EXPERIENCEXCHANGE - TIME-BASED ACCESS PASSES FOR UNIQUE EXPERIENCES

A

MINI PROJECT REPORT

for

JAVA - OBJECT ORIENTED PROGRAMMING LAB

Submitted by

DIVYA HIREN SAVLA Roll No. 49

DHARMIK VYAS Roll No. 62

DEPARTMENT OF COMPUTER ENGINEERING

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE
(An Autonomous Institute Affilated to University of Mumbai)
MUMBAI - 400 088, MAHARASHTRA (INDIA)

2024

EXPERIENCEXCHANGE - TIME-BASED ACCESS PASSES FOR UNIQUE EXPERIENCES

Introduction

The modern experience economy has created a demand for unique, curated, and local experiences that go beyond conventional ticketed events. This report presents **ExperienceXchange**, a platform designed to facilitate the buying, selling, and booking of **time-based access passes** for non-ticketed experiences such as workshops, guided tours, classes, and personal coaching sessions. The focus of the platform is on fair pricing, verified hosts, and a transparent booking process.

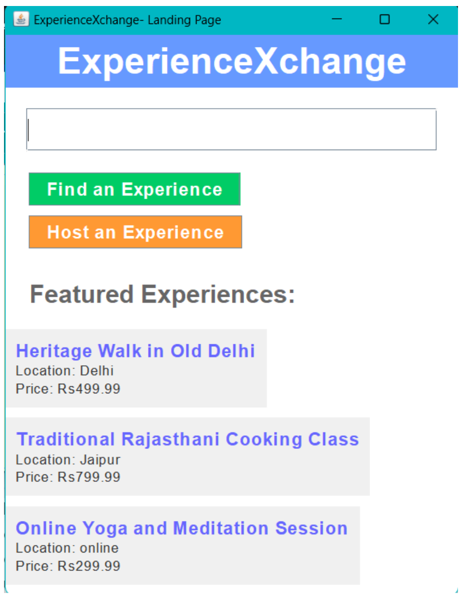
Objectives

The primary objective of **ExperienceXchange** is to provide a user-friendly, secure, and dynamic platform where hosts can offer, and users can discover and book, both local and online experiences. The key goals include:

- Offering a seamless discovery and booking system for unique experiences.
- Providing a robust host verification and management system.
- Ensuring secure payment processing and transparency for users and hosts.

Features and Pages

Landing Page


ExperienceXchange- Landing Page

ExperienceXchange

Find an Experience

Host an Experience

Featured Experiences:

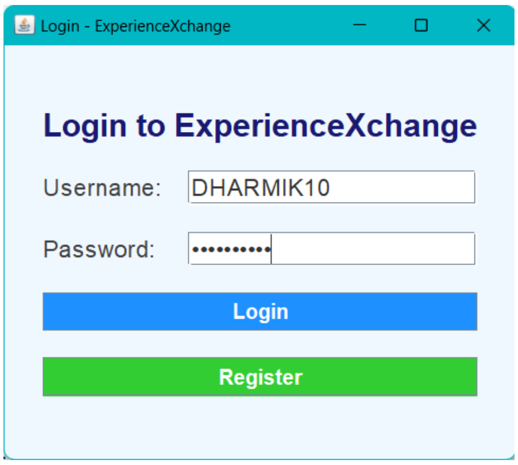
Heritage Walk in Old Delhi
Location: Delhi
Price: Rs499.99

Traditional Rajasthani Cooking Class
Location: Jaipur
Price: Rs799.99

Online Yoga and Meditation Session
Location: online
Price: Rs299.99

The landing page serves as the primary gateway, featuring the top experiences and offering easy access to both users and hosts via intuitive buttons like *Find an Experience* and *Host an Experience*.

User Authentication


Login - ExperienceXchange

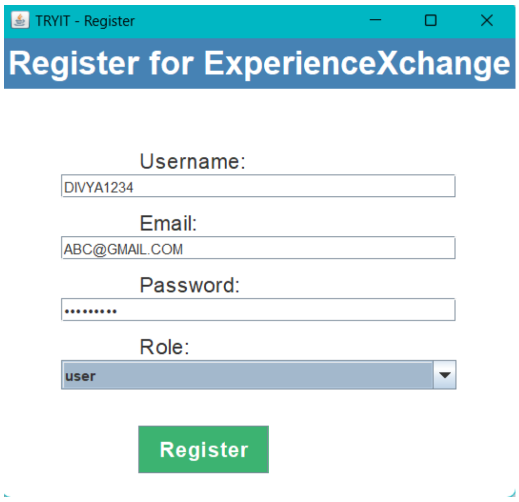
Login to ExperienceXchange

Username: DHARMIK10

Password: ••••••••••

Login

Register


TRYIT - Register

Register for ExperienceXchange

Username: DIVYA1234

Email: ABC@GMAIL.COM

Password: •••••••••

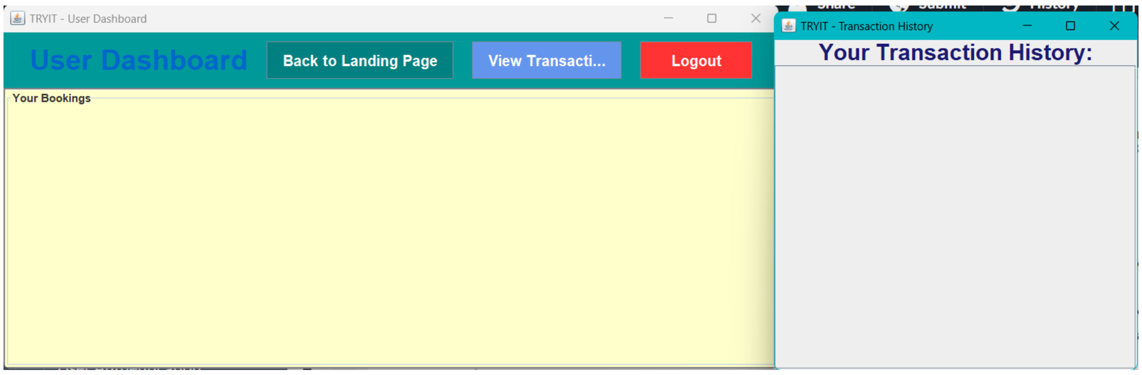
Role: user

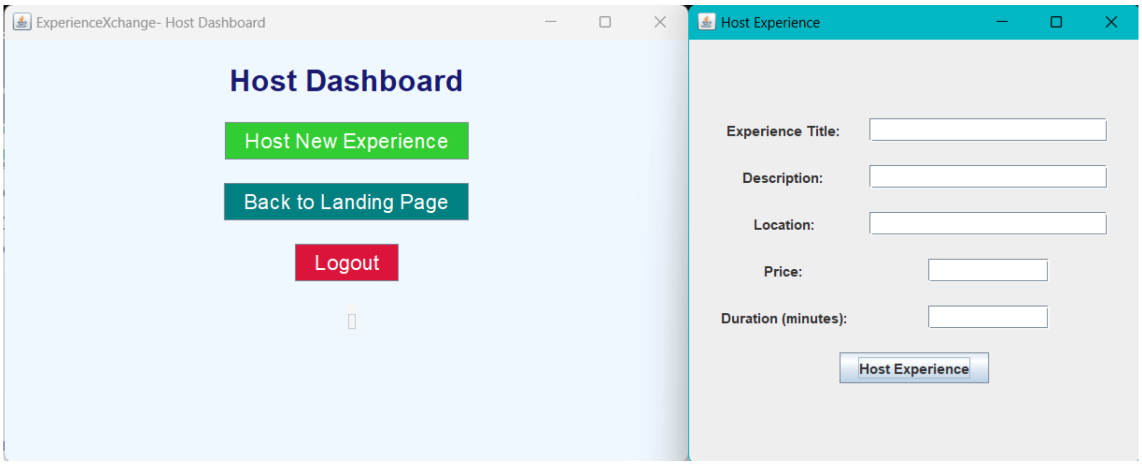
Register

- **Login/Signup Pages**: Users and hosts can sign up using email(Google, Facebook).

