
    profileButton.addMouseListener(new
ButtonHoverListener(profileButton, buttonColor, buttonHoverColor));
    logoutButton.setBackground(buttonColor);
    logoutButton.setForeground(Color.WHITE);
    logoutButton.setFocusPainted(false);
    logoutButton.addMouseListener(new
ButtonHoverListener(logoutButton, buttonColor, buttonHoverColor));
```

```
exitButton.setBackground(buttonColor);
    exitButton.setForeground(Color.WHITE);
    exitButton.setFocusPainted(false);
    exitButton.addMouseListener(new
ButtonHoverListener(exitButton, buttonColor, buttonHoverColor));

    JLabel welcomeLabel = new JLabel("<html><body><font
size='6'>Welcome!</font></body></html>");

    // Create the project info label
    JLabel infoLabel = new JLabel("<html><body><font size='5'>This
is a project that allows users to search for medicine reviews, "
            + "insert new reviews, delete existing reviews, and update
their own reviews.<br><br>"
            + "Please explore the various functionalities
available!</font></body></html>");

    infoLabel.setHorizontalAlignment(SwingConstants.CENTER);

    // Create the image label
    JLabel imageLabel = new JLabel();
    ImageIcon imageIcon = new ImageIcon("image.jpg"); // Replace
"path_to_image.jpg" with your image path

imageIcon.setImage(imageIcon.getImage().getScaledInstance(390,
-1, Image.SCALE_SMOOTH)); // Adjust image size
    imageLabel.setIcon(imageIcon);
```

```
// Set the layouts
    welcomeInfoImagePanel.setLayout(new BorderLayout());
    topPanel.setLayout(new BoxLayout(topPanel, BoxLayout.X_AXIS));
    infoImagePanel.setLayout(new GridLayout(1, 2));
    infoPanel.setLayout(new BorderLayout());
    imagePanel.setLayout(new BorderLayout());

    // Adjust the welcome message panel
    JPanel welcomePanel = new JPanel(new
FlowLayout(FlowLayout.CENTER));
    welcomePanel.add(welcomeLabel);
    welcomePanel.setBorder(BorderFactory.createEmptyBorder(10, 10,
10, 10)); // Add padding around the label

    // Set background color for panels
    topPanel.setBackground(Color.WHITE);
    welcomePanel.setBackground(new Color(224, 236, 255));
    infoPanel.setBackground(Color.WHITE);
    imagePanel.setBackground(new Color(224, 236, 255));

    // Add components to their respective containers
    topPanel.add(homeButton);
    topPanel.add(reviewButton);
    topPanel.add(profileButton);
    topPanel.add(logoutButton);
    topPanel.add(Box.createHorizontalGlue()); // Add horizontal glue to
right-align the exit button
    topPanel.add(exitButton);
    topPanel.setBackground(new Color(224, 236, 255));
```

```
welcomeInfoImagePanel.add(topPanel, BorderLayout.NORTH);
   welcomeInfoImagePanel.add(welcomePanel,
BorderLayout.CENTER);

   infoPanel.add(infoLabel, BorderLayout.CENTER);
   imagePanel.add(imageLabel, BorderLayout.CENTER);

   infoImagePanel.add(infoPanel);
   infoImagePanel.add(imagePanel);
   infoPanel.setBackground(new Color(224, 236, 255));

   // Add panels to the frame
   getContentPane().setLayout(new BorderLayout());
   getContentPane().add(welcomeInfoImagePanel,
BorderLayout.NORTH);
   getContentPane().add(infoImagePanel, BorderLayout.CENTER);

   exitButton.addActionListener(e -> dispose());

      /*homeButton.addActionListener(e -> {
         dispose();
         // Create and show the Home screen
         HomePage homePage = new HomePage();
         homePage.createAndShowGUI();
      });*/
      reviewButton.addActionListener(e -> {
         dispose();
         // Create and show the Review screen
         Review review=new Review();
         review.createAndShowGUI();
```

```
profileButton.addActionListener(e -> {
            dispose();
            // Create and show the Profile screen
            Profile profilePage = new Profile();
            profilePage.createAndShowGUI();
        });
        logoutButton.addActionListener(e -> {
          dispose();
          LoginSignUpScreen loginSignUpScreen = new
LoginSignUpScreen();
          loginSignUpScreen.setVisible(true);
        });



    setVisible(true);
}


    private static class ButtonHoverListener extends
MouseAdapter {
        private final JButton button;
        private final Color originalColor;
        private final Color hoverColor;

        public ButtonHoverListener(JButton button, Color
originalColor, Color hoverColor) {
            this.button = button;
            this.originalColor = originalColor;
            this.hoverColor = hoverColor;
        }
```

```
 @Override
        public void mouseEntered(MouseEvent e) {
           button.setBackground(hoverColor);
        }

        @Override
        public void mouseExited(MouseEvent e) {
           button.setBackground(originalColor);
        }
    }
}
```

CODE FOR EDIT PROFILE :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.*;

public class EditProfile extends JFrame {

   private static final String JDBC_URL = "jdbc:oracle:thin:@localhost:1521:orcl"; // Update with your
database connection URL
   private static final String USERNAME = "sys as sysdba"; // Replace with your database username
   private static final String PASSWORD = "vasavi"; // Replace with your database password

   private JButton homeButton;
   private JButton reviewButton;
   private JButton profileButton;
   private JButton logoutButton;
   private JButton exitButton;

   private JLabel usernameValueLabel;
   private JTextField passwordTextField;
   private JTextField fullNameTextField;
   private JTextField phoneTextField;

   public static void main(String[] args) {
      SwingUtilities.invokeLater(EditProfile::new);
   }

   public EditProfile() {
      createAndShowGUI();
   }


    public void createAndShowGUI() {
         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
         setTitle("Edit Profile");
         setSize(800, 600);
         setLocationRelativeTo(null);

         // Create the layout components
         JPanel topPanel = new JPanel();
         JPanel infoPanel = new JPanel();
         JPanel titlePanel = new JPanel();
         JPanel buttonPanel = new JPanel();

         // Create the buttons
         homeButton = new JButton("Home");
         reviewButton = new JButton("Review");
         profileButton = new JButton("Profile");
         logoutButton = new JButton("Logout");
         exitButton = new JButton("Exit");
         JButton submitButton = new JButton("Submit");
```

```
// Set the button colors
     Color buttonColor = new Color(17, 30, 108);
    Color buttonHoverColor = new Color(102, 204, 255);
    setButtonStyles(homeButton, buttonColor, buttonHoverColor);
    setButtonStyles(reviewButton, buttonColor, buttonHoverColor);
    setButtonStyles(profileButton, buttonColor, buttonHoverColor);
    setButtonStyles(logoutButton, buttonColor, buttonHoverColor);
    setButtonStyles(exitButton, buttonColor, buttonHoverColor);

    JLabel editProfileLabel = new JLabel("<html><body><font size='6'>Edit Profile</font></body></html>");

    // Set the layouts
    topPanel.setLayout(new BoxLayout(topPanel, BoxLayout.X_AXIS));
    infoPanel.setLayout(new GridLayout(0, 2, 10, 10)); // GridLayout with 2 columns and 10px spacing
    titlePanel.setLayout(new BorderLayout());
    buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER));

    // Set background color for panels
    topPanel.setBackground(new Color(224, 236, 255));
    infoPanel.setBackground(new Color(224, 236, 255));
    titlePanel.setBackground(new Color(224, 236, 255)); // Light blue color
    buttonPanel.setBackground(new Color(224, 236, 255));

    // Add components to their respective containers
    topPanel.add(homeButton);
    topPanel.add(reviewButton);
    topPanel.add(profileButton);
    topPanel.add(logoutButton);
    topPanel.add(Box.createHorizontalGlue()); // Add horizontal glue to right-align the exit button
    topPanel.add(exitButton);

    titlePanel.add(editProfileLabel, BorderLayout.NORTH);


  // Add labels and text fields to infoPanel
        infoPanel.add(new JLabel("Username:"));
        usernameValueLabel = new JLabel();
        infoPanel.add(usernameValueLabel);
        infoPanel.add(new JLabel("Password:"));
        passwordTextField = new JTextField();
        infoPanel.add(passwordTextField);
        infoPanel.add(new JLabel("Full Name:"));
        fullNameTextField = new JTextField();
        infoPanel.add(fullNameTextField);
        infoPanel.add(new JLabel("Phone:"));
        phoneTextField = new JTextField();
        infoPanel.add(phoneTextField);

        // Set label and text field properties
        usernameValueLabel.setBorder(BorderFactory.createLineBorder(Color.GRAY, 1));
        usernameValueLabel.setHorizontalAlignment(SwingConstants.CENTER);
        usernameValueLabel.setOpaque(true);
        usernameValueLabel.setBackground(new Color(230, 230, 230)); // Light gray color
        usernameValueLabel.setForeground(Color.BLACK);
```

```
passwordTextField.setBorder(BorderFactory.createLineBorder(Color.GRAY, 1));
        fullNameTextField.setBorder(BorderFactory.createLineBorder(Color.GRAY, 1));
        phoneTextField.setBorder(BorderFactory.createLineBorder(Color.GRAY, 1));

        // Add submit button to buttonPanel
        buttonPanel.add(submitButton);

        // Add panels to the frame
        getContentPane().setLayout(new BorderLayout());
        getContentPane().add(topPanel, BorderLayout.NORTH);
        getContentPane().add(titlePanel, BorderLayout.CENTER);
        getContentPane().add(infoPanel, BorderLayout.CENTER);
        getContentPane().add(buttonPanel, BorderLayout.SOUTH);

        exitButton.addActionListener(e -> dispose());

        logoutButton.addActionListener(e -> {
            dispose();
            LoginSignUpScreen loginSignUpScreen = new LoginSignUpScreen();
            loginSignUpScreen.setVisible(true);
        });

        // Add action listeners to the buttons
        homeButton.addActionListener(e -> {
            dispose();
            HomePage homepage = new HomePage();
            homepage.createAndShowGUI();
        });

        reviewButton.addActionListener(e -> {
            dispose();
            Review review = new Review();
            review.createAndShowGUI();
        });

        profileButton.addActionListener(e -> {
            dispose();
            Profile profile = new Profile();
            profile.setVisible(true);
        });

        fetchUserDetails(); // Fetch user details from the database

        submitButton.addActionListener(e -> {
            updateProfile();
        });

        setVisible(true);
    }
```

```
private void setButtonStyles(JButton button, Color buttonColor, Color buttonHoverColor) {
        button.setBackground(buttonColor);
        button.setForeground(Color.WHITE);
        button.setFocusPainted(false);
        button.addMouseListener(new ButtonHoverListener(button, buttonColor, buttonHoverColor));
    }

    private void fetchUserDetails() {
        try {
            Connection connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);
            PreparedStatement preparedStatement = connection.prepareStatement("SELECT * FROM
 user_details WHERE isloggedin = 'Y'");
            ResultSet resultSet = preparedStatement.executeQuery();

            if (resultSet.next()) {
                String username = resultSet.getString("USERNAME");
                String password = resultSet.getString("PASSWORD");
                String fullName = resultSet.getString("FULLNAME");
                String phone = resultSet.getString("PHONE");

                usernameValueLabel.setText(username);
                passwordTextField.setText(password);
                fullNameTextField.setText(fullName);
                phoneTextField.setText(phone);
            } else {
                JOptionPane.showMessageDialog(this, "User details not found!");
            }

            resultSet.close();
            preparedStatement.close();
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Failed to fetch user details from the database!");
        }
    }

    private void updateProfile() {
        try {
            String password = passwordTextField.getText();
            String fullName = fullNameTextField.getText();
            String phone = phoneTextField.getText();


   Connection connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);
            PreparedStatement preparedStatement = connection.prepareStatement("UPDATE user_details SET
 password=?, fullname = ?, phone = ? WHERE isloggedin = 'Y'");
            preparedStatement.setString(1, password);
            preparedStatement.setString(2, fullName);
            preparedStatement.setString(3, phone);

            int rowsUpdated = preparedStatement.executeUpdate();
            if (rowsUpdated > 0) {
                JOptionPane.showMessageDialog(this, "Profile updated successfully!");
            } else {
                JOptionPane.showMessageDialog(this, "Failed to update profile!");
            }
```

```
preparedStatement.close();
            connection.close();
            } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Failed to update profile!");
        }
    }

    private static class ButtonHoverListener extends MouseAdapter {
        private final JButton button;
        private final Color originalColor;
        private final Color hoverColor;

        public ButtonHoverListener(JButton button, Color originalColor, Color hoverColor) {
            this.button = button;
            this.originalColor = originalColor;
            this.hoverColor = hoverColor;
        }

        @Override
        public void mouseEntered(MouseEvent e) {
            button.setBackground(hoverColor);
        }

        @Override
        public void mouseExited(MouseEvent e) {
            button.setBackground(originalColor);
        }
    }
}
```

