EXPLORE INN

HOTEL MANAGEMENT SYSTEM

SANSITA
22011101049

KARTHIKA
22011101052

JAVA INN SIGHTS

OUR TEAM

SANSITA

220111101049

-HomePage

-LoginPage

-AboutPage

KARTHIKA

220111101052

-SelectHotel

-BookingWindow

-HelpPage

INTRODUCTION

Explore Inn is a software application that provides an integrated solution for managing various aspects of a hotel's operations. It streamlines and automates a wide range of tasks, helping hotel staff and management deliver efficient and high-quality services to guests.

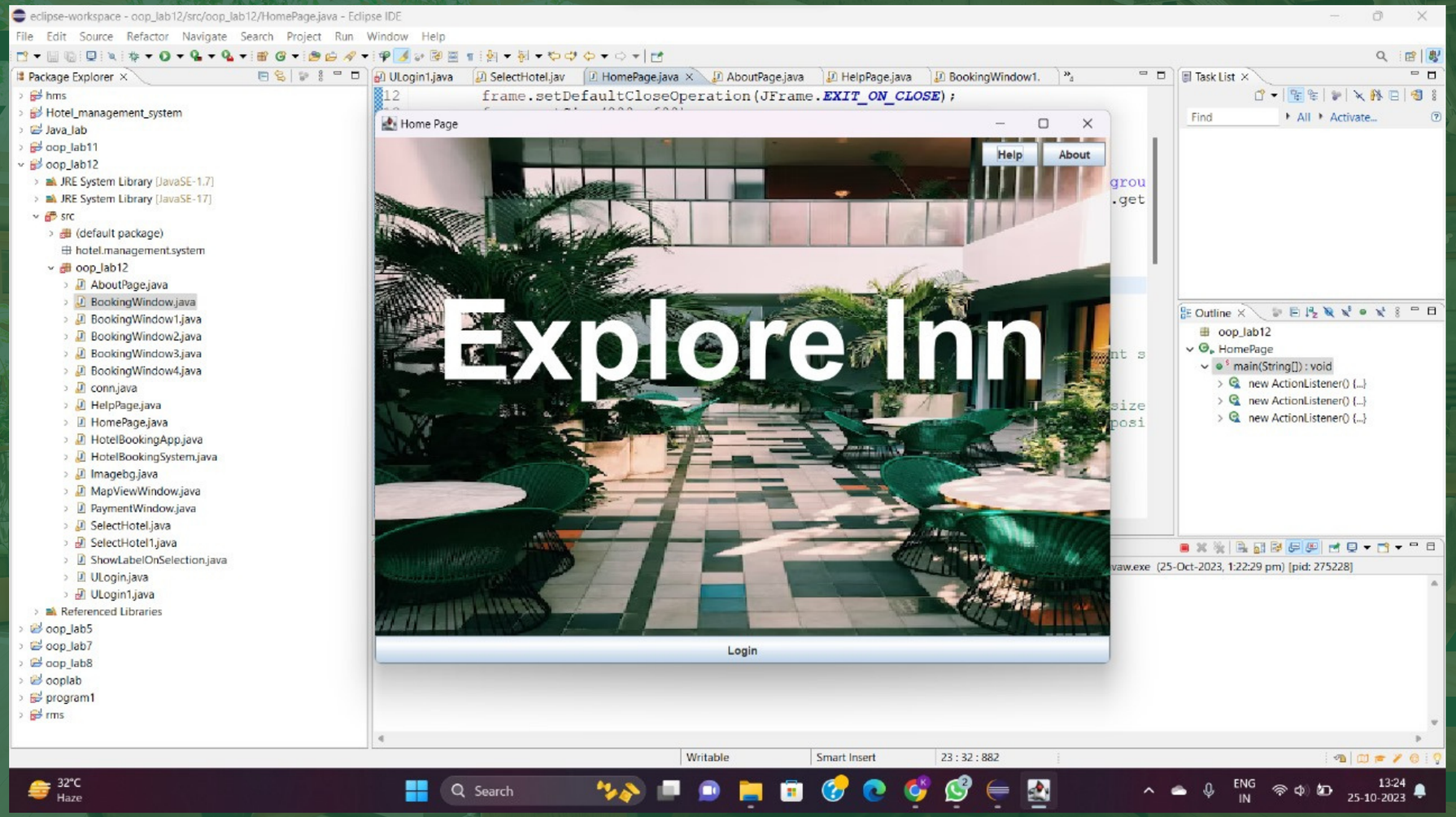
Features:

- User Authentication and Access Control
- Reservation Management
- Billing
- Inventory Management
- Online Booking and Payment
- Integration
- User Interfaces
- Database Management using HeidiSQL
- Error Handling and Logging

HOME PAGE

- Swing Library: The javax.swing package is used for creating graphical user interface (GUI) components.
- AWT Library: The java.awt package is used for creating and managing the GUI components.
- Event Handling: The code uses event handling for button clicks using ActionListener and ActionEvent.
- JFrame: A class from Swing that provides the main window for the GUI application.
- JLabel: Used to display a short string or an image icon.
- JPanel: A container that provides space to add other components.
- BorderLayout: A layout manager used to arrange components at the borders (North, South, East, West) and the center of a container.
- ImageIcon: Used to display images.
- ActionEvent and ActionListener: Used for handling events triggered by user actions like button clicks.
- Dimension: Used to set the size of components.
- Insets: Used to create padding around the button.
- Image: Used to handle images.
- Dispose(): Used to close the current window.
- Opaque Property: Used to make the background of the panel transparent to display the background image.