- **User Profiles**: Profiles for both user and hosts to manage bookings, listings, and schedules.

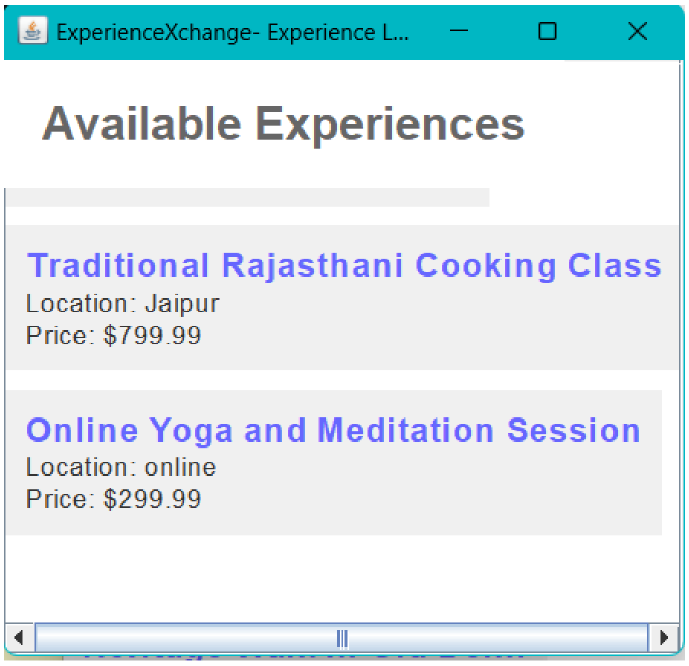
Dashboard





- **User Dashboard**: Displays booked experiences and upcoming schedules.
- **Host Dashboard**: Allows hosts to manage their listed experiences, update availability, and track bookings.

Experience Listing Page


ExperienceXchange- Experience L...

Available Experiences

Traditional Rajasthani Cooking Class
Location: Jaipur
Price: $799.99

Online Yoga and Meditation Session
Location: online
Price: $299.99

Hosts can list new experiences by adding details like type, location, price, and date/time, along with images or videos.

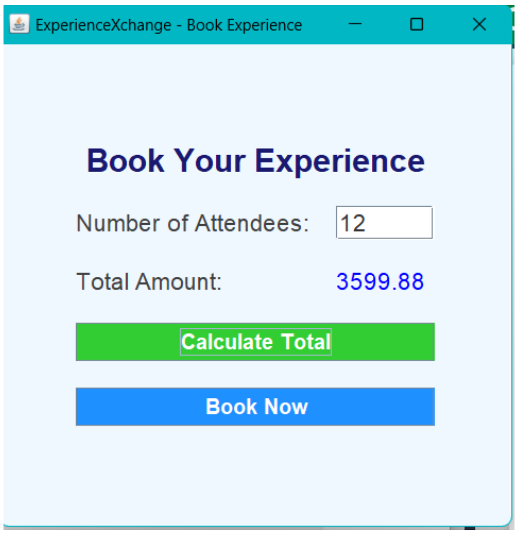
Search and Browse Experiences

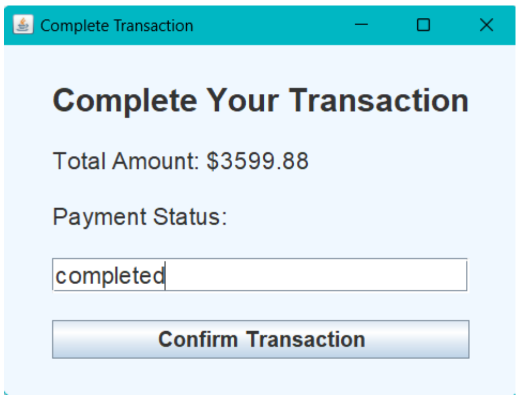
Users can explore experiences based on location, type, or date using filters and a dynamic search bar.

Host an Experience Page

Each experience has a dedicated page detailing the activity, date, time, price, location (or virtual link if online), and host information.

Booking and Transaction





A seamless booking process allows users to select the number of attendees and customize the experience (if applicable) before proceeding to a secure checkout.

Transaction History

Both users and hosts can access their transaction history to view past bookings, completed sessions, and payment details.

Database Structure

The platform's database is structured to store user information, experience listings, bookings, and transactions, ensuring an organized and efficient retrieval process.

Users Table

Field	**Type**	**Description**
id	INT	Primary key
username	VARCHAR(50)	Unique username
email	VARCHAR(100)	User email
password	VARCHAR(255)	Hashed password
role	ENUM('user', 'host')	Role of the user
date_created	TIMESTAMP	Account creation date

Experiences Table

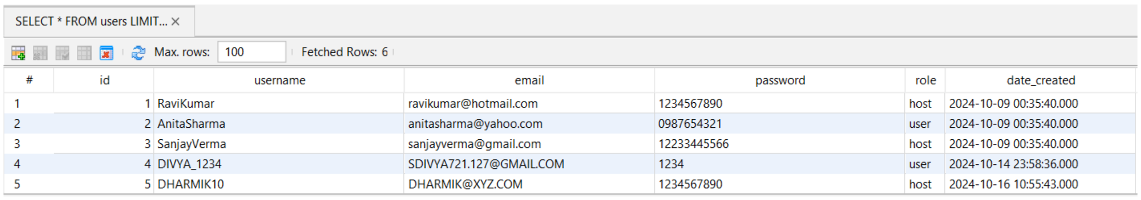
Field	Type	Description
id	INT	Primary key
title	VARCHAR(100)	Experience name
description	TEXT	Detailed description
location	VARCHAR(100)	Location (or "online" if virtual)
host_id	INT	Foreign key referencing Users.id
price	DECIMAL(10,2)	Price per session
duration	INT	Duration in minutes
date_listed	TIMESTAMP	Date experience was listed

Bookings Table

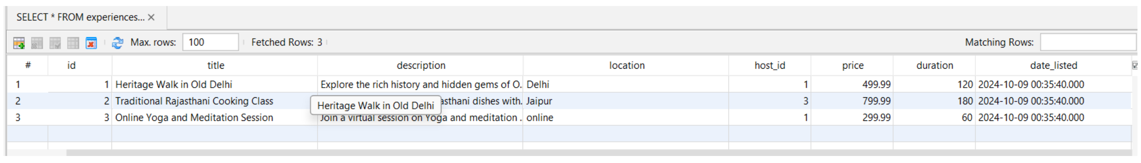
Field	Type	Description
id	INT	Primary key
experience_id	INT	Foreign key referencing Experiences.id
user_id	INT	Foreign key referencing Users.id
booking_date	TIMESTAMP	Date and time of booking
attendees	INT	Number of attendees
total_amount	DECIMAL(10,2)	Total amount paid