CODE FOR LOGIN

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class LoginSignUpScreen extends JFrame {

    private static final String DB_URL = "jdbc:oracle:thin:@localhost:1521:orcl";
    private static final String DB_USER = "sys as sysdba";
    private static final String DB_PASSWORD = "vasavi";

    private JTextField usernameField;
    private JPasswordField passwordField;

    public LoginSignUpScreen() {
        setTitle("Login and Sign Up");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(800, 600);
        setLocationRelativeTo(null); // Center the frame on the screen

        // Create the content pane
        JPanel contentPane = new JPanel(new BorderLayout());
        contentPane.setBackground(new Color(224, 236, 255)); // Set background color

        // Left section - Logo and title
        JPanel logoPanel = new JPanel();
        logoPanel.setBackground(new Color(224, 236, 255));
        logoPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        ImageIcon logoIcon = new ImageIcon("logo.png");
        Image scaledImage = logoIcon.getImage().getScaledInstance(200, 200, Image.SCALE_SMOOTH);
        ImageIcon scaledLogoIcon = new ImageIcon(scaledImage);
        JLabel logoLabel = new JLabel(scaledLogoIcon);
        logoPanel.add(logoLabel);

        JPanel titlePanel = new JPanel();
        titlePanel.setBackground(new Color(224, 236, 255));

        JLabel titleLabel = new JLabel("Drug Performance Analysis");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 28)); // Increase font size
        titlePanel.add(titleLabel);

       JPanel leftPanel = new JPanel(new BorderLayout());
            leftPanel.setBackground(new Color(224, 236, 255));
            leftPanel.add(logoPanel, BorderLayout.NORTH);
            leftPanel.add(titlePanel, BorderLayout.CENTER);

            contentPane.add(leftPanel, BorderLayout.WEST);
```

```
// Right section - Login and sign up form
    JPanel formPanel = new JPanel();
    formPanel.setBackground(new Color(224, 236, 255));
    formPanel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
    formPanel.setLayout(new BoxLayout(formPanel, BoxLayout.Y_AXIS));

    JLabel usernameLabel = new JLabel("Username:");
    usernameLabel.setFont(new Font("Arial", Font.BOLD, 16)); // Increase font size
    usernameField = new JTextField(20);

    JLabel passwordLabel = new JLabel("Password:");
    passwordLabel.setFont(new Font("Arial", Font.BOLD, 16)); // Increase font size
    passwordField = new JPasswordField(20);

    JButton loginButton = new JButton("Sign In");
    loginButton.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
          String username = usernameField.getText();
          String password = new String(passwordField.getPassword());
          verifyLogin(username, password);
       }
    });

    JButton createUserButton = new JButton("Create User");

    createUserButton.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
          dispose();
          new CreateUserScreen();
       }
    });

    formPanel.add(usernameLabel);
    formPanel.add(Box.createVerticalStrut(5)); // Add separation
    formPanel.add(usernameField);
    formPanel.add(Box.createVerticalStrut(10));
    formPanel.add(passwordLabel);
    formPanel.add(Box.createVerticalStrut(5)); // Add separation
    formPanel.add(passwordField);
    formPanel.add(Box.createVerticalStrut(20)); // Add separation
    formPanel.add(loginButton);
    formPanel.add(Box.createVerticalStrut(10));
    formPanel.add(createUserButton);

    JPanel rightPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
    rightPanel.setBackground(new Color(224, 236, 255));
    rightPanel.setBorder(BorderFactory.createMatteBorder(0, 1, 0, 0, Color.GRAY)); // Add border on the right side
    rightPanel.add(formPanel);

   contentPane.add(rightPanel, BorderLayout.CENTER);

      setContentPane(contentPane);
     }
```

```
 private void verifyLogin(String username, String password) {
      try {
         Connection connection = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
         PreparedStatement statement = connection.prepareStatement("SELECT * FROM USER_DETAILS WHERE
USERNAME = ? AND PASSWORD = ?");
         statement.setString(1, username);
         statement.setString(2, password);

         ResultSet resultSet = statement.executeQuery();
         boolean loginSuccessful = resultSet.next();

         if (loginSuccessful) {
            // Update isLoggedIn column to 'Y' for the logged-in user
            PreparedStatement updateStatement = connection.prepareStatement("UPDATE USER_DETAILS SET
isLoggedIn = 'Y' WHERE USERNAME = ?");
            updateStatement.setString(1, username);
            updateStatement.executeUpdate();
            updateStatement.close();

            // Close the LoginSignUpScreen
            dispose();

            // Open the HomePage screen
            HomePage homePage = new HomePage();
            homePage.createAndShowGUI();
         } else {
            JOptionPane.showMessageDialog(this, "Invalid username or password. Please try again.");
         }

         resultSet.close();
         statement.close();
         connection.close();
      } catch (SQLException e) {
         e.printStackTrace();
      }
   }

   public static void main(String[] args) {
      SwingUtilities.invokeLater(new Runnable() {
         public void run() {
            try {
               // Set look and feel to the system default
               UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
            } catch (Exception e) {
               e.printStackTrace();
            }

            LoginSignUpScreen loginSignUpScreen = new LoginSignUpScreen();
            loginSignUpScreen.setVisible(true);
         }
      });
   }
}
```

CODE FOR INSERT

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class InsertReview {
    private JFrame frame;
    private JPanel mainPanel;
    private JTextField drugNameTextField;
    private JTextField priceTextField;
    private JTextField categoryTextField;
    private JTextArea drugDescriptionTextArea;
    private JTextField docNameTextField;
    private JTextField specializationTextField;
    private JTextField contactDetailsTextField;
    private JTextArea diagnosisTextArea;
    private JTextArea reviewTextArea;
    private JTextField ratingTextField;
    private JCheckBox recommendCheckBox;
    private JTextField effectTextField;
    private JTextField symptomTextField;
    private JButton submitButton;
    private JButton backButton;

    public InsertReview() {
        createGUI();
        createListeners();
    }

    private void createGUI() {
        // Create the main frame
        frame = new JFrame("Insert Review");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


        // Create the main panel
        mainPanel = new JPanel();
        mainPanel.setLayout(new GridLayout(14, 2, 10, 8));
```

```
// Create the form components
        JLabel drugNameLabel = new JLabel("Drug Name:");
        drugNameTextField = new JTextField(20);
        JLabel priceLabel = new JLabel("Price:");
        priceTextField = new JTextField(20);
        JLabel categoryLabel = new JLabel("Category:");
        categoryTextField = new JTextField(20);
        JLabel drugDescriptionLabel = new JLabel("Drug Description:");
        drugDescriptionTextArea = new JTextArea(5, 20);
        drugDescriptionTextArea.setLineWrap(true);
        JLabel docNameLabel = new JLabel("Doctor Name:");
        docNameTextField = new JTextField(20);
        JLabel specializationLabel = new JLabel("Specialization:");
        specializationTextField = new JTextField(20);
        JLabel contactDetailsLabel = new JLabel("Contact Details:");
        contactDetailsTextField = new JTextField(20);
        JLabel diagnosisLabel = new JLabel("Diagnosis:");
        diagnosisTextArea = new JTextArea(5, 20);
        diagnosisTextArea.setLineWrap(true);
        JLabel reviewLabel = new JLabel("Review:");
        reviewTextArea = new JTextArea(5, 20);
        reviewTextArea.setLineWrap(true);
        JLabel ratingLabel = new JLabel("Rating:");
        ratingTextField = new JTextField(20);
        JLabel recommendLabel = new JLabel("Recommend:");
        recommendCheckBox = new JCheckBox();
        JLabel effectLabel = new JLabel("Side Effect:");
        effectTextField = new JTextField( 20);
        JLabel symptomLabel = new JLabel("Symptom:");
        symptomTextField= new JTextField( 20);

        // Create the buttons
        submitButton = new JButton("Submit");
        backButton = new JButton("Back");

        // Add components to the main panel
        mainPanel.add(drugNameLabel);
        mainPanel.add(drugNameTextField);
        mainPanel.add(priceLabel);
        mainPanel.add(priceTextField);
        mainPanel.add(categoryLabel);
        mainPanel.add(categoryTextField);
        mainPanel.add(drugDescriptionLabel);
        mainPanel.add(new JScrollPane(drugDescriptionTextArea));
        mainPanel.add(docNameLabel);
        mainPanel.add(docNameTextField);
        mainPanel.add(specializationLabel);
        mainPanel.add(specializationTextField);
        mainPanel.add(contactDetailsLabel);
```

```
mainPanel.add(contactDetailsTextField);
        mainPanel.add(diagnosisLabel);
        mainPanel.add(new JScrollPane(diagnosisTextArea));
        mainPanel.add(reviewLabel);
        mainPanel.add(new JScrollPane(reviewTextArea));
        mainPanel.add(ratingLabel);
        mainPanel.add(ratingTextField);
        mainPanel.add(recommendLabel);
        mainPanel.add(recommendCheckBox);
        mainPanel.add(effectLabel);
        mainPanel.add(effectTextField);
        mainPanel.add(symptomLabel);
        mainPanel.add(symptomTextField);
        mainPanel.add(submitButton);
        mainPanel.add(backButton);

        mainPanel.setBackground(new Color(224, 236, 255));
        // Add the main panel to the frame
        frame.getContentPane().add(mainPanel);
        frame.pack();
        frame.setVisible(true);
    }

    private void createListeners() {
        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Get input values
                String drugName = drugNameTextField.getText();
                String price = priceTextField.getText();
                String category = categoryTextField.getText();
                String drugDescription = drugDescriptionTextArea.getText();
                String docName = docNameTextField.getText();
                String specialization = specializationTextField.getText();
                String contactDetails = contactDetailsTextField.getText();
                String diagnosis = diagnosisTextArea.getText();
                String review = reviewTextArea.getText();
                String rating = ratingTextField.getText();
                boolean recommend = recommendCheckBox.isSelected();
                String effect=effectTextField.getText();
                String symptom=symptomTextField.getText();

                // Insert data into tables
                try {
                    // Create database connection
                    Connection connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "sys as
sysdba", "vasavi");

                    // Insert into drug table
                    String drugQuery = "INSERT INTO drug (DRUG_NAME, DRUG_DESCRIPTION, PRICE, CATEGORY) VALUES (?,
?, ?)";
                    PreparedStatement drugStatement = connection.prepareStatement(drugQuery);
                    drugStatement.setString(1, drugName);
                    drugStatement.setString(2, drugDescription);
                    drugStatement.setString(3, price);
                    drugStatement.setString(4, category);
                    drugStatement.executeUpdate();
```

```
// Get the auto-generated drug_id
            String lastDrugIdQuery = "SELECT MAX(DRUG_ID) FROM drug";
            PreparedStatement lastDrugIdStatement = connection.prepareStatement(lastDrugIdQuery);
            int drugId = 0;
            ResultSet lastDrugIdResult = lastDrugIdStatement.executeQuery();
            if (lastDrugIdResult.next()) {
               drugId = lastDrugIdResult.getInt(1);
            }

            // Insert into doctor table
            String doctorQuery = "INSERT INTO doctor (DOC_NAME, SPECIALIZATION, CONTACT_DETAILS) VALUES (?, ?,
?)";
            PreparedStatement doctorStatement = connection.prepareStatement(doctorQuery);
            doctorStatement.setString(1, docName);
            doctorStatement.setString(2, specialization);
            doctorStatement.setString(3, contactDetails);
            doctorStatement.executeUpdate();

            // Get the auto-generated doc_id
            String lastDocIdQuery = "SELECT MAX(DOC_ID) FROM doctor";
            PreparedStatement lastDocIdStatement = connection.prepareStatement(lastDocIdQuery);
            int docId = 0;
            ResultSet lastDocIdResult = lastDocIdStatement.executeQuery();
            if (lastDocIdResult.next()) {
               docId = lastDocIdResult.getInt(1);
            }

            // Insert into prescription table
            String prescriptionQuery = "INSERT INTO prescription (USERNAME, DOC_ID, DRUG_ID, DIAGNOSIS) VALUES
(?, ?, ?, ?)";
            PreparedStatement prescriptionStatement = connection.prepareStatement(prescriptionQuery);
            prescriptionStatement.setString(1, "username"); // replace with the actual username
            prescriptionStatement.setInt(2, docId);
            prescriptionStatement.setInt(3, drugId);
            prescriptionStatement.setString(4, diagnosis);
            prescriptionStatement.executeUpdate();

            // Insert into review table
            String reviewQuery = "INSERT INTO review (DRUG_ID, USERNAME, REVIEW_TEXT, OVERALL_RATING,
RECOMMEND) VALUES (?, ?, ?, ?, ?)";
            PreparedStatement reviewStatement = connection.prepareStatement(reviewQuery);
            reviewStatement.setInt(1, drugId);
            reviewStatement.setString(2, "username"); // replace with the actual username
            reviewStatement.setString(3, review);
            reviewStatement.setString(4, rating);
            reviewStatement.setBoolean(5, recommend);
            reviewStatement.executeUpdate();

            // Insert into side_effects table
            String sideEffectsQuery = "INSERT INTO side_effects (DRUG_ID, USERNAME, DATE_ADDED,
SIDE_EFFECTS_DESCRIPTION, SYMPTOM) VALUES (?, ?, NOW(), ?, ?)";
            PreparedStatement sideEffectsStatement = connection.prepareStatement(sideEffectsQuery);
            sideEffectsStatement.setInt(1, drugId);
            sideEffectsStatement.setString(2, "username"); // replace with the actual username
            sideEffectsStatement.setString(3, effect); // replace with the actual side effects description
            sideEffectsStatement.setString(4, symptom); // replace with the actual symptom
            sideEffectsStatement.executeUpdate();
```

```
// Close database connection
                connection.close();

                // Show success message
                JOptionPane.showMessageDialog(frame, "Review inserted successfully!");

            } catch (SQLException ex) {
                ex.printStackTrace();
                // Show error message
                JOptionPane.showMessageDialog(frame, "Error inserting review. Please try again.");
            }
        }
    });

    backButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Go back to previous screen
            frame.dispose();
            // Call the method or navigate to the previous screen
            Review review = new Review();
            review.createAndShowGUI();
        }
    });
  }

  public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new InsertReview();
        }
    });
  }
}
```