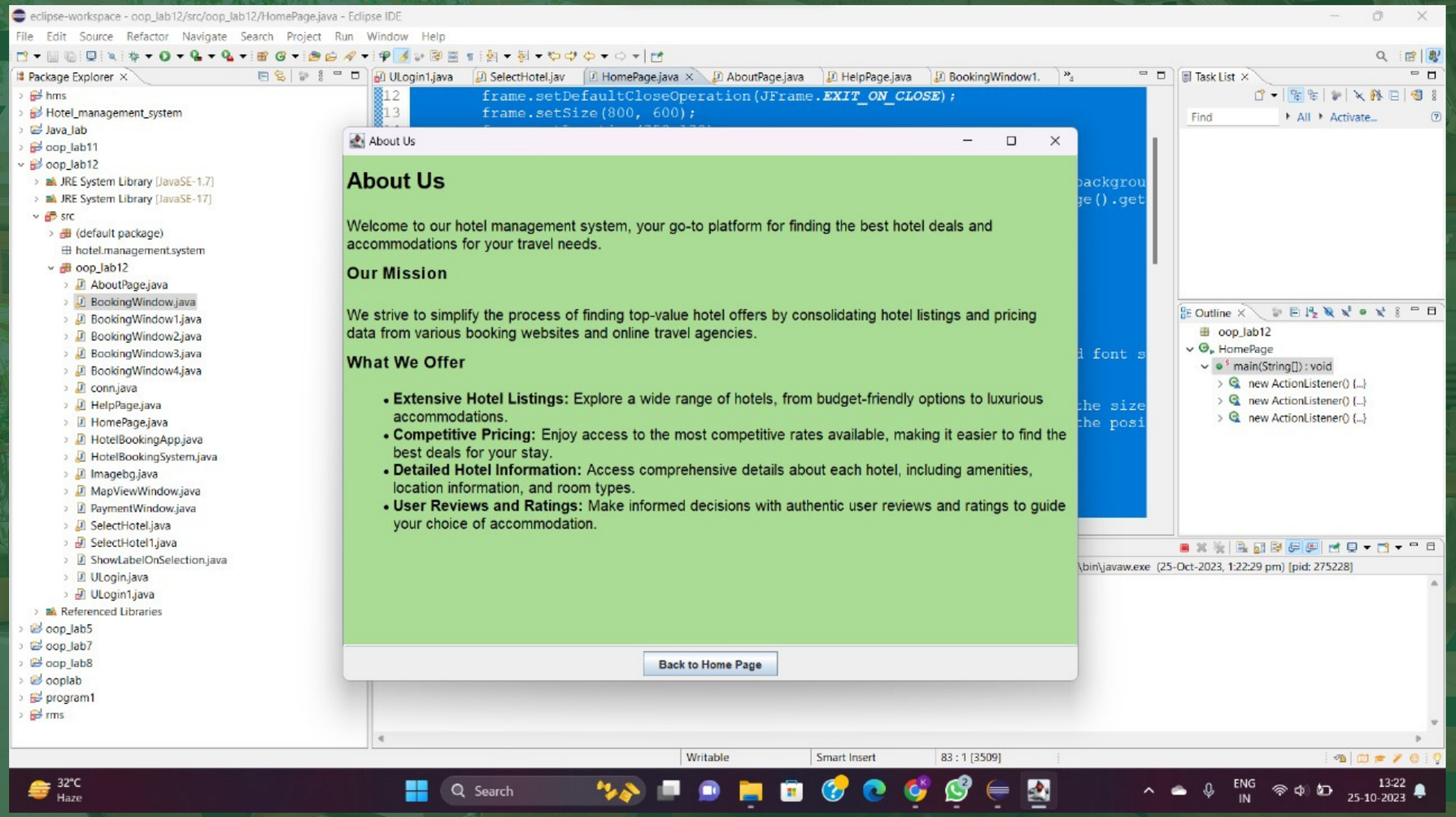
HOME PAGE


eclipse-workspace - oop_lab12/src/oop_lab12/HomePage.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
hms
Hotel_management_system
Java_lab
oop_lab11
oop_lab12
JRE System Library [JavaSE-1.7]
JRE System Library [JavaSE-17]
src
(default package)
hotel.management.system
oop_lab12
AboutPage.java
BookingWindow.java
BookingWindow1.java
BookingWindow2.java
BookingWindow3.java
BookingWindow4.java
conn.java
HelpPage.java
HomePage.java
HotelBookingApp.java
HotelBookingSystem.java
Imagebg.java
MapViewWindow.java
PaymentWindow.java
SelectHotel.java
SelectHotel1.java
ShowLabelOnSelection.java
ULogin.java
ULogin1.java
Referenced Libraries
oop_lab5
oop_lab7
oop_lab8
ooplab
program1
rms
ULogin1.java
SelectHotel.jav
HomePage.java
AboutPage.java
HelpPage.java
BookingWindow1.
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
Home Page
Help
About
Explore Inn
Login
Task List
Find
All
Activate...
Outline
oop_lab12
HomePage
main(String[]) : void
new ActionListener() {...}
new ActionListener() {...}
new ActionListener() {...}
vaw.exe (25-Oct-2023, 1:22:29 pm) [pid: 275228]
Writable
Smart Insert
23 : 32 : 882
32°C
Haze
Search
ENG
IN
13:24
25-10-2023

ABOUT PAGE

- **Swing GUI Components**: The code utilizes Swing, which is Java's standard library for creating graphical user interfaces (GUI). It employs Swing components such as JFrame, JPanel, JTextPane, JScrollPane, and JButton to design the graphical interface.
- **Event Handling**: Event handling is implemented through ActionListener interfaces associated with buttons. For example, when the "Back to Home Page" button is clicked, the actionPerformed method handles the event and initiates the action of returning to the home page.
- **Layout Management**: BorderLayout to arrange components within the panel. It specifies where components should be placed (e.g., BorderLayout.CENTER, BorderLayout.SOUTH) to achieve the desired layout.
- **Method Overriding**: The paintComponent method is overridden in both JPanel and JTextPane to customize their appearance.
- **Frame Creation and Configuration**: Creates a JFrame, configures its properties (such as title, size, and location), and adds a JPanel to it for displaying the user interface.
- C**omponent Hierarchy**: Components like JPanel, JTextPane, and JButton are organized hierarchically within the JFrame.
- **Event-Driven Programming**: User actions (e.g., button clicks) trigger events that are handled by the associated event listeners.