Transactions Table

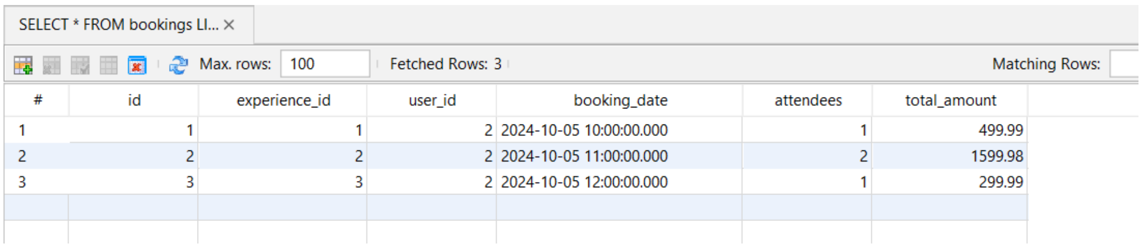
Field	Type	Description
id	INT	Primary key
booking_id	INT	Foreign key referencing Bookings.id
payment_status	ENUM('pending', 'completed')	Payment status
transaction_date	TIMESTAMP	Date of transaction
amount	DECIMAL(10,2)	Total price



#	id	username	email	password	role	date_created
1	1	RaviKumar	ravikumar@hotmail.com	1234567890	host	2024-10-09 00:35:40.000
2	2	AnitaSharma	anitasharma@yahoo.com	0987654321	user	2024-10-09 00:35:40.000
3	3	SanjayVerma	sanjayverma@gmail.com	12233445566	host	2024-10-09 00:35:40.000
4	4	DIVYA_1234	SDIVYA721.127@GMAIL.COM	1234	user	2024-10-14 23:58:36.000
5	5	DHARMIK10	DHARMIK@XYZ.COM	1234567890	host	2024-10-16 10:55:43.000

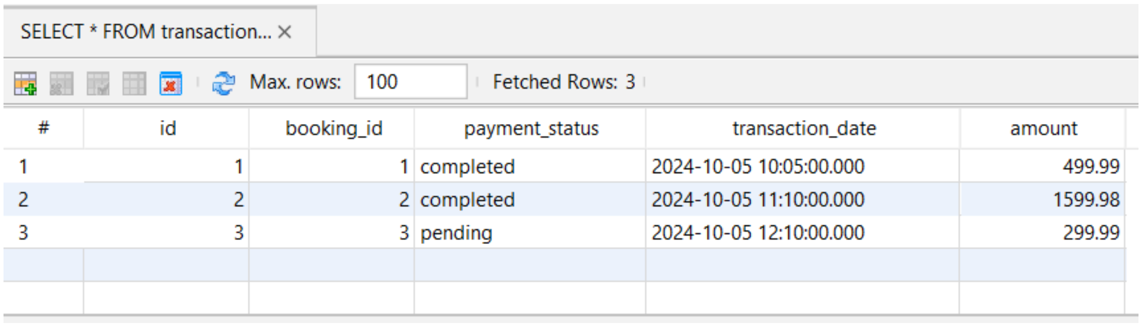


#	id	title	description	location	host_id	price	duration	date_listed
1	1	Heritage Walk in Old Delhi	Explore the rich history and hidden gems of O.	Delhi	1	499.99	120	2024-10-09 00:35:40.000
2	2	Traditional Rajasthani Cooking Class	asthani dishes with.	Jaipur	3	799.99	180	2024-10-09 00:35:40.000
3	3	Online Yoga and Meditation Session	Join a virtual session on Yoga and meditation .	online	1	299.99	60	2024-10-09 00:35:40.000


SELECT * FROM bookings LI...

Max. rows: 100 Fetched Rows: 3 Matching Rows:

#	id	experience_id	user_id	booking_date	attendees	total_amount
1	1	1	2	2024-10-05 10:00:00.000	1	499.99
2	2	2	2	2024-10-05 11:00:00.000	2	1599.98
3	3	3	2	2024-10-05 12:00:00.000	1	299.99


SELECT * FROM transaction...

Max. rows: 100 Fetched Rows: 3

#	id	booking_id	payment_status	transaction_date	amount
1	1	1	completed	2024-10-05 10:05:00.000	499.99
2	2	2	completed	2024-10-05 11:10:00.000	1599.98
3	3	3	pending	2024-10-05 12:10:00.000	299.99

Implementation code

LoginPage

```
package UI;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class LoginPage extends JFrame {

    public LoginPage() {
        initComponents();
    }

    private void initComponents() {
        setTitle("Login - ExperienceXchange");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Main Panel
        JPanel mainPanel = new JPanel();
        mainPanel.setBackground(new Color(240, 248, 255)); // Light
            Blue Background
        mainPanel.setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.fill = GridBagConstraints.HORIZONTAL;

        // Title
        JLabel titleLabel = new JLabel("Login to ExperienceXchange"
            , JLabel.CENTER);
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
```

```java
        titleLabel.setForeground(new Color(25, 25, 112)); // Midnight blue

        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.gridwidth = 2;
        mainPanel.add(titleLabel, gbc);

        // Username Label & Field
        JLabel usernameLabel = new JLabel("Username:");
        usernameLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        usernameLabel.setForeground(Color.DARK_GRAY);

        JTextField usernameField = new JTextField(15);
        usernameField.setFont(new Font("Arial", Font.PLAIN, 18));

        gbc.gridy = 1;
        gbc.gridwidth = 1;
        mainPanel.add(usernameLabel, gbc);

        gbc.gridx = 1;
        mainPanel.add(usernameField, gbc);

        // Password Label & Field
        JLabel passwordLabel = new JLabel("Password:");
        passwordLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        passwordLabel.setForeground(Color.DARK_GRAY);

        JPasswordField passwordField = new JPasswordField(15);
        passwordField.setFont(new Font("Arial", Font.PLAIN, 18));

        gbc.gridx = 0;
        gbc.gridy = 2;
        mainPanel.add(passwordLabel, gbc);

        gbc.gridx = 1;
        mainPanel.add(passwordField, gbc);

        // Login Button
        JButton loginButton = new JButton("Login");
        loginButton.setFont(new Font("Arial", Font.BOLD, 16));
        loginButton.setBackground(new Color(30, 144, 255)); // Dodger Blue
        loginButton.setForeground(Color.WHITE);
        loginButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                login(usernameField.getText(), new String(passwordField.getPassword()));
            }
        });

        gbc.gridx = 0;
        gbc.gridy = 3;
        gbc.gridwidth = 2;
        mainPanel.add(loginButton, gbc);

        // Register Button
        JButton registerButton = new JButton("Register");
        registerButton.setFont(new Font("Arial", Font.BOLD, 16));
        registerButton.setBackground(new Color(50, 205, 50)); // Lime Green
        registerButton.setForeground(Color.WHITE);
        registerButton.addActionListener(e -> openRegisterPage());

        gbc.gridy = 4; // Positioning the Register button below the Login button
        mainPanel.add(registerButton, gbc);

        add(mainPanel);
        setSize(400, 350);
        setLocationRelativeTo(null); // Center the frame
        setVisible(true);
    }

    private void login(String username, String password) {
        try (Connection conn = DatabaseConnection.getConnection()) {
            String sql = "SELECT id, role FROM Users WHERE username = ? AND password = ?";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, username);
            stmt.setString(2, password); // You should hash passwords for security
            ResultSet rs = stmt.executeQuery();

            if (rs.next()) {
                int userId = rs.getInt("id");
                String role = rs.getString("role");

                JOptionPane.showMessageDialog(this, "Login Successful!");

                // Open the appropriate dashboard based on the user's role
                if ("user".equalsIgnoreCase(role)) {
                    new UserDashboard(userId).setVisible(true);
                } else if ("host".equalsIgnoreCase(role)) {
                    new HostDashboard(userId).setVisible(true); // Ensure you have a HostDashboard class
                }

                this.dispose(); // Close the login window
            } else {
                JOptionPane.showMessageDialog(this, "Invalid username or password.", "Login Error", JOptionPane.ERROR_MESSAGE);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error during login: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    private void openRegisterPage() {
        new RegisterPage().setVisible(true); // Open the
            RegisterPage
    }

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

}
```