CODE FOR DELETE :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class DeleteReview {
    private JFrame frame;
    private JTextField drugNameTextField;
    private JButton searchButton;
    private JTextArea resultTextArea;
    private JButton backButton;
    private JButton exitButton;

    public DeleteReview() {
        frame = new JFrame("Delete Review");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());
        frame.setPreferredSize(new Dimension(800, 600));
        frame.setLocationRelativeTo(null);

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new BorderLayout());

        JPanel inputPanel = new JPanel();
        inputPanel.setLayout(new FlowLayout());
        inputPanel.add(new JLabel("Drug Name:"));
        drugNameTextField = new JTextField(20);
        inputPanel.add(drugNameTextField);

        JPanel buttonPanel = new JPanel();
        searchButton = new JButton("Search");
        backButton = new JButton("Back");
        exitButton = new JButton("Exit");
        buttonPanel.add(searchButton);
        buttonPanel.add(backButton);
        buttonPanel.add(exitButton);

        mainPanel.add(inputPanel, BorderLayout.NORTH);
        mainPanel.add(buttonPanel, BorderLayout.SOUTH);

        resultTextArea = new JTextArea(10, 30);
        resultTextArea.setEditable(false);
        resultTextArea.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(), "Reviews"));
        resultTextArea.setBackground(new Color(224, 236, 255));

        frame.add(mainPanel, BorderLayout.NORTH);
        frame.add(new JScrollPane(resultTextArea), BorderLayout.CENTER);

        createListeners();
```

```
 frame.pack();
      frame.setVisible(true);
   }

   private void createListeners() {
      searchButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            String drugName = drugNameTextField.getText();

            if (drugName.isEmpty()) {
               JOptionPane.showMessageDialog(frame, "Please enter a drug name.");
               return;
            }

            try {
               Connection connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "sys as
sysdba", "vasavi");

               // Check if review exists for the specified drug and user
               String loggedInUserQuery = "SELECT * FROM user_details WHERE isloggedin = 'Y'";
               PreparedStatement loggedInUserStatement = connection.prepareStatement(loggedInUserQuery);
               ResultSet loggedInUserResult = loggedInUserStatement.executeQuery();
               String loggedInUser = "";
               if (loggedInUserResult.next()) {
                  loggedInUser = loggedInUserResult.getString("USERNAME");
               }

               String reviewQuery = "SELECT * FROM review WHERE USERNAME = ?";
               PreparedStatement reviewStatement = connection.prepareStatement(reviewQuery);
               reviewStatement.setString(1, loggedInUser);
               ResultSet reviewResult = reviewStatement.executeQuery();

               StringBuilder reviews = new StringBuilder();
               while (reviewResult.next()) {
                  int reviewID = reviewResult.getInt("REVIEW_ID");
                  String reviewText = reviewResult.getString("REVIEW_TEXT");
                  reviews.append("Review ID: ").append(reviewID).append("\n")
                        .append("Review Text: ").append(reviewText).append("\n\n");
               }

               if (reviews.length() > 0) {
                  resultTextArea.setText(reviews.toString());
               } else {
                  resultTextArea.setText("No reviews found for the specified drug.");
               }

               connection.close();
            } catch (SQLException ex) {
               ex.printStackTrace();
               JOptionPane.showMessageDialog(frame, "Error retrieving reviews. Please try again.");
            }
         }
      });
```

```
backButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Code to navigate back to the previous screen (LoginSignUpScreen.java)
            frame.dispose();
            new LoginSignUpScreen();
        }
    });

    exitButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            frame.dispose();
        }
    });
  }

  public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new DeleteReview();
        }
    });
  }
}
```

CODE FOR CREATE USER :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class CreateUserScreen {
    private JFrame frame;
    private JTextField usernameTextField;
    private JPasswordField passwordField;
    private JPasswordField reenterPasswordField; // New field for re-entering password
    private JTextField fullNameTextField;
    private JTextField phoneTextField;
    private JButton createButton;
    private JButton exitButton;
    private JButton backButton; // New back button

    public CreateUserScreen() {
        frame = new JFrame("Create User");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());
        frame.setPreferredSize(new Dimension(800, 600));
        frame.setLocationRelativeTo(null);

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new GridBagLayout());
        mainPanel.setBackground(new Color(224, 236, 255)); // Set light blue background color

        GridBagConstraints gbc = new GridBagConstraints();
        gbc.anchor = GridBagConstraints.WEST;
        gbc.insets = new Insets(10, 10, 10, 10); // Add padding

        mainPanel.add(createTitleLabel("Register", SwingConstants.CENTER), gbc); // Add title label

        gbc.gridy = 1;
        mainPanel.add(createLabel("Username:"), gbc);
        gbc.gridx = 1;
        usernameTextField = new JTextField(20);
        mainPanel.add(usernameTextField, gbc);

        gbc.gridy = 2;
        gbc.gridx = 0;
        mainPanel.add(createLabel("Password:"), gbc);
        gbc.gridx = 1;
        passwordField = new JPasswordField(20);
        mainPanel.add(passwordField, gbc);

        gbc.gridy = 3;
        gbc.gridx = 0;
        mainPanel.add(createLabel("Re-enter Password:"), gbc);
        gbc.gridx = 1;
        reenterPasswordField = new JPasswordField(20);
        mainPanel.add(reenterPasswordField, gbc);
```

```
gbc.gridy = 4;
      gbc.gridx = 0;
      mainPanel.add(createLabel("Full Name:"), gbc);
      gbc.gridx = 1;
      fullNameTextField = new JTextField(20);
      mainPanel.add(fullNameTextField, gbc);

      gbc.gridy = 5;
      gbc.gridx = 0;
      mainPanel.add(createLabel("Phone:"), gbc);
      gbc.gridx = 1;
      phoneTextField = new JTextField(20);
      mainPanel.add(phoneTextField, gbc);

      createButton = new JButton("Create");
      exitButton = new JButton("Exit");
      backButton = new JButton("Back"); // Create back button

      JPanel buttonPanel = new JPanel();
      buttonPanel.setBackground(new Color(224, 236, 255)); // Set light blue background color
      buttonPanel.add(createButton);
      buttonPanel.add(exitButton);
      buttonPanel.add(backButton); // Add back button to button panel

      frame.add(mainPanel, BorderLayout.CENTER);
      frame.add(buttonPanel, BorderLayout.SOUTH);

      createButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            String username = usernameTextField.getText();
            String password = new String(passwordField.getPassword());
            String reenteredPassword = new String(reenterPasswordField.getPassword());
            String fullName = fullNameTextField.getText();
            String phone = phoneTextField.getText();

            if (username.isEmpty() || password.isEmpty() || fullName.isEmpty() || phone.isEmpty()) {
               JOptionPane.showMessageDialog(frame, "Please enter all fields.");
               return;
            }

            if (!password.equals(reenteredPassword)) {
               JOptionPane.showMessageDialog(frame, "Passwords do not match.");
               return;
            }

            try {
               Connection connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "sys as
sysdba", "vasavi");

               String insertQuery = "INSERT INTO user_details (USERNAME, PASSWORD, FULLNAME, PHONE) VALUES (?, ?,
?, ?)";
               PreparedStatement insertStatement = connection.prepareStatement(insertQuery);
               insertStatement.setString(1, username);
               insertStatement.setString(2, password);
               insertStatement.setString(3, fullName);
               insertStatement.setString(4, phone);
               int rowsInserted = insertStatement.executeUpdate();
```

```
if (rowsInserted > 0) {
                JOptionPane.showMessageDialog(frame, "User created successfully.");

                // Navigate to LoginSignUpScreen
                frame.dispose();
                new LoginSignUpScreen();
            } else {
                JOptionPane.showMessageDialog(frame, "User creation failed. Please try again.");
            }

            connection.close();
        } catch (SQLException ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(frame, "Error creating user. Please try again.");
        }
      }
    });

    exitButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        frame.dispose();
      }
    });

    backButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        frame.dispose();
        LoginSignUpScreen loginSignUpScreen = new LoginSignUpScreen();
        loginSignUpScreen.setVisible(true);
      }
    });

    frame.pack();
    frame.setVisible(true);
  }

  private JLabel createLabel(String text) {
    JLabel label = new JLabel(text);
    label.setFont(new Font("Arial", Font.BOLD, 16)); // Increase font size
    return label;
  }

  private JLabel createTitleLabel(String text, int alignment) {
    JLabel label = new JLabel(text);
    label.setFont(new Font("Arial", Font.BOLD, 24)); // Increase font size
    label.setHorizontalAlignment(alignment);
    return label;
  }

  public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
      @Override
      public void run() {
        try {
          // Set look and feel to the system default
          UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
          e.printStackTrace();
        }

        new CreateUserScreen();
      }
    });
  }
}
```