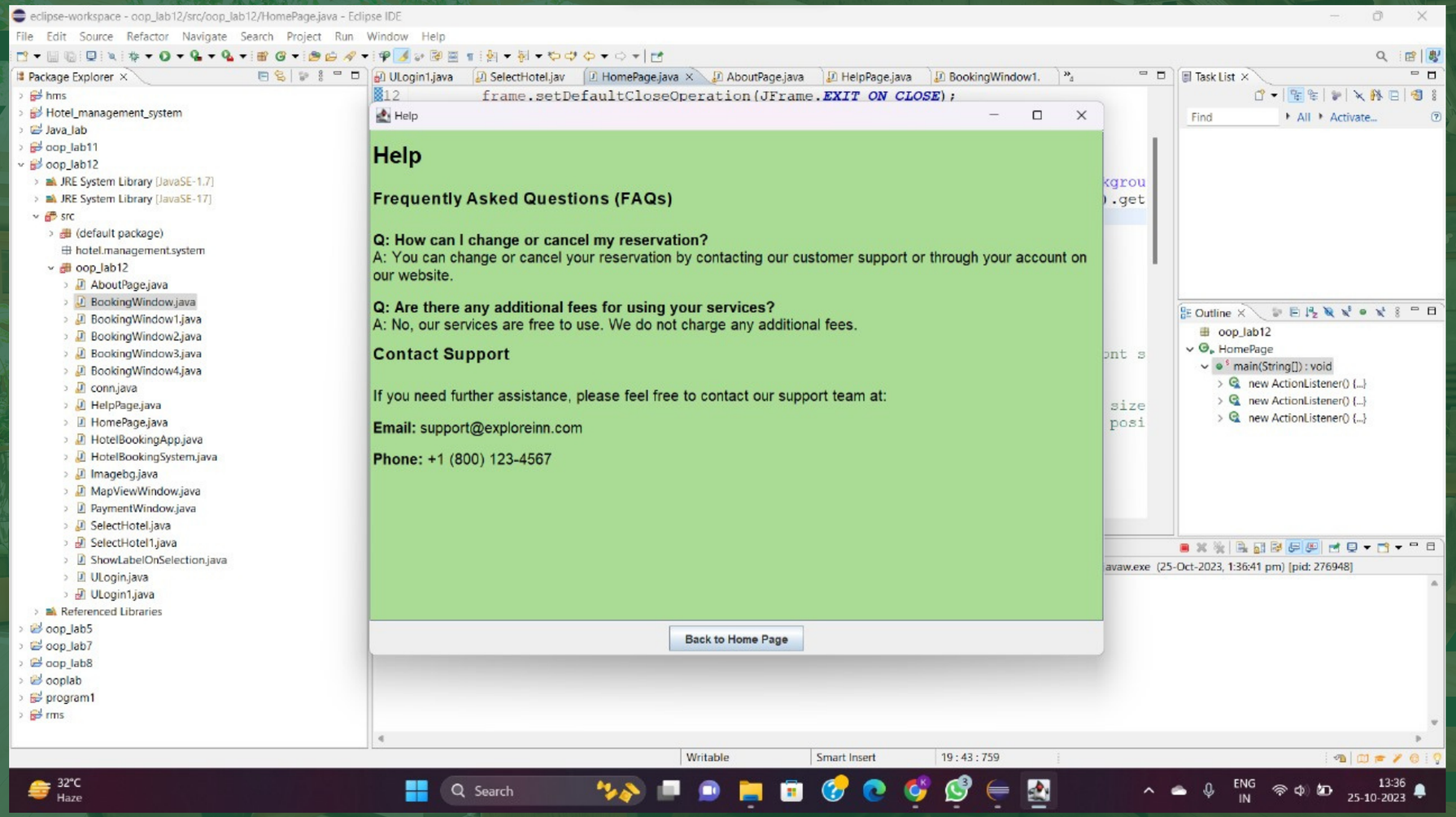
ABOUT PAGE


About Us
About Us
Welcome to our hotel management system, your go-to platform for finding the best hotel deals and accommodations for your travel needs.
Our Mission
We strive to simplify the process of finding top-value hotel offers by consolidating hotel listings and pricing data from various booking websites and online travel agencies.
What We Offer
Extensive Hotel Listings: Explore a wide range of hotels, from budget-friendly options to luxurious accommodations.
Competitive Pricing: Enjoy access to the most competitive rates available, making it easier to find the best deals for your stay.
Detailed Hotel Information: Access comprehensive details about each hotel, including amenities, location information, and room types.
User Reviews and Ratings: Make informed decisions with authentic user reviews and ratings to guide your choice of accommodation.
Back to Home Page

HELP PAGE

- **Swing Components:** Utilizes Swing components like JFrame, JPanel, JTextPane, JScrollPane, and JButton for the user interface.
- **Event Handling:** Implements event handling with ActionListener interfaces for button interactions.
- **Custom Painting:** Customizes the appearance with background colors for JPanel and JTextPane.
- **HTML Content:** Displays richly formatted HTML content in JTextPane.
- **Anonymous Classes:** Employs anonymous inner classes for ActionListener interfaces to handle button events.
- **Method Overriding:** Overrides the paintComponent method for JPanel and JTextPane to achieve custom visual effects.
- **Frame Creation and Configuration:** Creates and configures a JFrame, specifying its title, size, and position.
- **Component Hierarchy:** Organizes components hierarchically within the JFrame for structured interaction.
- **Event-Driven Programming:** Follows an event-driven model where user actions trigger event listeners for responsive interfaces.