RegisterPage

```java
package UI;
import UI.DatabaseConnection;
import java.sql.Connection;
import java.sql.PreparedStatement;
import javax.swing.*;
import java.awt.*;

public class RegisterPage extends JFrame {
    private JTextField usernameField, emailField;
    private JPasswordField passwordField;
    private JComboBox<String> roleComboBox;

    public RegisterPage() {
        initComponents();
    }

    private void initComponents() {
        setTitle("TRYIT - Register");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(500, 600);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());

        JPanel topPanel = new JPanel();
        topPanel.setBackground(new Color(70, 130, 180));
        JLabel titleLabel = new JLabel("Register for
            ExperienceXchange");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 28));
        titleLabel.setForeground(Color.WHITE);
        topPanel.add(titleLabel);
        add(topPanel, BorderLayout.NORTH);

        JPanel formPanel = new JPanel();
        formPanel.setLayout(new BoxLayout(formPanel, BoxLayout.
            Y_AXIS));
        formPanel.setBackground(Color.WHITE);
        formPanel.setBorder(BorderFactory.createEmptyBorder(20, 50,
             20, 50));

        JLabel usernameLabel = new JLabel("Username:");
        usernameLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        usernameField = new JTextField(20);
        usernameField.setMaximumSize(new Dimension(Integer.
            MAX_VALUE, 40));

        JLabel emailLabel = new JLabel("Email:");
        emailLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        emailField = new JTextField(20);
        emailField.setMaximumSize(new Dimension(Integer.MAX_VALUE, 40));

        JLabel passwordLabel = new JLabel("Password:");
        passwordLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        passwordField = new JPasswordField(20);
        passwordField.setMaximumSize(new Dimension(Integer.MAX_VALUE, 40));

        JLabel roleLabel = new JLabel("Role:");
        roleLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        roleComboBox = new JComboBox<>(new String[]{"user", "host"});
        roleComboBox.setMaximumSize(new Dimension(Integer.MAX_VALUE, 40));

        JButton registerButton = new JButton("Register");
        registerButton.setFont(new Font("Arial", Font.BOLD, 18));
        registerButton.setBackground(new Color(60, 179, 113));
        registerButton.setForeground(Color.WHITE);
        registerButton.setPreferredSize(new Dimension(120, 40));
        registerButton.addActionListener(e -> register());

        formPanel.add(Box.createVerticalStrut(30));
        formPanel.add(usernameLabel);
        formPanel.add(usernameField);
        formPanel.add(Box.createVerticalStrut(10));
        formPanel.add(emailLabel);
        formPanel.add(emailField);
        formPanel.add(Box.createVerticalStrut(10));
        formPanel.add(passwordLabel);
        formPanel.add(passwordField);
        formPanel.add(Box.createVerticalStrut(10));
        formPanel.add(roleLabel);
        formPanel.add(roleComboBox);
        formPanel.add(Box.createVerticalStrut(30));
        formPanel.add(registerButton);

        add(formPanel, BorderLayout.CENTER);
        pack();
    }

    private void register() {
        String username = usernameField.getText();
        String email = emailField.getText();
        String password = new String(passwordField.getPassword());
        String role = (String) roleComboBox.getSelectedItem();
        if (username.isEmpty() || email.isEmpty() || password.isEmpty()) {
            JOptionPane.showMessageDialog(this, "All fields are required", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }
        try (Connection conn = DatabaseConnection.getConnection())
            {
            String sql = "INSERT INTO Users (username, email,
                password, role) VALUES (?, ?, ?, ?)";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, username);
            stmt.setString(2, email);
            stmt.setString(3, password);
            stmt.setString(4, role);
            stmt.executeUpdate();
            JOptionPane.showMessageDialog(this, "Registration
                successful!");
            new LoginPage().setVisible(true);
            this.dispose();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error: " + e.
                getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new RegisterPage().
            setVisible(true));
    }
}
```

LandingPage

```
package UI;
// LandingPage.javapackage UI;
import UI.DatabaseConnection;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.*;
import java.awt.*;

public class LandingPage extends JFrame {
    private int userId;

    public LandingPage(int userId) {
        this.userId = userId;
        initComponents();
        fetchFeaturedExperiences();
    }

    private void initComponents() {
        setTitle("ExperienceXchange- Landing Page");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(600, 600);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());

        // Header Panel
        JPanel headerPanel = new JPanel();
        headerPanel.setBackground(new Color(102, 153, 255));
```

```java
        JLabel logoLabel = new JLabel("ExperienceXchange");
        logoLabel.setFont(new Font("Arial", Font.BOLD, 36));
        logoLabel.setForeground(Color.WHITE);
        headerPanel.add(logoLabel);
        add(headerPanel, BorderLayout.NORTH);

        // Main Content Panel
        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new BoxLayout(mainPanel, BoxLayout.Y_AXIS));
        mainPanel.setBackground(Color.WHITE);
        mainPanel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));

        // Search Field
        JTextField searchField = new JTextField(20);
        searchField.setFont(new Font("Arial", Font.PLAIN, 18));
        searchField.setMaximumSize(new Dimension(Integer.MAX_VALUE, 50));

        // Buttons
        JButton findExperienceButton = new JButton("Find an Experience");
        findExperienceButton.setFont(new Font("Arial", Font.BOLD, 18));
        findExperienceButton.setBackground(new Color(0, 204, 102));
        findExperienceButton.setForeground(Color.WHITE);
        findExperienceButton.addActionListener(e -> new ExperienceListingPage(userId).setVisible(true));

        JButton hostExperienceButton = new JButton("Host an Experience");
        hostExperienceButton.setFont(new Font("Arial", Font.BOLD, 18));
        hostExperienceButton.setBackground(new Color(255, 153, 51));
        hostExperienceButton.setForeground(Color.WHITE);
        hostExperienceButton.addActionListener(e -> new RegisterPage().setVisible(true));

        // Featured Experiences Label
        JLabel featuredLabel = new JLabel("Featured Experiences:");
        featuredLabel.setFont(new Font("Arial", Font.BOLD, 24));
        featuredLabel.setForeground(new Color(102, 102, 102));

        // Adding Components to Main Panel
        mainPanel.add(searchField);
        mainPanel.add(Box.createVerticalStrut(20));
        mainPanel.add(findExperienceButton);
        mainPanel.add(Box.createVerticalStrut(10));
        mainPanel.add(hostExperienceButton);
        mainPanel.add(Box.createVerticalStrut(30));
        mainPanel.add(featuredLabel);

        add(mainPanel, BorderLayout.CENTER);
        pack();
    }

    private void fetchFeaturedExperiences() {
        try (Connection conn = DatabaseConnection.getConnection()) {
            String sql = "SELECT title, location, price FROM Experiences ORDER BY date_listed DESC LIMIT 5";
            PreparedStatement stmt = conn.prepareStatement(sql);
            ResultSet rs = stmt.executeQuery();

            // Panel to hold featured experiences
            JPanel featuredPanel = new JPanel();
            featuredPanel.setLayout(new BoxLayout(featuredPanel, BoxLayout.Y_AXIS));
            featuredPanel.setBackground(Color.WHITE);

            while (rs.next()) {
                String title = rs.getString("title");
                String location = rs.getString("location");
                double price = rs.getDouble("price");

                // Experience panel
                JPanel experiencePanel = new JPanel();
                experiencePanel.setBackground(new Color(240, 240, 240));
                experiencePanel.setLayout(new BoxLayout(experiencePanel, BoxLayout.Y_AXIS));
                experiencePanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

                JLabel titleLabel = new JLabel(title);
                titleLabel.setFont(new Font("Arial", Font.BOLD, 18));
                titleLabel.setForeground(new Color(102, 102, 255));

                JLabel locationLabel = new JLabel("Location: " + location);
                locationLabel.setFont(new Font("Arial", Font.PLAIN, 14));

                JLabel priceLabel = new JLabel("Price: Rs" + price);
                priceLabel.setFont(new Font("Arial", Font.PLAIN, 14));

                experiencePanel.add(titleLabel);
                experiencePanel.add(locationLabel);
                experiencePanel.add(priceLabel);
                experiencePanel.setAlignmentX(Component.LEFT_ALIGNMENT);

                experiencePanel.addMouseListener(new java.awt.event.MouseAdapter() {
                    public void mouseClicked(java.awt.event.MouseEvent evt) {
                        new ExperienceListingPage(userId).setVisible(true);
                    }
                });

                featuredPanel.add(experiencePanel);
                featuredPanel.add(Box.createVerticalStrut(10));
            }

            // Adding featured experiences to main panel
            add(featuredPanel, BorderLayout.SOUTH);
            revalidate();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error fetching experiences: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            try (Connection conn = DatabaseConnection.getConnection()) {
                String sql = "SELECT id FROM Users WHERE username = 'your_username'"; // Replace 'your_username' with the actual username
                PreparedStatement stmt = conn.prepareStatement(sql);
                ResultSet rs = stmt.executeQuery();
                if (rs.next()) {
                    int userId = rs.getInt("id");
                    new LandingPage(userId).setVisible(true);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        });
    }
}
```