CODE FOR REVIEW :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class Review extends JFrame {

    private JButton homeButton;
    private JButton reviewButton;
    private JButton profileButton;
    private JButton logoutButton;
    private JButton exitButton;
    private JButton searchButton;
    private JButton insertButton;
    private JButton deleteButton;

    public static void main(String[] args) {
        SwingUtilities.invokeLater(Review::new);
    }

    public Review() {
        createAndShowGUI();
    }

    public void createAndShowGUI() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Review");
        setSize(800, 600);
        setLocationRelativeTo(null);

        // Create the layout components
        JPanel topPanel = new JPanel();
        JPanel leftPanel = new JPanel();
        JPanel rightPanel = new JPanel();
        JPanel buttonPanel = new JPanel();
        JPanel imagePanel = new JPanel();
        JPanel exitPanel = new JPanel();

        // Create the buttons
        homeButton = new JButton("Home");
        reviewButton = new JButton("Review");
        profileButton = new JButton("Profile");
        logoutButton = new JButton("Logout");
        exitButton = new JButton("Exit");
        searchButton = new JButton("Search");
        insertButton = new JButton("Insert");
        deleteButton = new JButton("Delete");

        // Set the button colors
        Color buttonColor = new Color(17, 30, 108);
        Color buttonHoverColor = new Color(102, 204, 255);
        setButtonStyles(homeButton, buttonColor, buttonHoverColor);
        setButtonStyles(reviewButton, buttonColor, buttonHoverColor);
        setButtonStyles(profileButton, buttonColor, buttonHoverColor);
        setButtonStyles(logoutButton, buttonColor, buttonHoverColor);
        setButtonStyles(exitButton, buttonColor, buttonHoverColor);
        setButtonStyles(searchButton, buttonColor, buttonHoverColor);
        setButtonStyles(insertButton, buttonColor, buttonHoverColor);
        setButtonStyles(deleteButton, buttonColor, buttonHoverColor);
```

```
// Set the layouts
    topPanel.setLayout(new BoxLayout(topPanel, BoxLayout.X_AXIS));
    leftPanel.setLayout(new GridLayout(0, 1, 0, 10)); // Updated GridLayout with vertical alignment and spacing
    rightPanel.setLayout(new BorderLayout());
    buttonPanel.setLayout(new GridLayout(3, 1, 0, 10)); // Updated GridLayout with vertical alignment and spacing
    imagePanel.setLayout(new BorderLayout());
    exitPanel.setLayout(new FlowLayout(FlowLayout.RIGHT));

    // Set background color for panels
    topPanel.setBackground(new Color(224, 236, 255));
    leftPanel.setBackground(new Color(224,236,255));
    rightPanel.setBackground(new Color(224,236,255));
    buttonPanel.setBackground(new Color(224,236,255));
    imagePanel.setBackground(new Color(224,236,255));
    exitPanel.setBackground(new Color(224, 236, 255));

    // Add components to their respective containers
    topPanel.add(homeButton);
    topPanel.add(reviewButton);
    topPanel.add(profileButton);
    topPanel.add(logoutButton);
    topPanel.add(Box.createHorizontalGlue()); // Add horizontal glue to right-align the exit button
    exitPanel.add(exitButton);

    JLabel titleLabel = new JLabel("Explore the Effects of Drugs");
    titleLabel.setFont(new Font("Arial", Font.BOLD, 18)); // Adjust font style and size if needed
    leftPanel.add(titleLabel);

    leftPanel.add(Box.createVerticalGlue()); // Add vertical glue to align buttons at the top
    buttonPanel.add(searchButton);
    buttonPanel.add(insertButton);
    buttonPanel.add(deleteButton);
    leftPanel.add(buttonPanel);
    leftPanel.add(Box.createVerticalGlue()); // Add vertical glue to align buttons at the bottom

    ImageIcon image = new ImageIcon("image2.jpg"); // Replace with the actual path to your image
    JLabel imageLabel = new JLabel(image);
    image.setImage(image.getImage().getScaledInstance(390, -1, Image.SCALE_SMOOTH));
    imagePanel.add(imageLabel);
    imageLabel.setIcon(image);

    // Add panels to the frame
    getContentPane().setLayout(new BorderLayout());
    getContentPane().add(topPanel, BorderLayout.NORTH);
    getContentPane().add(leftPanel, BorderLayout.WEST);
    getContentPane().add(rightPanel, BorderLayout.CENTER);
    getContentPane().add(exitPanel, BorderLayout.SOUTH);
    rightPanel.add(imagePanel, BorderLayout.CENTER);


    exitButton.addActionListener(e -> dispose());

    logoutButton.addActionListener(e -> {
       dispose();
       LoginSignUpScreen loginSignUpScreen = new LoginSignUpScreen();
       loginSignUpScreen.setVisible(true);
    });

    // Add action listeners to the buttons
    homeButton.addActionListener(e -> {
       dispose();
       HomePage homepage = new HomePage();
       homepage.createAndShowGUI();
    });
```

```
reviewButton.addActionListener(e -> {
        dispose();
        Review review = new Review();
        review.createAndShowGUI();
    });

    profileButton.addActionListener(e -> {
        dispose();
        Profile profile = new Profile();
        profile.createAndShowGUI();
    });

    searchButton.addActionListener(e -> {
        dispose();
        SearchReview search = new SearchReview();
        search.createAndShowGUI();
    });

    insertButton.addActionListener(e -> {
        dispose();
        new InsertReview();
    });

    deleteButton.addActionListener(e -> {
        dispose();
        new DeleteReview();
    });

    setVisible(true);
  }

  private void setButtonStyles(JButton button, Color buttonColor, Color buttonHoverColor) {
    button.setBackground(buttonColor);
    button.setForeground(Color.WHITE);
    button.setFocusPainted(false);
    button.setFont(new Font("Arial", Font.BOLD, 14));
    button.addMouseListener(new ButtonHoverListener(button, buttonColor, buttonHoverColor));
  }

  private static class ButtonHoverListener extends MouseAdapter {
    private final JButton button;
    private final Color originalColor;
    private final Color hoverColor;

    public ButtonHoverListener(JButton button, Color originalColor, Color hoverColor) {
        this.button = button;
        this.originalColor = originalColor;
        this.hoverColor = hoverColor;
    }

    @Override
    public void mouseEntered(MouseEvent e) {
        button.setBackground(hoverColor);
    }

    @Override
    public void mouseExited(MouseEvent e) {
        button.setBackground(originalColor);
    }
  }
}
```

CODE FOR PROFILE :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.*;

public class Profile extends JFrame {

    private static final String JDBC_URL = "jdbc:oracle:thin:@localhost:1521:orcl"; // Update with your database connection URL
    private static final String USERNAME = "sys as sysdba"; // Replace with your database username
    private static final String PASSWORD = "vasavi"; // Replace with your database password

    private JLabel usernameValueLabel;
    private JLabel fullNameValueLabel;
    private JLabel phoneValueLabel;
    private JLabel lastLoginValueLabel;

    public static void main(String[] args) {
        SwingUtilities.invokeLater(Profile::new);
    }

    public Profile() {
        createAndShowGUI();
    }

    public void createAndShowGUI() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Profile");
        setSize(800, 600);
        setLocationRelativeTo(null);

        // Create the layout components
        JPanel topPanel = new JPanel();
        JPanel infoPanel = new JPanel();
        JPanel titlePanel = new JPanel();

        // Create the buttons
        JButton homeButton = new JButton("Home");
        JButton reviewButton = new JButton("Review");
        JButton profileButton = new JButton("Profile");
        JButton logoutButton = new JButton("Logout");
        JButton exitButton = new JButton("Exit");
        JButton editProfileButton = new JButton("Edit Profile");

        // Set the button colors
        Color buttonColor = new Color(17, 30, 108);
    Color buttonHoverColor = new Color(175, 238, 238);
        homeButton.setBackground(buttonColor);
        homeButton.setForeground(Color.WHITE);
        homeButton.setFocusPainted(false);
        homeButton.addMouseListener(new ButtonHoverListener(homeButton, buttonColor, buttonHoverColor));
        reviewButton.setBackground(buttonColor);
        reviewButton.setForeground(Color.WHITE);
        reviewButton.setFocusPainted(false);
        reviewButton.addMouseListener(new ButtonHoverListener(reviewButton, buttonColor, buttonHoverColor));
        profileButton.setBackground(buttonColor);
        profileButton.setForeground(Color.WHITE);
        profileButton.setFocusPainted(false);
```

```
profileButton.addMouseListener(new ButtonHoverListener(profileButton, buttonColor, buttonHoverColor));
    logoutButton.setBackground(buttonColor);
    logoutButton.setForeground(Color.WHITE);
    logoutButton.setFocusPainted(false);
    logoutButton.addMouseListener(new ButtonHoverListener(logoutButton, buttonColor, buttonHoverColor));
    exitButton.setBackground(buttonColor);
    exitButton.setForeground(Color.WHITE);
    exitButton.setFocusPainted(false);
    exitButton.addMouseListener(new ButtonHoverListener(exitButton, buttonColor, buttonHoverColor));
    editProfileButton.setBackground(buttonColor);
    editProfileButton.setForeground(Color.WHITE);
    editProfileButton.setFocusPainted(false);
    editProfileButton.setPreferredSize(new Dimension(100, 30)); // Set the preferred size

    JLabel profileLabel = new JLabel("<html><body><font size='6'>Profile</font></body></html>");

    // Create labels for user information
    JLabel usernameLabel = new JLabel("Username:");
    JLabel fullNameLabel = new JLabel("Full Name:");
    JLabel phoneLabel = new JLabel("Phone:");
    JLabel lastLoginLabel = new JLabel("Last Login:");

    // Set font size for labels
    Font labelFont = new Font(Font.SANS_SERIF, Font.PLAIN, 16);
    usernameLabel.setFont(labelFont);
    fullNameLabel.setFont(labelFont);
    phoneLabel.setFont(labelFont);
    lastLoginLabel.setFont(labelFont);

    // Set font size for values
    Font valueFont = new Font(Font.SANS_SERIF, Font.BOLD, 18);

    // Set the layouts
    topPanel.setLayout(new BoxLayout(topPanel, BoxLayout.X_AXIS));
    infoPanel.setLayout(new GridLayout(6, 2));
    titlePanel.setLayout(new BorderLayout());

    // Set background color for panels
    topPanel.setBackground(new Color(224, 236, 255));
    infoPanel.setBackground(Color.WHITE);
    titlePanel.setBackground(new Color(224, 236, 255));

    // Add components to their respective containers
    topPanel.add(homeButton);
    topPanel.add(reviewButton);
    topPanel.add(profileButton);
    topPanel.add(logoutButton);
    topPanel.add(Box.createHorizontalGlue()); // Add horizontal glue to right-align the exit button
    topPanel.add(editProfileButton);

    titlePanel.add(profileLabel, BorderLayout.WEST);

    infoPanel.add(usernameLabel);
    usernameValueLabel = new JLabel();
    usernameValueLabel.setFont(valueFont);
    infoPanel.add(usernameValueLabel);
    infoPanel.add(fullNameLabel);
    fullNameValueLabel = new JLabel();
    fullNameValueLabel.setFont(valueFont);
    infoPanel.add(fullNameValueLabel);
    infoPanel.add(phoneLabel);
    phoneValueLabel = new JLabel();
    phoneValueLabel.setFont(valueFont);
    infoPanel.add(phoneValueLabel);
    infoPanel.add(lastLoginLabel);
    lastLoginValueLabel = new JLabel();
     lastLoginValueLabel.setFont(valueFont);
     infoPanel.add(lastLoginValueLabel);
     infoPanel.add(new JLabel()); // Empty label for spacing
     infoPanel.setBackground(new Color(224, 236, 255));
```

```
// Add edit profile button to a separate panel for layout control
      JPanel exitPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
      exitPanel.add(exitButton);

      // Add panels to the frame
      getContentPane().setLayout(new BorderLayout());
      getContentPane().add(topPanel, BorderLayout.NORTH);
      getContentPane().add(infoPanel, BorderLayout.CENTER);
      getContentPane().add(exitPanel, BorderLayout.SOUTH);

      exitButton.addActionListener(e -> dispose());

      homeButton.addActionListener(e -> {
         dispose();
         HomePage homepage = new HomePage();
         homepage.createAndShowGUI();
      });

      reviewButton.addActionListener(e -> {
         dispose();
         // Create and show the Review screen
         new Review();
      });

      logoutButton.addActionListener(e -> {
         dispose();
         LoginSignUpScreen loginSignUpScreen = new LoginSignUpScreen();
         loginSignUpScreen.setVisible(true);
      });

      profileButton.addActionListener(e -> {
         dispose();
         // Create and show the Profile screen
         Profile profilePage = new Profile();
         profilePage.createAndShowGUI();
      });

      editProfileButton.addActionListener(e -> {
         dispose();
         new EditProfile();
      });

      fetchUserDetails(); // Fetch user details from the database

      setVisible(true);
   }

   private void fetchUserDetails() {
      try {
         Connection connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);
         PreparedStatement preparedStatement = connection.prepareStatement("SELECT * FROM user_details WHERE isloggedin
'Y'");
         ResultSet resultSet = preparedStatement.executeQuery();

         if (resultSet.next()) {
            String username = resultSet.getString("USERNAME");
            String fullName = resultSet.getString("FULLNAME");
            String phone = resultSet.getString("PHONE");
            Date lastLogin = resultSet.getDate("LAST_LOGIN");
```

```
 usernameValueLabel.setText(username);
            fullNameValueLabel.setText(fullName);
            phoneValueLabel.setText(phone);
            lastLoginValueLabel.setText(lastLogin.toString());
          } else {
            JOptionPane.showMessageDialog(this, "User details not found!");
          }

          resultSet.close();
          preparedStatement.close();
          connection.close();
        } catch (SQLException e) {
          e.printStackTrace();
          JOptionPane.showMessageDialog(this, "Failed to fetch user details from the database!");
        }
    }

    private static class ButtonHoverListener extends MouseAdapter {
        private final JButton button;
        private final Color originalColor;
        private final Color hoverColor;

        public ButtonHoverListener(JButton button, Color originalColor, Color hoverColor) {
            this.button = button;
            this.originalColor = originalColor;
            this.hoverColor = hoverColor;
        }

        @Override
        public void mouseEntered(MouseEvent e) {
            button.setBackground(hoverColor);
        }

        @Override
        public void mouseExited(MouseEvent e) {
            button.setBackground(originalColor);
        }
    }
}
```