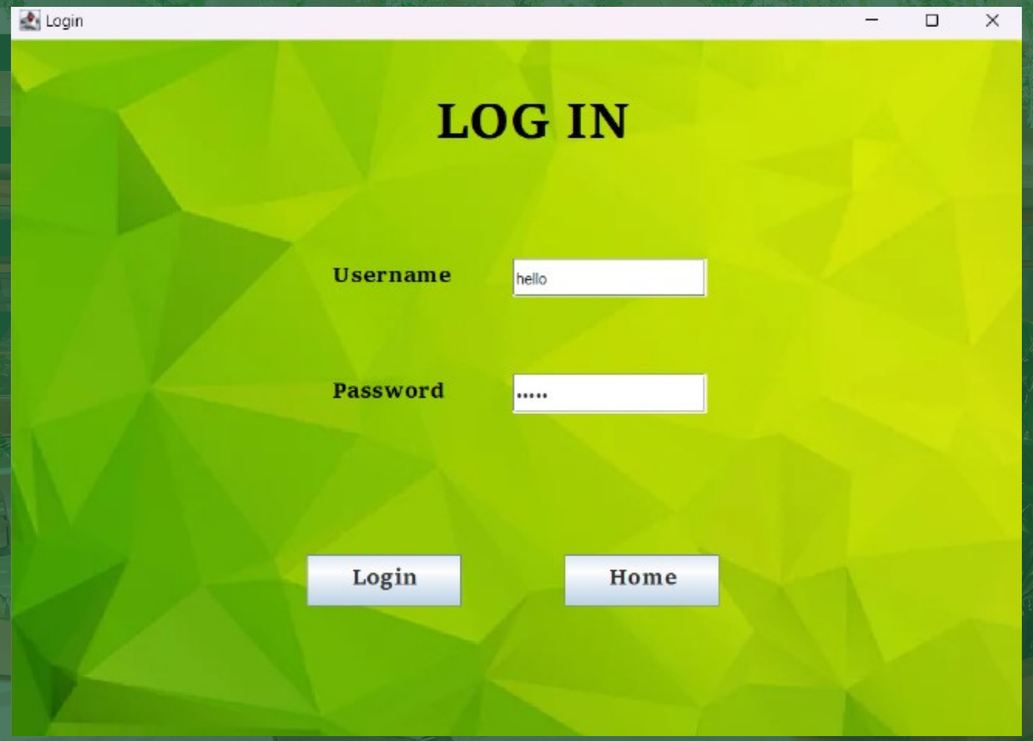
HELP PAGE

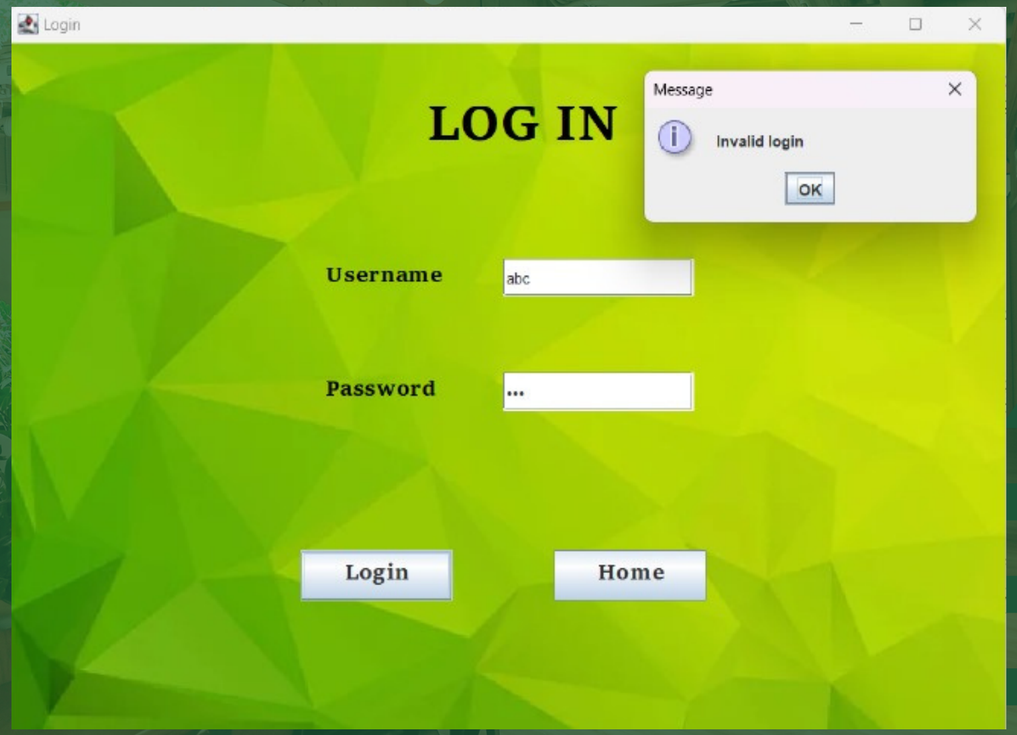

eclipse-workspace - oop_lab12/src/oop_lab12/HomePage.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
hms
Hotel_management_system
Java_lab
oop_lab11
oop_lab12
JRE System Library [JavaSE-1.7]
JRE System Library [JavaSE-17]
src
(default package)
hotel.management.system
oop_lab12
AboutPage.java
BookingWindow.java
BookingWindow1.java
BookingWindow2.java
BookingWindow3.java
BookingWindow4.java
conn.java
HelpPage.java
HomePage.java
HotelBookingApp.java
HotelBookingSystem.java
Imagebg.java
MapViewWindow.java
PaymentWindow.java
SelectHotel.java
SelectHotel1.java
ShowLabelOnSelection.java
ULogin.java
ULogin1.java
Referenced Libraries
oop_lab5
oop_lab7
oop_lab8
ooplab
program1
rms
ULogin1.java
SelectHotel.jav
HomePage.java
AboutPage.java
HelpPage.java
BookingWindow1.
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
Help
Help
Frequently Asked Questions (FAQs)
Q: How can I change or cancel my reservation?
A: You can change or cancel your reservation by contacting our customer support or through your account on our website.
Q: Are there any additional fees for using your services?
A: No, our services are free to use. We do not charge any additional fees.
Contact Support
If you need further assistance, please feel free to contact our support team at:
Email: support@exploreinn.com
Phone: +1 (800) 123-4567
Back to Home Page