HostDashboard

```java
package UI;

import javax.swing.*;
import java.awt.*;
import java.sql.*;

public class HostDashboard extends JFrame {
    private int hostId;
    private JPanel experiencePanel; // Declare experiencePanel as an instance variable

    public HostDashboard(int hostId) {
        this.hostId = hostId;
        initComponents();
        fetchHostedExperiences();
    }

    private void initComponents() {
        setTitle("ExperienceXchange- Host Dashboard");
        setSize(600, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
        panel.setBackground(new Color(240, 248, 255));

        JLabel dashboardLabel = new JLabel("Host Dashboard");
        dashboardLabel.setFont(new Font("Arial", Font.BOLD, 26));
        dashboardLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
        dashboardLabel.setForeground(new Color(25, 25, 112));

        JButton hostExperienceButton = new JButton("Host New
            Experience");
        hostExperienceButton.setFont(new Font("Arial", Font.PLAIN,
            18));
        hostExperienceButton.setBackground(new Color(50, 205, 50));
        hostExperienceButton.setForeground(Color.WHITE);
        hostExperienceButton.setFocusPainted(false);
        hostExperienceButton.setAlignmentX(Component.
            CENTER_ALIGNMENT);
        hostExperienceButton.setPreferredSize(new Dimension(200,
            50));
        hostExperienceButton.addActionListener(e -> new
            HostExperiencePage(hostId).setVisible(true));

        JButton logOutButton = new JButton("Logout");
        logOutButton.setFont(new Font("Arial", Font.PLAIN, 18));
        logOutButton.setBackground(new Color(220, 20, 60));
        logOutButton.setForeground(Color.WHITE);
        logOutButton.setFocusPainted(false);
        logOutButton.setAlignmentX(Component.CENTER_ALIGNMENT);
        logOutButton.setPreferredSize(new Dimension(200, 50));
        logOutButton.addActionListener(e -> {
            new LoginPage().setVisible(true);
            this.dispose();
        });

        JButton backButton = new JButton("Back to Landing Page");
        backButton.setFont(new Font("Arial", Font.PLAIN, 18));
        backButton.setBackground(new Color(0, 128, 128));
        backButton.setForeground(Color.WHITE);
        backButton.setFocusPainted(false);
        backButton.setAlignmentX(Component.CENTER_ALIGNMENT);
        backButton.setPreferredSize(new Dimension(200, 50));
        backButton.addActionListener(e -> {
            new LandingPage(hostId).setVisible(true);
            this.dispose();
        });

        // Initialize experiencePanel here
        experiencePanel = new JPanel();
        experiencePanel.setLayout(new BoxLayout(experiencePanel,
            BoxLayout.Y_AXIS));
        experiencePanel.setBackground(new Color(245, 245, 245));
        experiencePanel.setBorder(BorderFactory.createTitledBorder(
            "Your Hosted Experiences"));

        panel.add(Box.createRigidArea(new Dimension(0, 20)));
        panel.add(dashboardLabel);
        panel.add(Box.createRigidArea(new Dimension(0, 20)));
        panel.add(hostExperienceButton);
        panel.add(Box.createRigidArea(new Dimension(0, 20)));
        panel.add(backButton);
        panel.add(Box.createRigidArea(new Dimension(0, 20)));
        panel.add(logOutButton);
        panel.add(Box.createRigidArea(new Dimension(0, 20)));
        panel.add(experiencePanel); // Add experiencePanel to the
            main panel

        add(panel);
    }

    private void fetchHostedExperiences() {
        try (Connection conn = DatabaseConnection.getConnection())
            {
            String sql = "SELECT title, date_listed, price FROM
                Experiences WHERE host_id = ?";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setInt(1, hostId);
            ResultSet rs = stmt.executeQuery();

            // Clear previous experiences
            experiencePanel.removeAll();

            while (rs.next()) {
                String title = rs.getString("title");
                String dateListed = rs.getString("date_listed");
                double price = rs.getDouble("price");

                JLabel experienceLabel = new JLabel("Experience: "
                    + title + " | Listed on: " + dateListed + " |
                    Price: $" + price);
                experienceLabel.setFont(new Font("Arial", Font.
                    PLAIN, 16));
                experienceLabel.setForeground(new Color(25, 25,
                    112));
                experiencePanel.add(experienceLabel);
            }

            // Revalidate and repaint to update the UI
            experiencePanel.revalidate();
            experiencePanel.repaint();

        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error fetching
                hosted experiences: " + e.getMessage(), "Error",
                JOptionPane.ERROR_MESSAGE);
        }
    }

    public static void main(String[] args) {
        // Replace with actual host ID after login
        SwingUtilities.invokeLater(() -> {
            int hostId = 1; // Example host ID for testing; replace
                with actual retrieval from the database
            new HostDashboard(hostId).setVisible(true);
        });
    }
}
```