CODE FOR SEARCH :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.*;

public class SearchReview extends JFrame {

    private JButton homeButton;
    private JButton reviewButton;
    private JButton profileButton;
    private JButton logoutButton;
    private JButton exitButton;
    private JButton submitButton;
    private JButton backButton;
    private JTextField drugNameTextField;
    private JTextArea resultTextArea;
    private JScrollPane resultScrollPane;

    public static void main(String[] args) {
        SwingUtilities.invokeLater(SearchReview::new);
    }

    public SearchReview() {
        createAndShowGUI();
    }

    public void createAndShowGUI() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Search Review");
        setSize(800, 600);
        setLocationRelativeTo(null);

        // Create the layout components
        JPanel topPanel = new JPanel();
        JPanel leftPanel = new JPanel();
        JPanel rightPanel = new JPanel();
        JPanel buttonPanel = new JPanel();
        JPanel resultPanel = new JPanel();
        JPanel exitPanel = new JPanel();

        // Create the buttons
        homeButton = new JButton("Home");
        reviewButton = new JButton("Review");
        profileButton = new JButton("Profile");
        logoutButton = new JButton("Logout");
        exitButton = new JButton("Exit");
        submitButton = new JButton("Submit");
        backButton = new JButton("Back");

        // Set the button colors
        Color buttonColor = new Color(17, 30, 108);
        Color buttonHoverColor = new Color(102, 204, 255);
        setButtonStyles(homeButton, buttonColor, buttonHoverColor);
        setButtonStyles(reviewButton, buttonColor, buttonHoverColor);
        setButtonStyles(profileButton, buttonColor, buttonHoverColor);
        setButtonStyles(logoutButton, buttonColor, buttonHoverColor);
        setButtonStyles(exitButton, buttonColor, buttonHoverColor);
        setButtonStyles(submitButton, buttonColor, buttonHoverColor);
        setButtonStyles(backButton, buttonColor, buttonHoverColor);
```

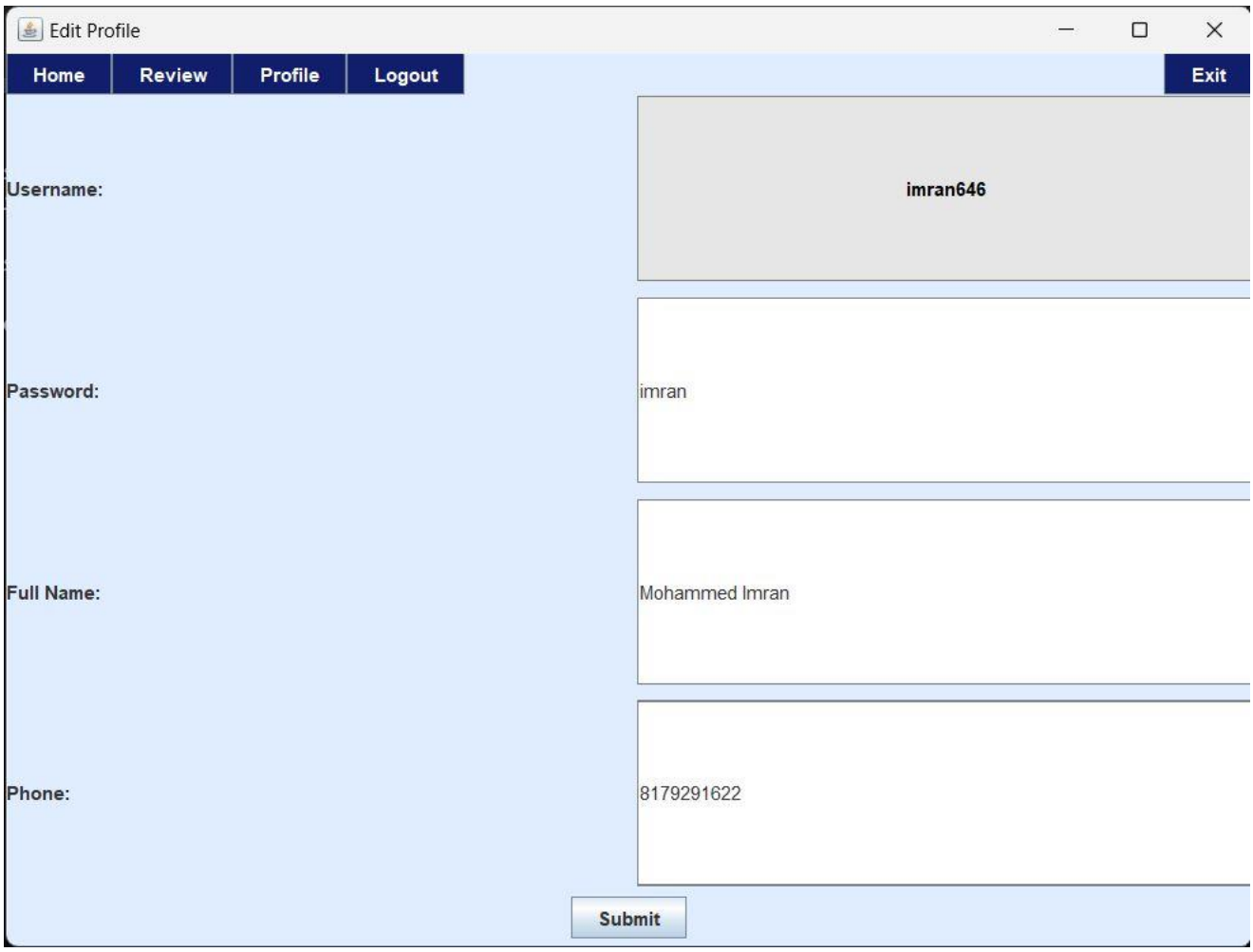
```
submitButton.setPreferredSize(new Dimension(1, submitButton.getPreferredSize().height));
        // Set the layouts
        topPanel.setLayout(new BoxLayout(topPanel, BoxLayout.X_AXIS));
    leftPanel.setLayout(new GridLayout(0, 1, 0, 10));
    rightPanel.setLayout(new BorderLayout());
    buttonPanel.setLayout(new GridLayout(4, 1, 0, 10));
    resultPanel.setLayout(new BorderLayout());
    exitPanel.setLayout(new FlowLayout(FlowLayout.RIGHT));

    // Set background color for panels
    topPanel.setBackground(Color.WHITE);
    leftPanel.setBackground(Color.WHITE);
    rightPanel.setBackground(Color.WHITE);
    buttonPanel.setBackground(Color.WHITE);
    resultPanel.setBackground(Color.WHITE);
    exitPanel.setBackground(Color.WHITE);

    // Add components to their respective containers
    topPanel.add(homeButton);
    topPanel.add(reviewButton);
    topPanel.add(profileButton);
    topPanel.add(logoutButton);
    topPanel.add(Box.createHorizontalGlue()); // Add horizontal glue to right-align the exit button
    topPanel.add(exitButton); // Add the exit button to the top panel

    JLabel titleLabel = new JLabel("Search Drug Review");
    titleLabel.setFont(new Font("Arial", Font.BOLD, 18));
   // leftPanel.add(titleLabel);
    leftPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

    drugNameTextField = new JTextField();
    buttonPanel.add(titleLabel);
    buttonPanel.add(drugNameTextField);
    buttonPanel.add(submitButton);
    buttonPanel.add(backButton);
    buttonPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

    resultTextArea = new JTextArea();
    resultTextArea.setEditable(false);
    resultScrollPane = new JScrollPane(resultTextArea);
    resultScrollPane.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
    resultPanel.add(resultScrollPane, BorderLayout.CENTER);
    resultPanel.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(), "Search Result"));

    // Add panels to the frame
    getContentPane().setLayout(new BorderLayout());
    getContentPane().add(topPanel, BorderLayout.NORTH);
    getContentPane().add(leftPanel, BorderLayout.WEST);
    getContentPane().add(rightPanel, BorderLayout.CENTER);
    getContentPane().add(exitPanel, BorderLayout.SOUTH);
    rightPanel.add(buttonPanel, BorderLayout.NORTH);
    rightPanel.add(resultPanel, BorderLayout.CENTER);


        // Register event listeners
        submitButton.addActionListener(e -> {
            String drugName = drugNameTextField.getText();

            // Clear previous results
            resultTextArea.setText("");
```

```
// Database connection details
        String url = "jdbc:oracle:thin:@localhost:1521:orcl";
        String username = "sys as sysdba";
        String password = "vasavi";

        try {
           // Establish a database connection
           Connection connection = DriverManager.getConnection(url, username, password);

           // Query to retrieve drug information
           String drugQuery = "SELECT * FROM drug WHERE DRUG_NAME = ?";
           PreparedStatement drugStatement = connection.prepareStatement(drugQuery);
           drugStatement.setString(1, drugName);
           ResultSet drugResult = drugStatement.executeQuery();

           // Display drug information

           if(!drugResult.next()) {
              resultTextArea.append("No drug information found for the given name.");
           }else{
           displayDrugInfo(drugResult);
           // Query to retrieve doctor information
           String doctorQuery = "SELECT * FROM doctor WHERE DOC_ID IN (SELECT DOC_ID FROM prescription WHERE DRUG_ID IN (SELECT
DRUG_ID FROM drug WHERE DRUG_NAME = ?))";
           PreparedStatement doctorStatement = connection.prepareStatement(doctorQuery);
           doctorStatement.setString(1, drugName);
           ResultSet doctorResult = doctorStatement.executeQuery();