Task List
Find
All
Activate...
Outline
oop_lab12
HomePage
main(String[]) : void
new ActionListener() {...}
new ActionListener() {...}
new ActionListener() {...}
avaw.exe (25-Oct-2023, 1:36:41 pm) [pid: 276948]
Writable
Smart Insert
19 : 43 : 759
32°C
Haze
Search
ENG
IN
13:36
25-10-2023

LOGIN PAGE

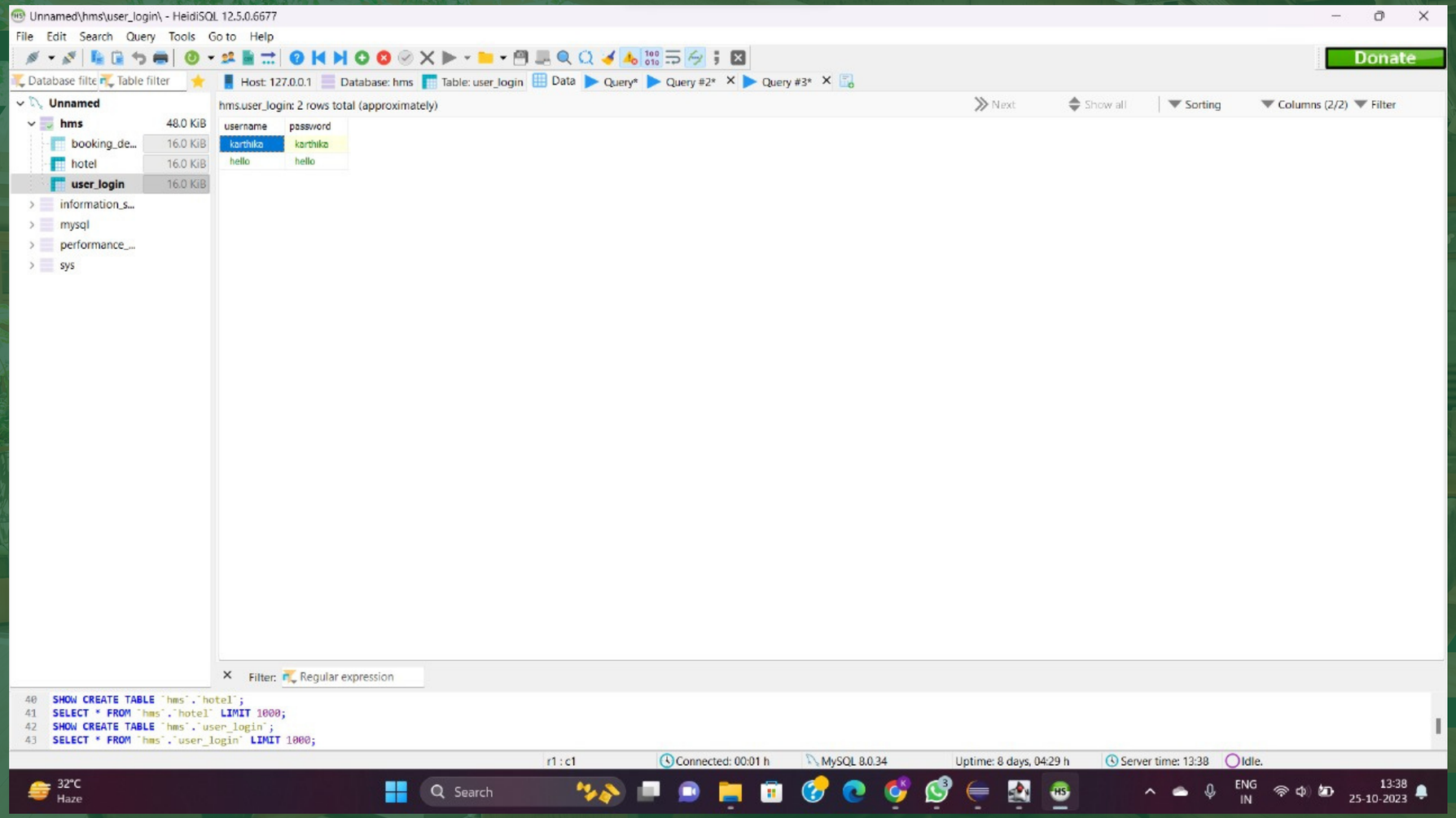
- **Swing GUI Components:** Utilizes Swing components like JLabel, JTextField, JPasswordField, and JButton to create a login window.
- **Event Handling:** Implements event handling with the ActionListener interface for buttons like "Login" and "Home."
- **Database Connectivity:** Connects to a MySQL database using JDBC to verify user login credentials.
- **Custom Painting:** Sets a background image using the ImagePanel class for aesthetic enhancement.
- **Layout Management:** Employs a null layout for precise component positioning.
- **Swing Styling:** Styles Swing components for an aesthetically pleasing user interface.
- **Exception Handling:** Catches and manages errors during database operations and UI interactions which is SQLClassException
- **Multi-Layered Application:** Organizes code into distinct layers for UI, event handling, and database interaction.
- **ActionListener within ActionListener:** Unusual approach to transition to a new window after successful login.
- **Dynamic Window Creation:** Creates and configures JFrame windows for login and post-login interfaces.