UserDashboard

```
package UI;
import UI.DatabaseConnection;
import java.awt.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.*;

public class UserDashboard extends JFrame {
    private int userId;

    public UserDashboard(int userId) {
        this.userId = userId;
        initComponents();
        fetchUserBookings();
    }

    private void initComponents() {
        setTitle("TRYIT - User Dashboard");
        getContentPane().setBackground(new Color(255, 250, 240));

        JLabel dashboardLabel = new JLabel("User Dashboard");
        dashboardLabel.setFont(new Font("Arial", Font.BOLD, 30));
        dashboardLabel.setForeground(new Color(0, 102, 204));

        JButton logOutButton = new JButton("Logout");
        logOutButton.setFont(new Font("Arial", Font.BOLD, 16));
        logOutButton.setBackground(new Color(255, 51, 51));
        logOutButton.setForeground(Color.WHITE);
        logOutButton.setFocusPainted(false);
        logOutButton.setPreferredSize(new Dimension(120, 40));
        logOutButton.addActionListener(e -> {
            new LoginPage().setVisible(true);
            this.dispose();
        });

        JButton backButton = new JButton("Back to Landing Page");
        backButton.setFont(new Font("Arial", Font.BOLD, 16));
        backButton.setBackground(new Color(0, 128, 128));
        backButton.setForeground(Color.WHITE);
        backButton.setFocusPainted(false);
        backButton.setPreferredSize(new Dimension(200, 40));
        backButton.addActionListener(e -> new LandingPage(userId).
            setVisible(true));

        // Assuming this is in a button click event or similar
JButton viewTransactionsButton = new JButton("View Transaction
    History");
```

```


        viewTransactionsButton.setFont(new Font("Arial", Font.BOLD,
            16));
        viewTransactionsButton.setBackground(new Color(100, 149,
           237));
        viewTransactionsButton.setForeground(Color.WHITE);
        viewTransactionsButton.setFocusPainted(false);
        viewTransactionsButton.setPreferredSize(new Dimension(160,
           40));
        viewTransactionsButton.addActionListener(e -> new
           TransactionHistoryPage(userId).setVisible(true));

        JPanel bookingsPanel = new JPanel();
        bookingsPanel.setBackground(new Color(255, 255, 204));
        bookingsPanel.setLayout(new BoxLayout(bookingsPanel,
           BoxLayout.Y_AXIS));
        bookingsPanel.setBorder(BorderFactory.createTitledBorder("
           Your Bookings"));

        setLayout(new BorderLayout());
        JPanel headerPanel = new JPanel();
        headerPanel.setBackground(new Color(0, 153, 153));
        headerPanel.setLayout(new FlowLayout(FlowLayout.CENTER, 20,
            10));
        headerPanel.add(dashboardLabel);
        headerPanel.add(backButton);
        headerPanel.add(viewTransactionsButton);
        headerPanel.add(logOutButton);

        JScrollPane scrollPane = new JScrollPane(bookingsPanel);
        scrollPane.setPreferredSize(new Dimension(600, 300));
        scrollPane.setBorder(BorderFactory.createLineBorder(Color.
           GRAY, 2));

        add(headerPanel, BorderLayout.NORTH);
        add(scrollPane, BorderLayout.CENTER);
        pack();
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }



// In the UserDashboard class, add:
private void fetchUserBookings() {
    try (Connection conn = DatabaseConnection.getConnection()) {
        String sql = "SELECT Experiences.title, Bookings.attendees,
            Bookings.total_amount FROM Bookings " +
                     "JOIN Experiences ON Bookings.experience_id =
                        Experiences.id " +
                     "WHERE Bookings.user_id = ?";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setInt(1, userId);
        ResultSet rs = stmt.executeQuery();

        while (rs.next()) {
```

```java
                String experienceTitle = rs.getString("title");
                int attendees = rs.getInt("attendees");
                double totalAmount = rs.getDouble("total_amount");

                // Add booking details to the dashboard
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error fetching
                bookings: " + e.getMessage());
        }
}


    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            try (Connection conn = DatabaseConnection.getConnection
                ()) {
                String sql = "SELECT id FROM Users WHERE username =
                    ''"; // Replace 'your_username' with the actual
                    username
                PreparedStatement stmt = conn.prepareStatement(sql)
                    ;
                ResultSet rs = stmt.executeQuery();
                if (rs.next()) {
                    int userId = rs.getInt("id");
                    new UserDashboard(userId).setVisible(true);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        });
    }}
```

ExperienceListingPage

```java
package UI;
import UI.DatabaseConnection;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.*;
import java.awt.*;

public class ExperienceListingPage extends JFrame {
    private int userId;

    public ExperienceListingPage(int userId) {
        this.userId = userId;
        initComponents();
        fetchExperiences();
    }

    private void initComponents() {
        setTitle("ExperienceXchange- Experience Listing");
        setSize(500, 600);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());

        // Main Content Panel
        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new BoxLayout(mainPanel, BoxLayout.Y_AXIS));
        mainPanel.setBackground(Color.WHITE);
        mainPanel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));

        // Title Label
        JLabel titleLabel = new JLabel("Available Experiences");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
        titleLabel.setForeground(new Color(102, 102, 102));
        mainPanel.add(titleLabel);

        add(mainPanel, BorderLayout.CENTER);
        pack();
    }

    private void fetchExperiences() {
    try (Connection conn = DatabaseConnection.getConnection()) {
        String sql = "SELECT id, title, location, price FROM Experiences ORDER BY date_listed DESC";
        PreparedStatement stmt = conn.prepareStatement(sql);
        ResultSet rs = stmt.executeQuery();

        // Panel to hold experience listings
        JPanel experiencePanel = new JPanel();
        experiencePanel.setLayout(new BoxLayout(experiencePanel, BoxLayout.Y_AXIS));
        experiencePanel.setBackground(Color.WHITE);

        while (rs.next()) {
            int experienceId = rs.getInt("id");
            String title = rs.getString("title");
            String location = rs.getString("location");
            double price = rs.getDouble("price");

            // Individual experience panel
            JPanel singleExperiencePanel = new JPanel();
            singleExperiencePanel.setBackground(new Color(240, 240, 240));
            singleExperiencePanel.setLayout(new BoxLayout(singleExperiencePanel, BoxLayout.Y_AXIS));
            singleExperiencePanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

            JLabel experienceTitle = new JLabel(title);
            experienceTitle.setFont(new Font("Arial", Font.BOLD, 18));
            experienceTitle.setForeground(new Color(102, 102, 255));

            JLabel experienceLocation = new JLabel("Location: " + location);
            experienceLocation.setFont(new Font("Arial", Font.PLAIN, 14));

            JLabel experiencePrice = new JLabel("Price: $" + price);
            experiencePrice.setFont(new Font("Arial", Font.PLAIN, 14));

            singleExperiencePanel.add(experienceTitle);
            singleExperiencePanel.add(experienceLocation);
            singleExperiencePanel.add(experiencePrice);
            singleExperiencePanel.setAlignmentX(Component.LEFT_ALIGNMENT);

            singleExperiencePanel.addMouseListener(new java.awt.event.MouseAdapter() {
                public void mouseClicked(java.awt.event.MouseEvent evt) {
                    new BookingPage(userId, experienceId, price).setVisible(true);
                }
            });

            experiencePanel.add(singleExperiencePanel);
            experiencePanel.add(Box.createVerticalStrut(10));
        }

        add(new JScrollPane(experiencePanel), BorderLayout.CENTER);
        revalidate();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Error fetching experiences: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}


    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            new ExperienceListingPage(1).setVisible(true);  // Example usage with a dummy user ID
        });
    }
}
```