           // Display doctor information
           displayDoctorInfo(doctorResult);

           // Query to retrieve prescription information
           String prescriptionQuery = "SELECT * FROM prescription WHERE DRUG_ID IN (SELECT DRUG_ID FROM drug WHERE DRUG_NAME =
?)";
           PreparedStatement prescriptionStatement = connection.prepareStatement(prescriptionQuery);
           prescriptionStatement.setString(1, drugName);
           ResultSet prescriptionResult = prescriptionStatement.executeQuery();

           // Display prescription information
           displayPrescriptionInfo(prescriptionResult);

           // Query to retrieve review information
           String reviewQuery = "SELECT * FROM review WHERE DRUG_ID IN (SELECT DRUG_ID FROM drug WHERE DRUG_NAME = ?)";
           PreparedStatement reviewStatement = connection.prepareStatement(reviewQuery);
           reviewStatement.setString(1, drugName);
           ResultSet reviewResult = reviewStatement.executeQuery();

           // Display review information
           displayReviewInfo(reviewResult);

           // Query to retrieve side effects information
           String sideEffectsQuery = "SELECT * FROM side_effects WHERE DRUG_ID IN (SELECT DRUG_ID FROM drug WHERE DRUG_NAME = ?)";
           PreparedStatement sideEffectsStatement = connection.prepareStatement(sideEffectsQuery);
           sideEffectsStatement.setString(1, drugName);
           ResultSet sideEffectsResult = sideEffectsStatement.executeQuery();

           // Display side effects information
           displaySideEffectsInfo(sideEffectsResult);

           // Close the database connection
           connection.close();
           }
        } catch (SQLException ex) {
           ex.printStackTrace();
            resultTextArea.append("Error occurred while retrieving data from the database.");
         }
      });
```

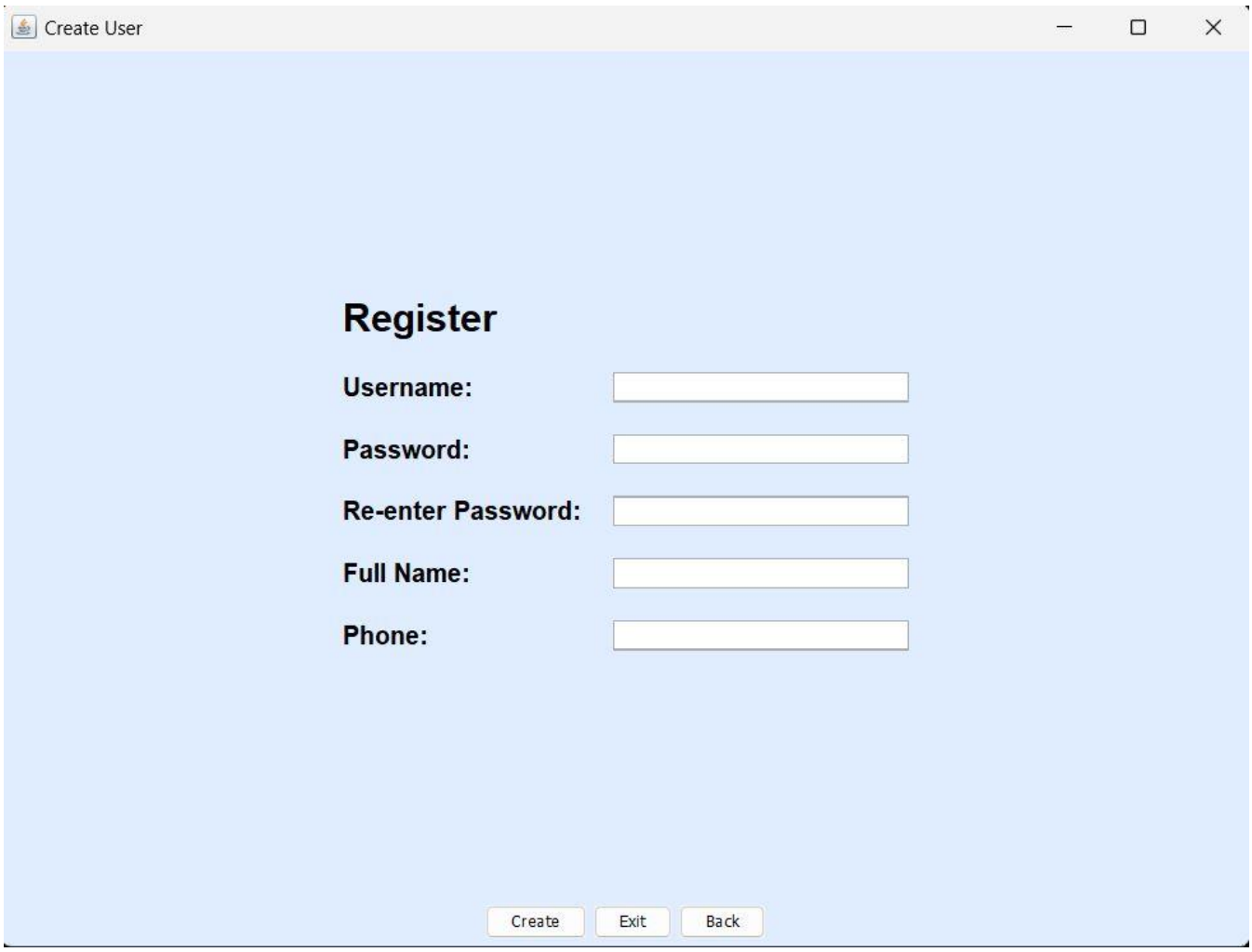
```
backButton.addActionListener(e -> {
        dispose();
        Review review = new Review();
        review.createAndShowGUI();
    });

    exitButton.addActionListener(e -> dispose());

    logoutButton.addActionListener(e -> {
        dispose();
        LoginSignUpScreen loginSignUpScreen = new LoginSignUpScreen();
        loginSignUpScreen.setVisible(true);
    });

    homeButton.addActionListener(e -> {
        dispose();
        HomePage homepage = new HomePage();
        homepage.createAndShowGUI();
    });

    reviewButton.addActionListener(e -> {
        dispose();
        Review review = new Review();
        review.createAndShowGUI();
    });

    profileButton.addActionListener(e -> {
        dispose();
        Profile profile = new Profile();
        profile.createAndShowGUI();
    });

    setVisible(true);
  }


  private void displayDrugInfo(ResultSet drugResult) throws SQLException {
    resultTextArea.append("Drug Information:\n");
    resultTextArea.append("Drug ID: " + drugResult.getInt("DRUG_ID") + "\n");
    resultTextArea.append("Drug Name: " + drugResult.getString("DRUG_NAME") + "\n");
    resultTextArea.append("Description: " + drugResult.getString("DRUG_DESCRIPTION") + "\n");
    resultTextArea.append("Price: " + drugResult.getDouble("PRICE") + "\n");
    resultTextArea.append("Category: " + drugResult.getString("CATEGORY") + "\n");
    resultTextArea.append("Date Added: " + drugResult.getDate("DATE_ADDED") + "\n");
    resultTextArea.append("\n");
  }


  private void displayDoctorInfo(ResultSet doctorResult) throws SQLException {
    resultTextArea.append("Doctor Information:\n");
    while (doctorResult.next()) {
        resultTextArea.append("Doctor ID: " + doctorResult.getInt("DOC_ID") + "\n");
        resultTextArea.append("Doctor Name: " + doctorResult.getString("DOC_NAME") + "\n");
        resultTextArea.append("Specialization: " + doctorResult.getString("SPECIALIZATION") + "\n");
        resultTextArea.append("Contact Details: " + doctorResult.getString("CONTACT_DETAILS") + "\n");
        resultTextArea.append("\n");
    }
  }
```

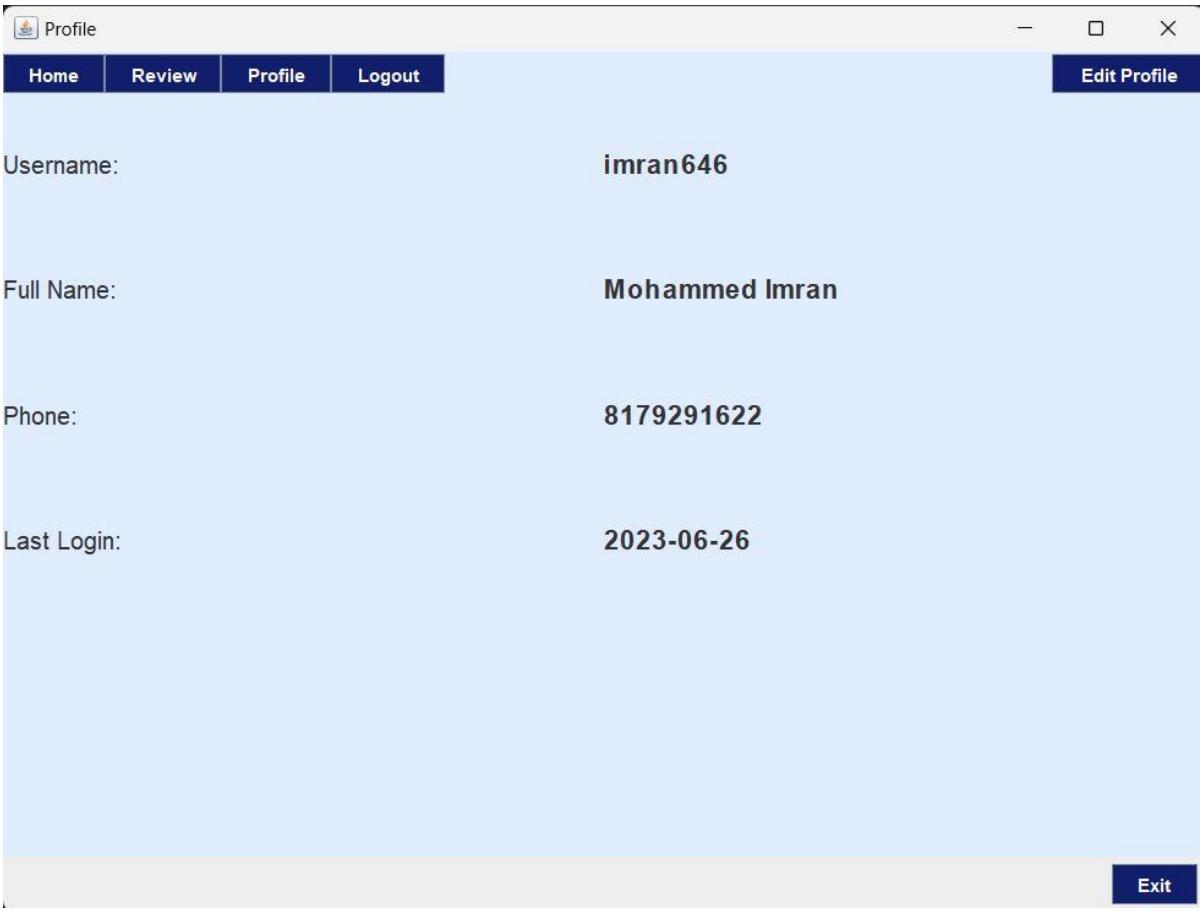
```
private void displayPrescriptionInfo(ResultSet prescriptionResult) throws SQLException {
      resultTextArea.append("Prescription Information:\n");
      while (prescriptionResult.next()) {
         resultTextArea.append("Prescription ID: " + prescriptionResult.getInt("PRES_ID") + "\n");
         resultTextArea.append("Username: " + prescriptionResult.getString("USERNAME") + "\n");
         resultTextArea.append("Doctor ID: " + prescriptionResult.getInt("DOC_ID") + "\n");
         resultTextArea.append("Drug ID: " + prescriptionResult.getInt("DRUG_ID") + "\n");
         resultTextArea.append("Diagnosis: " + prescriptionResult.getString("DIAGNOSIS") + "\n");
         resultTextArea.append("Date Added: " + prescriptionResult.getDate("DATE_ADDED") + "\n");
         resultTextArea.append("\n");
      }
   }