LOGIN PAGE


Login
LOG IN
Username
hello
Password
.....
Login
Home


Login
LOG IN
Message
Invalid login
OK
Username
abc
Password
...
Login
Home

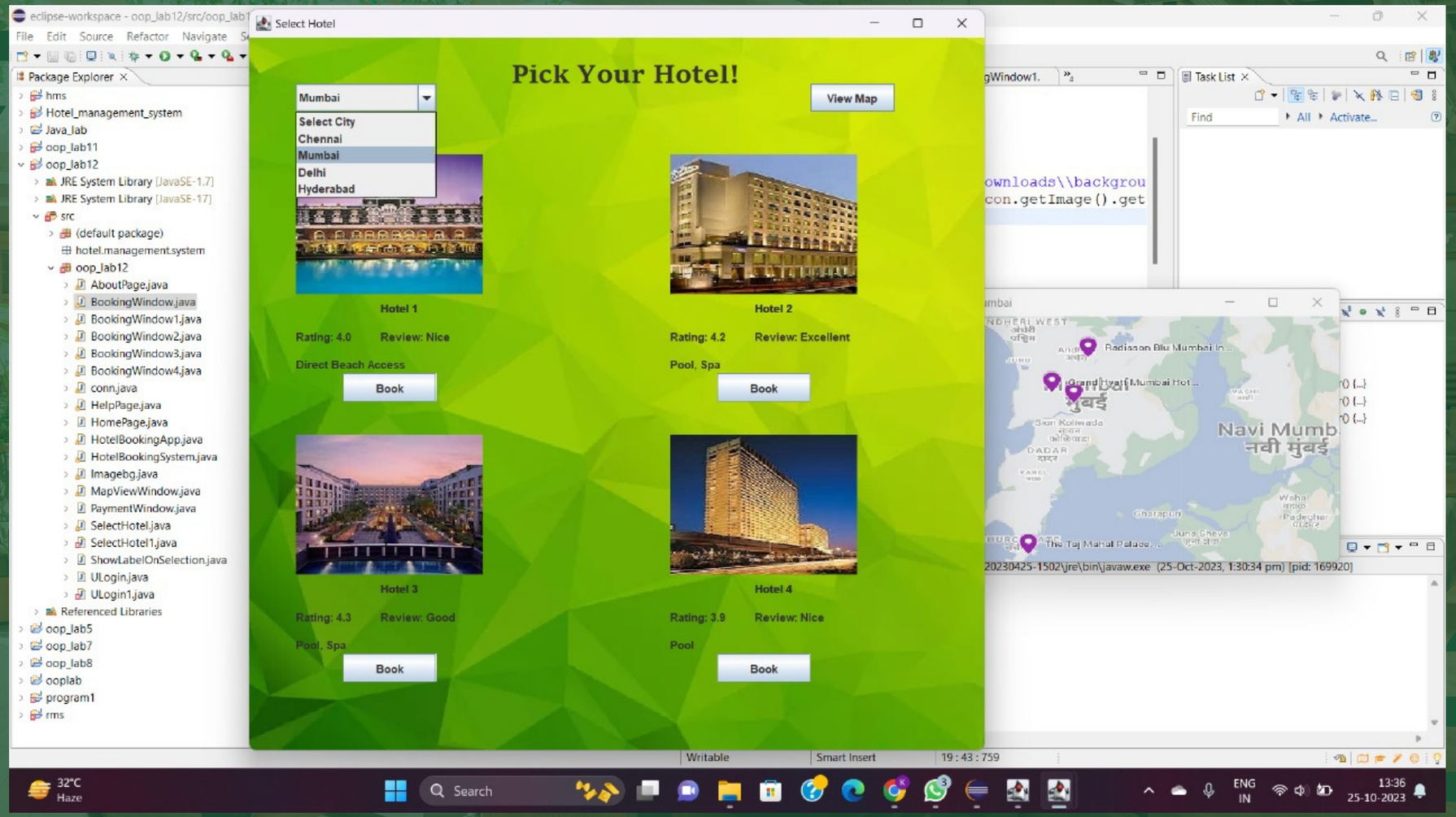
LOGIN PAGE


Unnamed\hms\user_login\ - HeidiSQL 12.5.0.6677
File Edit Search Query Tools Go to Help
Donate
Database filter
Table filter
Host: 127.0.0.1
Database: hms
Table: user_login
Data
Query*
Query #2*
Query #3*
Unnamed
hms 48.0 KiB
booking_de... 16.0 KiB
hotel 16.0 KiB
user_login 16.0 KiB
information_s...
mysql
performance_...
sys
hms.user_login: 2 rows total (approximately)
Next
Show all
Sorting
Columns (2/2)
Filter
username
password
karthika
karthika
hello
hello
Filter:
Regular expression
40 SHOW CREATE TABLE `hms`.`hotel`;
41 SELECT * FROM `hms`.`hotel` LIMIT 1000;
42 SHOW CREATE TABLE `hms`.`user_login`;
43 SELECT * FROM `hms`.`user_login` LIMIT 1000;
r1 : c1
Connected: 00:01 h
MySQL 8.0.34
Uptime: 8 days, 04:29 h
Server time: 13:38
Idle.
32°C
Haze
Search
ENG
IN
13:38
25-10-2023