HostExperiencePage

```
package UI;

import java.awt.*;
import java.math.BigDecimal;
import java.sql.*;
import javax.swing.*;

public class HostExperiencePage extends JFrame {
    private JTextField titleField, descriptionField, locationField, priceField, durationField;
```

```java
    private int hostId;

    public HostExperiencePage(int hostId) {
        this.hostId = hostId;
        initComponents();
    }

    private void initComponents() {
        setTitle("Host Experience");

        // Create the main panel
        JPanel mainPanel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);

        // Title field
        JLabel titleLabel = new JLabel("Experience Title:");
        titleField = new JTextField(20);
        gbc.gridx = 0;
        gbc.gridy = 0;
        mainPanel.add(titleLabel, gbc);
        gbc.gridx = 1;
        mainPanel.add(titleField, gbc);

        // Description field
        JLabel descriptionLabel = new JLabel("Description:");
        descriptionField = new JTextField(20);
        gbc.gridx = 0;
        gbc.gridy = 1;
        mainPanel.add(descriptionLabel, gbc);
        gbc.gridx = 1;
        mainPanel.add(descriptionField, gbc);

        // Location field
        JLabel locationLabel = new JLabel("Location:");
        locationField = new JTextField(20);
        gbc.gridx = 0;
        gbc.gridy = 2;
        mainPanel.add(locationLabel, gbc);
        gbc.gridx = 1;
        mainPanel.add(locationField, gbc);

        // Price field
        JLabel priceLabel = new JLabel("Price:");
        priceField = new JTextField(10);
        gbc.gridx = 0;
        gbc.gridy = 3;
        mainPanel.add(priceLabel, gbc);
        gbc.gridx = 1;
        mainPanel.add(priceField, gbc);

        // Duration field
        JLabel durationLabel = new JLabel("Duration (minutes):");
        durationField = new JTextField(10);
        gbc.gridx = 0;
        gbc.gridy = 4;
        mainPanel.add(durationLabel, gbc);
        gbc.gridx = 1;
        mainPanel.add(durationField, gbc);

        // Submit button
        JButton submitButton = new JButton("Host Experience");
        submitButton.addActionListener(e -> createExperience());
        gbc.gridx = 0;
        gbc.gridy = 5;
        gbc.gridwidth = 2;
        mainPanel.add(submitButton, gbc);

        add(mainPanel);
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
    }

    private void createExperience() {
        try {
            String title = titleField.getText();
            String description = descriptionField.getText();
            String location = locationField.getText();
            BigDecimal price = new BigDecimal(priceField.getText())
                ;
            int duration = Integer.parseInt(durationField.getText()
                );

            try (Connection conn = DatabaseConnection.getConnection
                ()) {
                String sql = "INSERT INTO Experiences (title,
                    description, location, host_id, price, duration)
                     VALUES (?, ?, ?, ?, ?, ?)";
                PreparedStatement stmt = conn.prepareStatement(sql)
                    ;
                stmt.setString(1, title);
                stmt.setString(2, description);
                stmt.setString(3, location);
                stmt.setInt(4, hostId);
                stmt.setBigDecimal(5, price);
                stmt.setInt(6, duration);
                stmt.executeUpdate();
                JOptionPane.showMessageDialog(this, "Experience
                    hosted successfully!");
                this.dispose();
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error: " + e.
                getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

TransactionConfirmationPage

```java
package UI;

import javax.swing.*;
import java.awt.*;
import java.sql.Connection;
import java.sql.PreparedStatement;

public class TransactionConfirmationPage extends JFrame {

    private int bookingId; // To link the transaction to the
        booking
    private double totalAmount; // The total amount for the
        transaction

    public TransactionConfirmationPage(int bookingId, double
        totalAmount) {
        this.bookingId = bookingId;
        this.totalAmount = totalAmount;
        initComponents();
    }

    private void initComponents() {
        setTitle("Complete Transaction");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setSize(400, 300);
        setLocationRelativeTo(null); // Center the frame

        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new GridBagLayout());
        mainPanel.setBackground(new Color(240, 248, 255)); // Light
            Blue Background
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.fill = GridBagConstraints.HORIZONTAL;

        // Title
        JLabel titleLabel = new JLabel("Complete Your Transaction",
            JLabel.CENTER);
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
        mainPanel.add(titleLabel, gbc);

        // Total Amount Label
        JLabel amountLabel = new JLabel("Total Amount: $" + String.
           format("%.2f", totalAmount));
        amountLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        gbc.gridy = 1;
        mainPanel.add(amountLabel, gbc);

        // Payment Status Label
        JLabel paymentStatusLabel = new JLabel("Payment Status:");
        paymentStatusLabel.setFont(new Font("Arial", Font.PLAIN,
           18));
        gbc.gridy = 2;
        mainPanel.add(paymentStatusLabel, gbc);

        // Payment Status (Placeholder for now)
        JTextField paymentStatusField = new JTextField("completed",
            15);
        paymentStatusField.setFont(new Font("Arial", Font.PLAIN,
           18));
        gbc.gridy = 3;
        mainPanel.add(paymentStatusField, gbc);

        // Confirm Button
        JButton confirmButton = new JButton("Confirm Transaction");
        confirmButton.setFont(new Font("Arial", Font.BOLD, 16));
        confirmButton.addActionListener(e -> completeTransaction())
            ;
        gbc.gridy = 4;
        mainPanel.add(confirmButton, gbc);

        add(mainPanel);
        setVisible(true);
    }

    private void completeTransaction() {
    try (Connection conn = DatabaseConnection.getConnection()) {
        String sql = "INSERT INTO Transactions (booking_id,
            payment_status, amount) VALUES (?, 'completed', ?)";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setInt(1, bookingId);
        stmt.setDouble(2, totalAmount);
        stmt.executeUpdate();

        JOptionPane.showMessageDialog(this, "Transaction Completed
            Successfully!");
        dispose(); // Close the transaction confirmation window
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Error completing
            transaction: " + e.getMessage(), "Error", JOptionPane.
            ERROR_MESSAGE);
    }
}

}
```

TransactionHistoryPage

```java
package UI;

import java.awt.*;
import java.sql.*;
import javax.swing.*;

public class TransactionHistoryPage extends JFrame {
    private int userId; // Actual user ID

    public TransactionHistoryPage(int userId) {
        this.userId = userId; // Store the user ID
        initComponents();
        fetchTransactionHistory(); // Fetch transaction history for
             the user
    }

    private void initComponents() {
        setTitle("TRYIT - Transaction History");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setSize(600, 500);
        setLocationRelativeTo(null);

        JPanel mainPanel = new JPanel(new BorderLayout());
        JLabel historyLabel = new JLabel("Your Transaction History:
           ");
        historyLabel.setFont(new Font("Arial", Font.BOLD, 24));
        historyLabel.setHorizontalAlignment(SwingConstants.CENTER);
        historyLabel.setForeground(new Color(25, 25, 112));

        mainPanel.add(historyLabel, BorderLayout.NORTH);

        JPanel transactionPanel = new JPanel();
        transactionPanel.setLayout(new BoxLayout(transactionPanel,
           BoxLayout.Y_AXIS));
        JScrollPane scrollPane = new JScrollPane(transactionPanel);
        mainPanel.add(scrollPane, BorderLayout.CENTER);

        add(mainPanel);
    }

    private void fetchTransactionHistory() {
        try (Connection conn = DatabaseConnection.getConnection())
           {
            String sql = "SELECT Experiences.title, Transactions.
               transaction_date, Transactions.amount, Transactions.
               payment_status " +
                         "FROM Transactions " +
                         "JOIN Bookings ON Transactions.booking_id
                            " +
                         "JOIN Experiences ON Bookings.
                            experience_id  " +
                         "WHERE Bookings.user_id = ?";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setInt(1, userId); // Set the actual user ID
            ResultSet rs = stmt.executeQuery();

            JPanel transactionPanel = (JPanel) ((JScrollPane)
               getContentPane().getComponent(1)).getViewport().
               getView();
            transactionPanel.removeAll(); // Clear previous entries

            if (!rs.next()) {
                JLabel noTransactionsLabel = new JLabel("No
                   transactions found.");
                noTransactionsLabel.setFont(new Font("Arial", Font.
                   ITALIC, 16));
                noTransactionsLabel.setForeground(new Color(255, 0,
                    0));
                transactionPanel.add(noTransactionsLabel);
            } else {
                do {
                    String title = rs.getString("title");
                    String date = rs.getString("transaction_date");
                    double amount = rs.getDouble("amount");
                    String status = rs.getString("payment_status");

```