   private void displayReviewInfo(ResultSet reviewResult) throws SQLException {
      resultTextArea.append("Review Information:\n");
      while (reviewResult.next()) {
         resultTextArea.append("Review ID: " + reviewResult.getInt("REVIEW_ID") + "\n");
         resultTextArea.append("Drug ID: " + reviewResult.getInt("DRUG_ID") + "\n");
         resultTextArea.append("Username: " + reviewResult.getString("USERNAME") + "\n");
         resultTextArea.append("Review Text: " + reviewResult.getString("REVIEW_TEXT") + "\n");
         resultTextArea.append("Overall Rating: " + reviewResult.getInt("OVERALL_RATING") + "\n");
         resultTextArea.append("Recommend: " + reviewResult.getString("RECOMMEND") + "\n");
         resultTextArea.append("Review Date: " + reviewResult.getDate("REVIEW_DATE") + "\n");
         resultTextArea.append("\n");
      }
   }

   private void displaySideEffectsInfo(ResultSet sideEffectsResult) throws SQLException {
      resultTextArea.append("Side Effects Information:\n");
      while (sideEffectsResult.next()) {
         resultTextArea.append("Side Effect ID: " + sideEffectsResult.getInt("SIDE_EFFECT_ID") + "\n");
         resultTextArea.append("Drug ID: " + sideEffectsResult.getInt("DRUG_ID") + "\n");
         resultTextArea.append("Username: " + sideEffectsResult.getString("USERNAME") + "\n");
         resultTextArea.append("Date Added: " + sideEffectsResult.getDate("DATE_ADDED") + "\n");
         resultTextArea.append("Side Effects Description: " + sideEffectsResult.getString("SIDE_EFFECTS_DESCRIPTION") + "\n");
         resultTextArea.append("Symptom: " + sideEffectsResult.getString("SYMPTOM") + "\n");
         resultTextArea.append("\n");
      }
   }


   private void setButtonStyles(JButton button, Color bgColor, Color hoverColor) {
      button.setBackground(bgColor);
      button.setForeground(Color.WHITE);
      button.setFocusPainted(false);
      button.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20));
      button.addMouseListener(new MouseAdapter() {
         public void mouseEntered(MouseEvent e) {
            button.setBackground(hoverColor);
         }