SELECT HOTEL

- **Swing GUI Components**: Utilizes Swing components like JFrame, JPanel, JLabel, JButton, and JComboBox for hotel selection.
- **Event Handling**: Implements event handling with the ActionListener interface for buttons and combo box selections.
- **Custom Painting**: Overrides paintComponent to set a background image.
- **Arrays and Loops**: Utilizes arrays and loops to display hotel information for different cities.
- **Conditional Statements**: Employs if and else if statements to display information based on the selected city.
- **ActionListeners within ActionListeners**: Handles dynamic hotel booking.
- **Swing Styling**: Styles Swing components by setting fonts, colors, bounds, and text.
- **Window Management**: Opens a new window for viewing city maps, demonstrating window interaction.

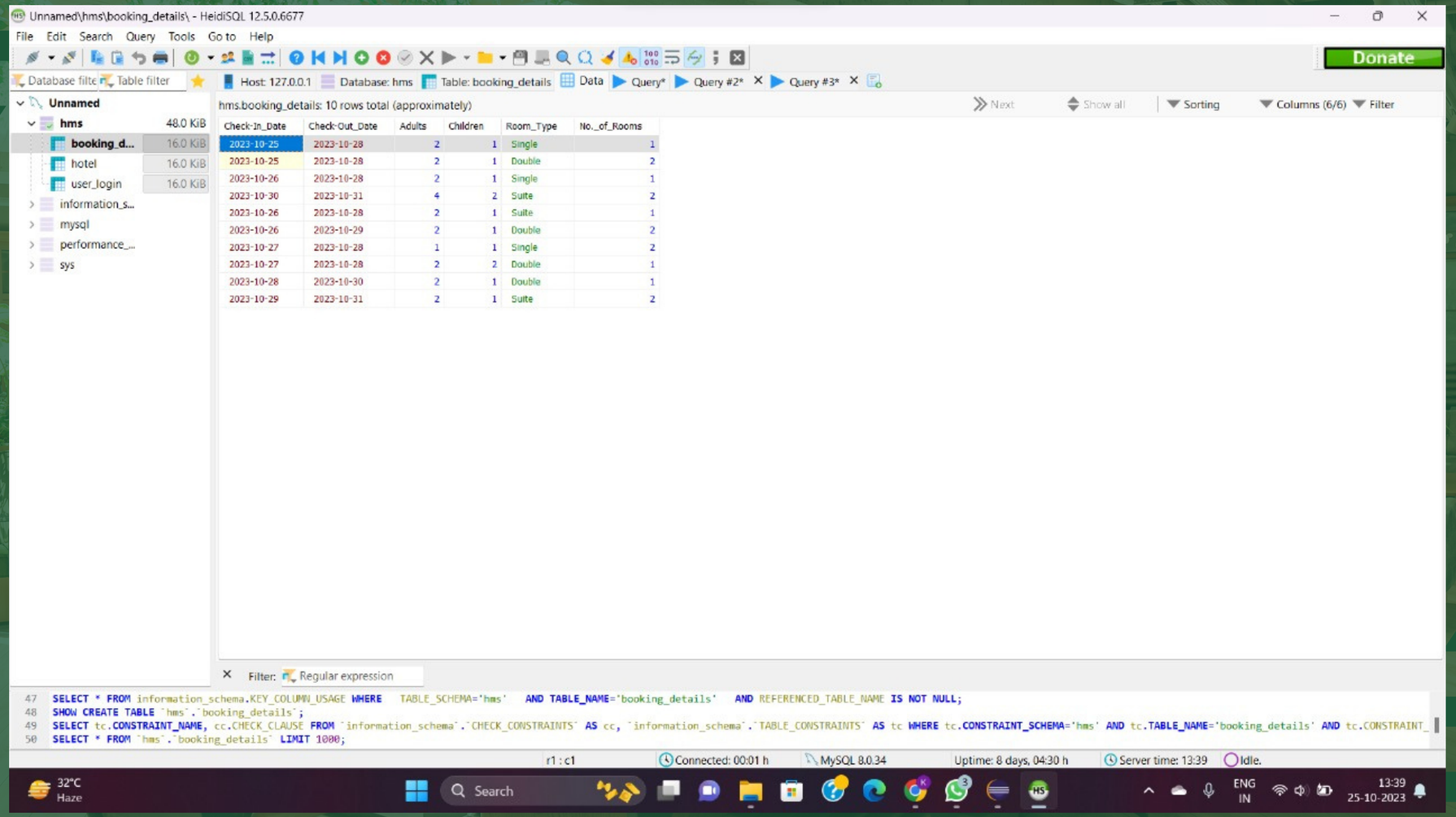
SELECT HOTEL



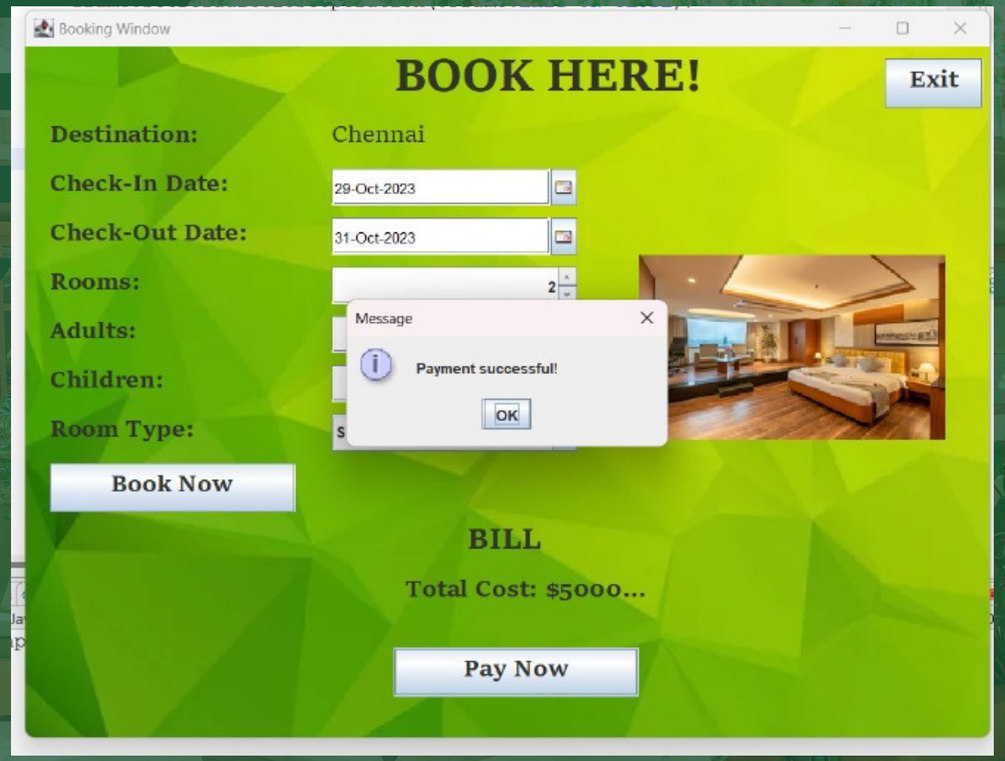
BOOKING WINDOW

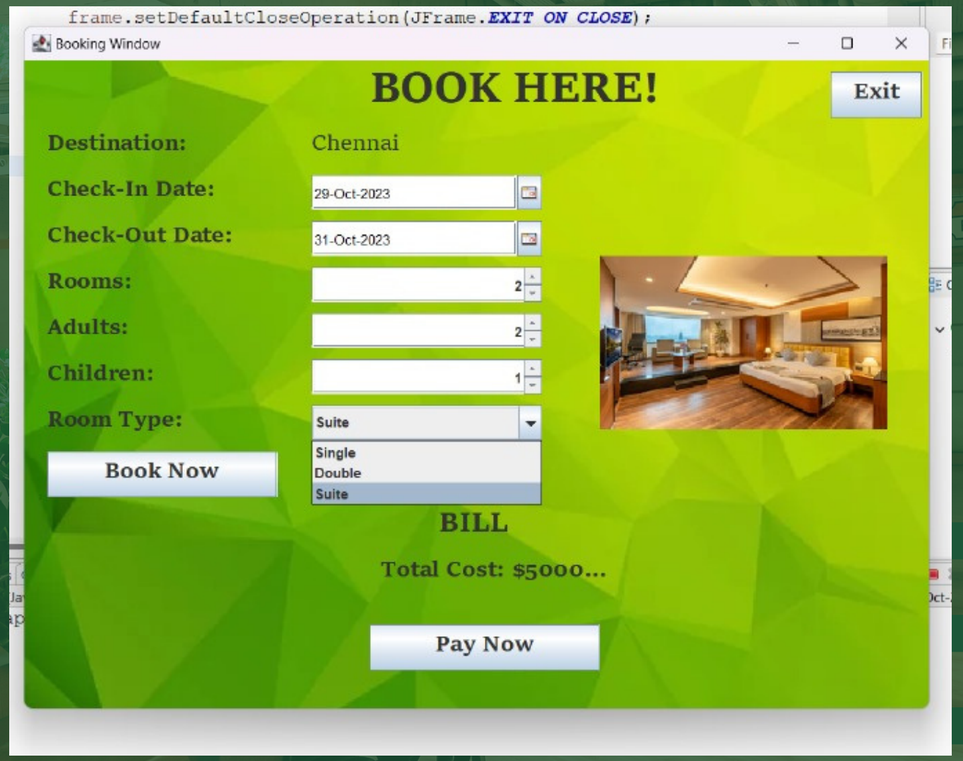
- Swing and AWT: Used for creating GUI elements like buttons, labels, and windows.
- JDBC: Establishes a connection with a MySQL database and handles SQL operations.
- JFrame and Components: Constructs the main application window and adds various GUI components.
- Event Handling: Implements event listeners for user interactions, such as button clicks.
- JDateChooser, JSpinner, JComboBox: Utilized for date selection, quantity selection, and dropdown menus.
- JLabel: Displays text and images within the GUI.
- Exception Handling: Manages potential errors during database interactions using try-catch blocks.
- Date Formatting: Uses SimpleDateFormat to format dates for SQL queries.
- User Feedback: Displays pop-up messages using JOptionPane to communicate application status.
- Button and Window Management: Controls application functionality and window behavior.
- GUI Styling: Applies custom fonts, colors, and styles to the GUI components.

BOOKING WINDOW



BOOKING WINDOW





THANK YOU