```java
                    JPanel itemPanel = new JPanel(new BorderLayout());
                    itemPanel.setBorder(BorderFactory.createMatteBorder(0, 0, 1, 0, Color.LIGHT_GRAY));
                    itemPanel.setBackground(new Color(245, 245, 245));

                    JLabel titleLabel = new JLabel("Experience: " + title);
                    titleLabel.setFont(new Font("Arial", Font.BOLD, 16));
                    titleLabel.setForeground(new Color(0, 100, 0));

                    JLabel detailsLabel = new JLabel("Date: " + date + " | Amount: $" + amount + " | Status: " + status);
                    detailsLabel.setFont(new Font("Arial", Font.PLAIN, 14));
                    detailsLabel.setForeground(new Color(105, 105, 105));

                    itemPanel.add(titleLabel, BorderLayout.NORTH);
                    itemPanel.add(detailsLabel, BorderLayout.CENTER);
                    transactionPanel.add(itemPanel);
                } while (rs.next());
            }

            transactionPanel.revalidate(); // Refresh the panel
            transactionPanel.repaint(); // Update the UI
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error fetching transaction history: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

BookingPage

```java
package UI;

import UI.DatabaseConnection;
import java.awt.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.*;

public class BookingPage extends JFrame {

    private int userId;
    private int experienceId;
    private JTextField attendeesField;
    private JLabel totalAmountLabel;
    private double experiencePrice;

    public BookingPage(int userId, int experienceId, double experiencePrice) {
        this.userId = userId;
        this.experienceId = experienceId;
        this.experiencePrice = experiencePrice;
        initComponents();
    }

    private void initComponents() {
        setTitle("ExperienceXchange - Book Experience");

        // Set main panel layout and background color
        JPanel mainPanel = new JPanel();
        mainPanel.setBackground(new Color(240, 248, 255)); // Light blue background
        mainPanel.setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.fill = GridBagConstraints.HORIZONTAL;

        // Heading
        JLabel headerLabel = new JLabel("Book Your Experience", JLabel.CENTER);
        headerLabel.setFont(new Font("Arial", Font.BOLD, 24));
        headerLabel.setForeground(new Color(25, 25, 112)); // Midnight blue color
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.gridwidth = 2;
        mainPanel.add(headerLabel, gbc);

        // Attendees label and field
        JLabel attendeesLabel = new JLabel("Number of Attendees:");
        attendeesLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        attendeesLabel.setForeground(Color.DARK_GRAY);
        gbc.gridy = 1;
        gbc.gridwidth = 1;
        mainPanel.add(attendeesLabel, gbc);

        attendeesField = new JTextField(5);
        attendeesField.setFont(new Font("Arial", Font.PLAIN, 18));
        gbc.gridx = 1;
        mainPanel.add(attendeesField, gbc);

        // Total amount label
        JLabel amountLabel = new JLabel("Total Amount:");
        amountLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        amountLabel.setForeground(Color.DARK_GRAY);
        gbc.gridx = 0;
        gbc.gridy = 2;
        mainPanel.add(amountLabel, gbc);

        totalAmountLabel = new JLabel("0.00");
        totalAmountLabel.setFont(new Font("Arial", Font.PLAIN, 18));
        totalAmountLabel.setForeground(Color.BLUE);
        gbc.gridx = 1;
        mainPanel.add(totalAmountLabel, gbc);

        // Calculate button
        JButton calculateButton = new JButton("Calculate Total");
        calculateButton.setFont(new Font("Arial", Font.BOLD, 16));
        calculateButton.setBackground(new Color(50, 205, 50)); // Lime green background
        calculateButton.setForeground(Color.WHITE);
        calculateButton.addActionListener(e -> calculateTotal());
        gbc.gridx = 0;
        gbc.gridy = 3;
        gbc.gridwidth = 2;
        mainPanel.add(calculateButton, gbc);

        // Book button
        JButton bookButton = new JButton("Book Now");
        bookButton.setFont(new Font("Arial", Font.BOLD, 16));
        bookButton.setBackground(new Color(30, 144, 255)); // Dodger blue background
        bookButton.setForeground(Color.WHITE);
        bookButton.addActionListener(e -> bookExperience());
        gbc.gridy = 4;
        mainPanel.add(bookButton, gbc);

        // Add padding around the content
        mainPanel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));

        // Set layout and window properties
        add(mainPanel);
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null); // Center on screen
        setVisible(true);
    }

    private void calculateTotal() {
        try {
            int attendees = Integer.parseInt(attendeesField.getText());
            double totalAmount = attendees * experiencePrice;
            totalAmountLabel.setText(String.format("%.2f", totalAmount));
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this, "Please enter a valid number of attendees.", "Input Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    private void bookExperience() {
    try {
        int attendees = Integer.parseInt(attendeesField.getText());
        double totalAmount = attendees * experiencePrice;

        try (Connection conn = DatabaseConnection.getConnection()) {
            String sql = "INSERT INTO Bookings (experience_id, user_id, attendees, total_amount) VALUES (?, ?, ?, ?)";
            PreparedStatement stmt = conn.prepareStatement(sql, PreparedStatement.RETURN_GENERATED_KEYS);
            stmt.setInt(1, experienceId);
            stmt.setInt(2, userId);
            stmt.setInt(3, attendees);
            stmt.setDouble(4, totalAmount);
            stmt.executeUpdate();

            // Get the generated booking ID
            ResultSet generatedKeys = stmt.getGeneratedKeys();
            if (generatedKeys.next()) {
                int bookingId = generatedKeys.getInt(1);
                // Open transaction confirmation page
                new TransactionConfirmationPage(bookingId, totalAmount).setVisible(true);
                JOptionPane.showMessageDialog(this, "Booking Successful! Please confirm the transaction.");
                this.dispose(); // Close the booking page
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error booking experience: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(this, "Please enter a valid number of attendees.", "Input Error", JOptionPane.ERROR_MESSAGE);
    }
}

}
```

DatabaseConnection

```java
package UI;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
/**
 *
 * @author sdivy
 */
public class DatabaseConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/test";
    private static final String USERNAME = "root"; // Replace with your MySQL username
    private static final String PASSWORD = ""; // Replace with your MySQL password

    public static Connection getConnection() throws SQLException {
        Connection connection = null;
        try {
            connection = DriverManager.getConnection(URL, USERNAME,
                PASSWORD);
            System.out.println("Database connected!");
        } catch (SQLException e) {
            System.err.println("Database connection failed: " + e.
               getMessage());
            throw e;
        }
        return connection;
    }
}
```

Conclusion

ExperienceXchange offers a comprehensive platform that connects users with unique experiences, both online and in-person. By focusing on an intuitive user interface, seamless booking and payment processes, and a robust database structure, the platform provides a complete solution for experience seekers and hosts alike.

Bibliography

[1] Oracle, *Creating a GUI with Swing*, 2024. Accessed: 2024-10-19. Available: https://docs.oracle.com/javase/tutorial/uiswing/.

[2] Oracle, *JDBC Basics*, 2024. Accessed: 2024-10-19. Available: https://docs.oracle.com/javase/tutorial/jdbc/.

[3] Stripe, *Stripe Java API*, 2024. Accessed: 2024-10-19. Available: https://stripe.com/docs/api/java.

[4] Spring, *Spring Boot Documentation*, 2024. Accessed: 2024-10-19. Available: https://spring.io/projects/spring-boot.

[5] Apache, *Apache Tomcat Documentation*, 2024. Accessed: 2024-10-19. Available: https://tomcat.apache.org/tomcat-9.0-doc/.