         public void mouseExited(MouseEvent e) {
            button.setBackground(bgColor);
         }
      });
   }
}
```

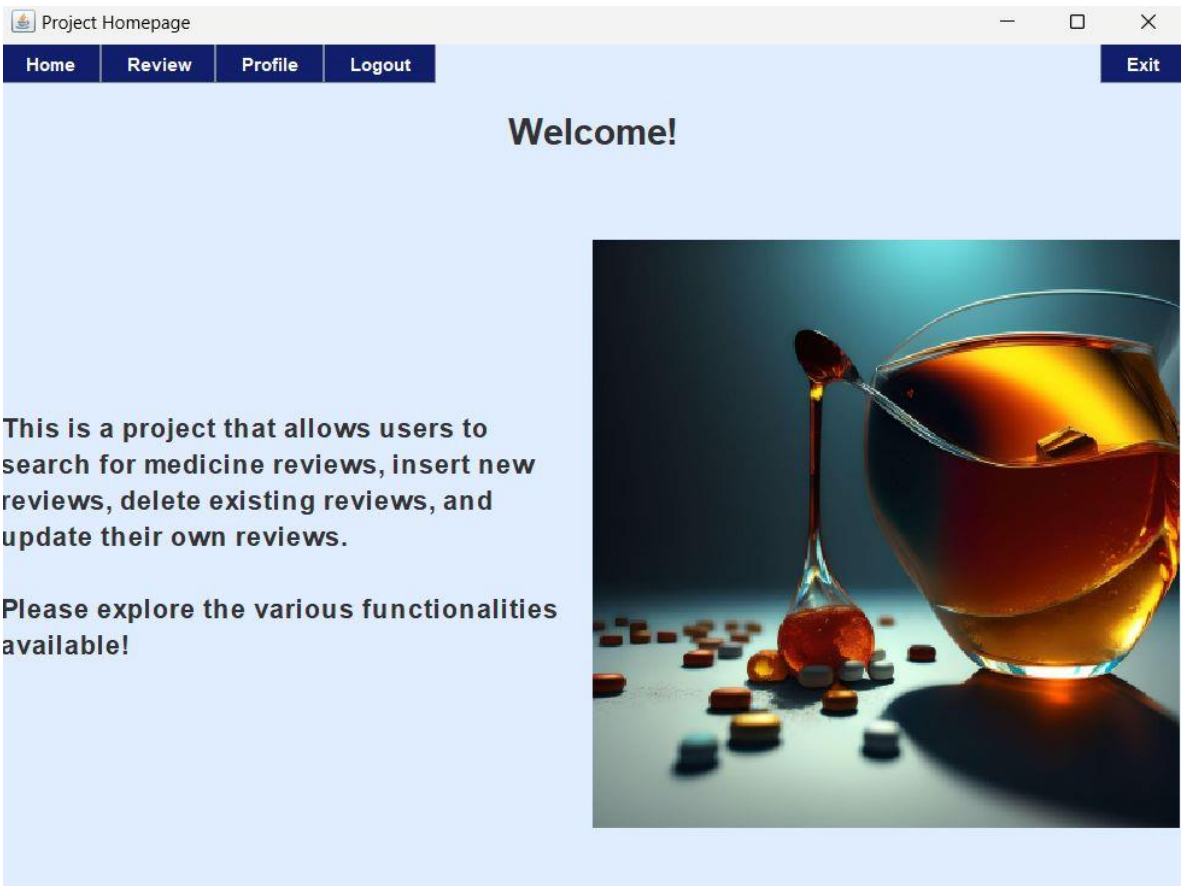
TESTING :

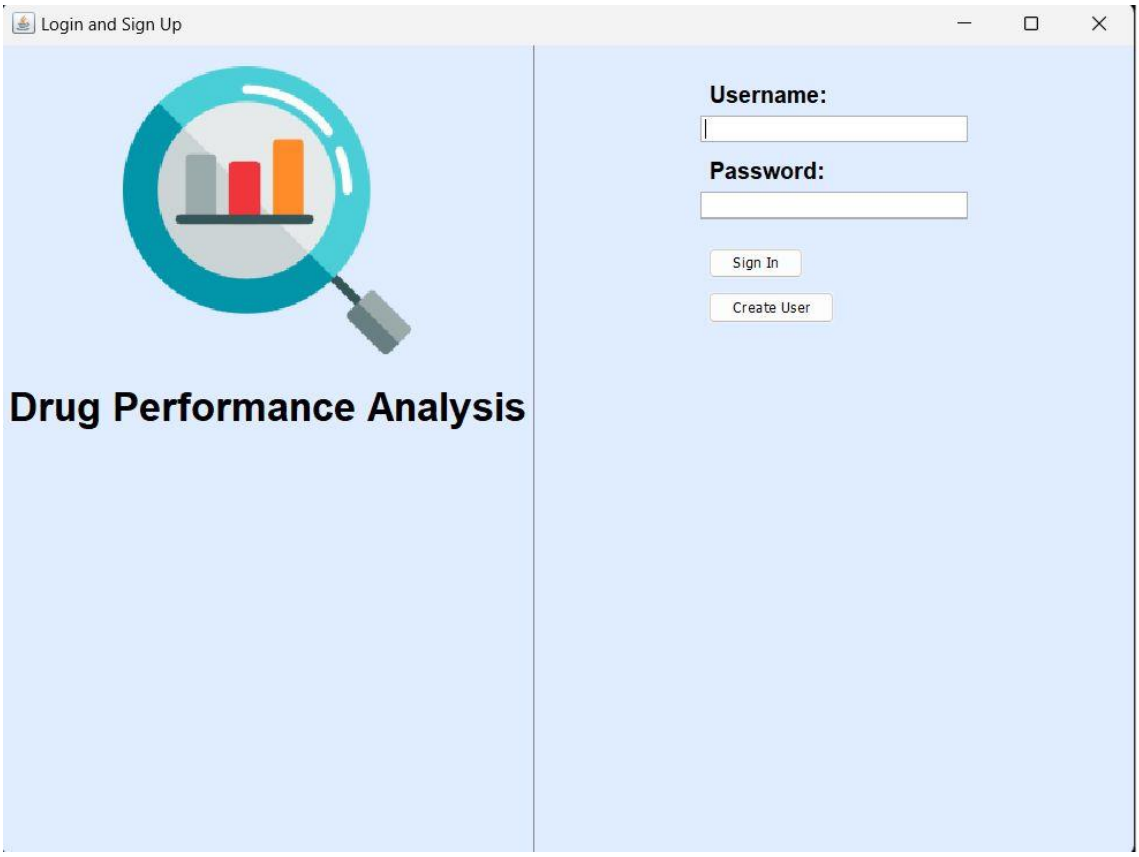

Edit Profile
Home
Review
Profile
Logout
Exit
Username:
imran646
Password:
imran
Full Name:
Mohammed Imran
Phone:
8179291622
Submit

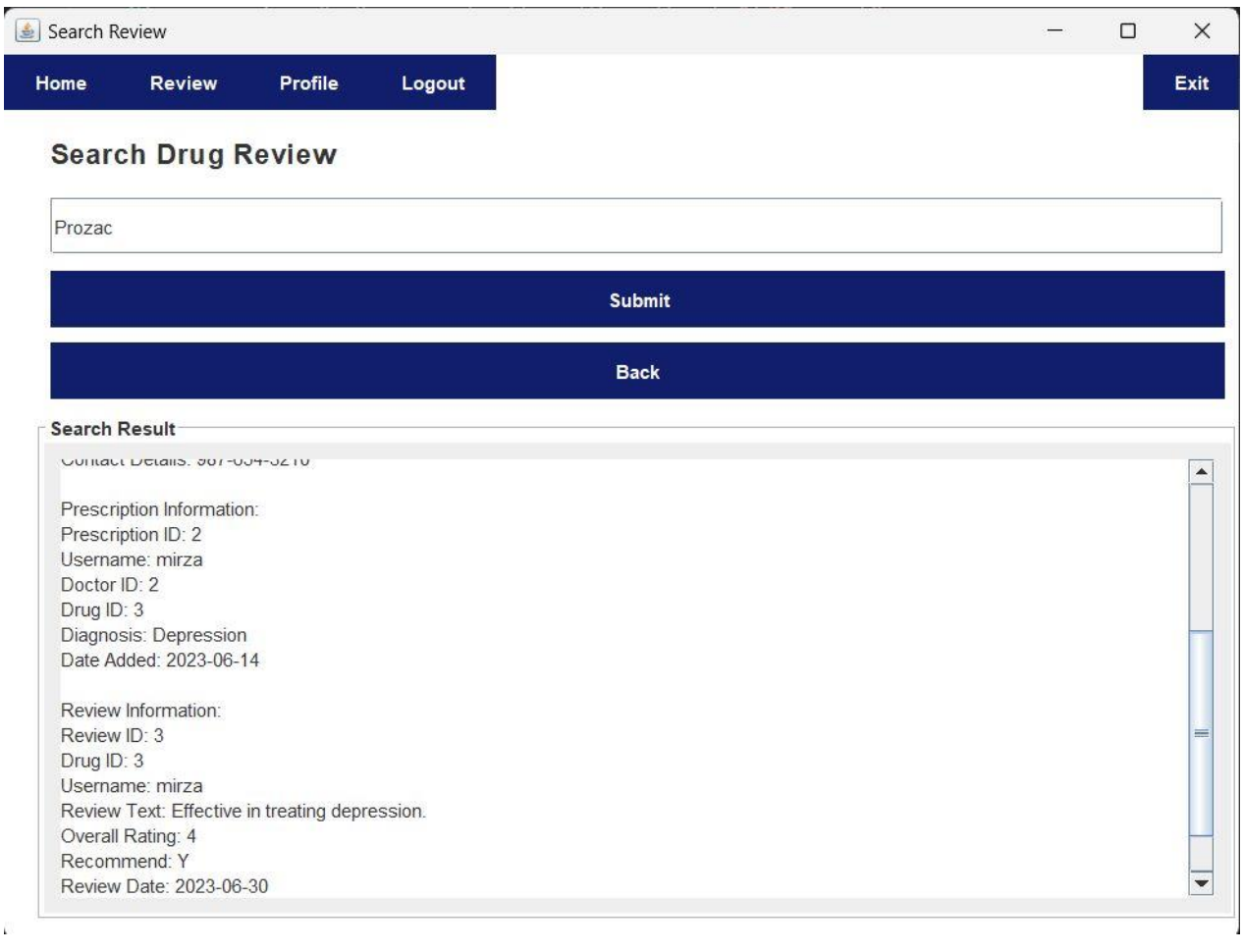

Message
Review Inserted Successfully
OK

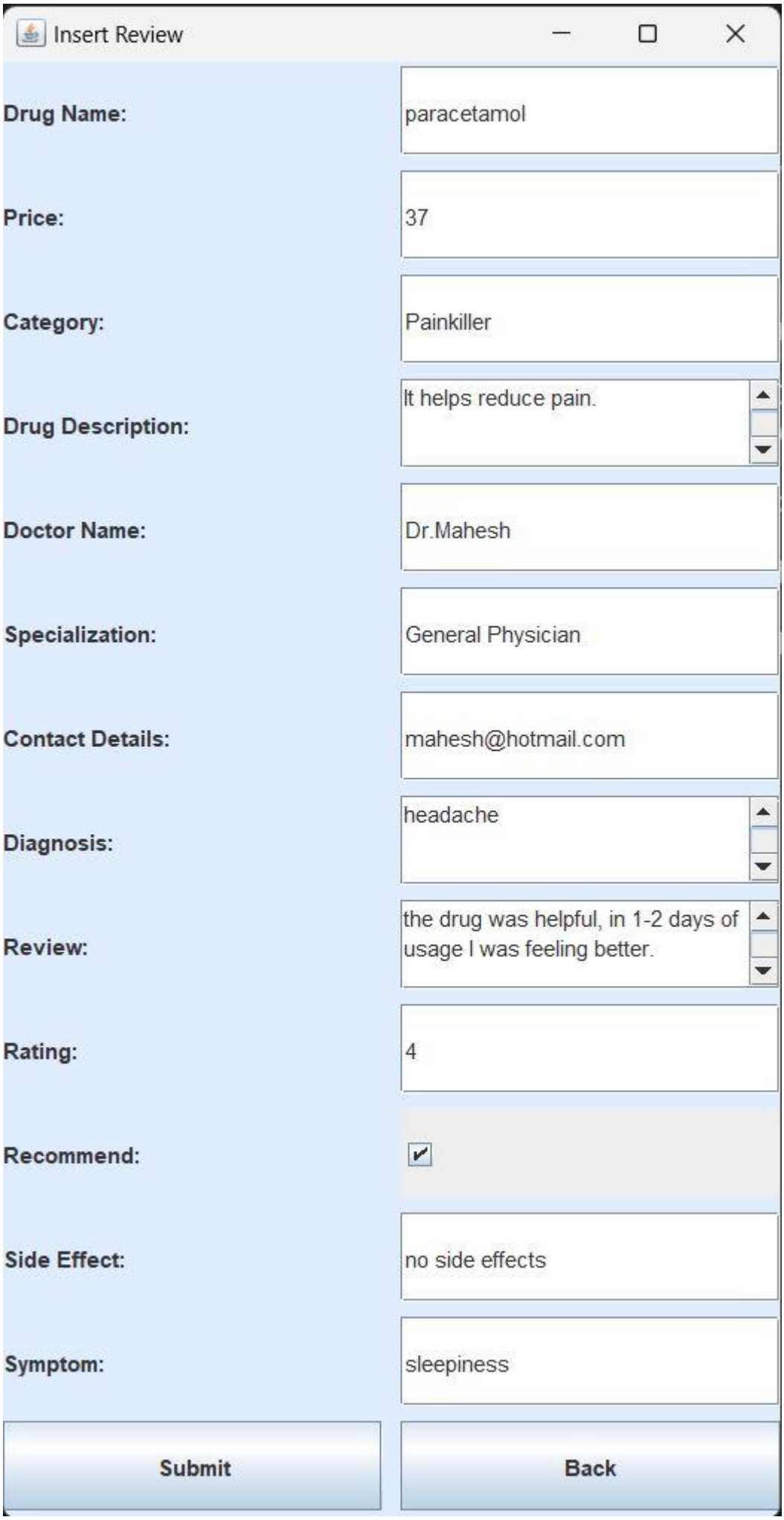

Create User
Register
Username:
Password:
Re-enter Password:
Full Name:
Phone:
Create
Exit
Back

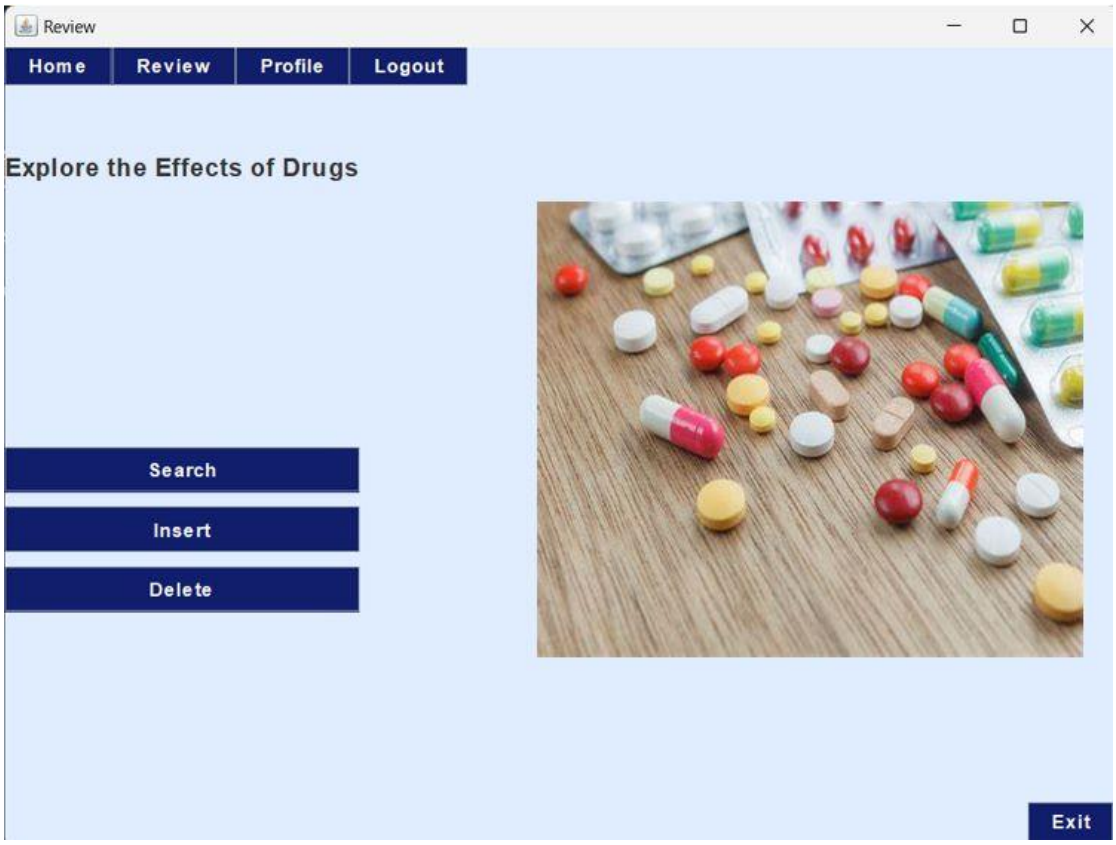

Profile
Home
Review
Profile
Logout
Edit Profile
Username:
imran646
Full Name:
Mohammed Imran
Phone:
8179291622
Last Login:
2023-06-26
Exit

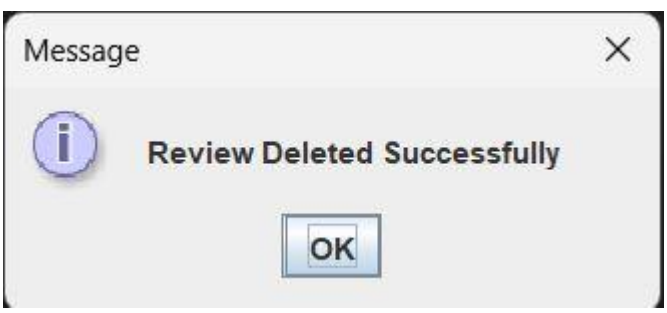

Project Homepage
Home
Review
Profile
Logout
Exit
Welcome!
This is a project that allows users to search for medicine reviews, insert new reviews, delete existing reviews, and update their own reviews.
Please explore the various functionalities available!


Login and Sign Up
Drug Performance Analysis
Username:
Password:
Sign In
Create User


Search Review
Home
Review
Profile
Logout
Exit
Search Drug Review
Prozac
Submit
Back
Search Result
Prescription Information:
Prescription ID: 2
Username: mirza
Doctor ID: 2
Drug ID: 3
Diagnosis: Depression
Date Added: 2023-06-14
Review Information:
Review ID: 3
Drug ID: 3
Username: mirza
Review Text: Effective in treating depression.
Overall Rating: 4
Recommend: Y
Review Date: 2023-06-30


Insert Review
Drug Name:
paracetamol
Price:
37
Category:
Painkiller
Drug Description:
It helps reduce pain.
Doctor Name:
Dr.Mahesh
Specialization:
General Physician
Contact Details:
mahesh@hotmail.com
Diagnosis:
headache
Review:
the drug was helpful, in 1-2 days of usage I was feeling better.
Rating:
4
Recommend:
Side Effect:
no side effects
Symptom:
sleepiness
Submit
Back


Review
Home
Review
Profile
Logout
Explore the Effects of Drugs
Search
Insert
Delete
Exit


Message
Review Deleted Successfully
OK

CONCLUSION :

In conclusion, drug performance analysis plays a crucial role in evaluating the effectiveness and safety of pharmaceutical products. By assessing various factors such as efficacy, side effects, dosing, and overall patient outcomes, drug performance analysis provides valuable insights into the real-world effectiveness of medications.

Additionally, drug performance analysis contributes to evidence-based medicine, facilitating the development of treatment guidelines and clinical protocols. It enhances our understanding of how drugs interact with specific patient populations, allowing for personalized medicine approaches and targeted interventions.

Overall, drug performance analysis is a vital component of the drug development and healthcare delivery process. It provides valuable information to healthcare professionals, regulators, and researchers, helping them make informed decisions and ultimately improving patient care and treatment outcomes.

FUTURE SCOPE OF THE PROJECT :

Real-world evidence (RWE) integration: Real-world data, collected from diverse patient populations and healthcare settings, can provide valuable insights into drug performance in real-life scenarios.

Artificial intelligence and machine learning: Advancements in artificial intelligence (AI) and machine learning (ML) techniques have the potential to revolutionize drug performance analysis.

Pharmacogenomics: Pharmacogenomics explores how an individual's genetic makeup influences their response to drugs.

Long-term safety monitoring: While clinical trials provide essential initial safety data, long-term safety monitoring is crucial to identify rare or delayed adverse events.

Comparative effectiveness research: Comparative effectiveness research compares the benefits and risks of different treatment options to inform clinical decision-making.

REFERENCES

Drugs & Medications A to Z - Drugs.com

Drug Performance Evaluation | Kaggle

Analytical techniques in pharmaceutical analysis: A review – ScienceDirect

UCI ML Drug Review dataset | Kaggle