

## Java Code and JFrame

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.text.SimpleDateFormat;
import java.text.ParseException;
import java.util.Date;
import javax.swing.border.TitledBorder;
import java.awt.event.ActionListener;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class HotelGUI4
{
    private String url = "jdbc:mysql://localhost:3306/dbhotelmanagement"; // Replace with your
    actual DB URL
    private String user = "your_user"; // Replace with your database username
    private String password = ""; // Replace with your database password, or leave empty if
    none
    private String[] queries = {
        "SELECT * FROM tblguest",
        "SELECT * FROM tblroom",
        "SELECT * FROM tblseasonal",
        "SELECT * FROM tbldestination",
        "SELECT * FROM tblamenities",
        "SELECT * FROM tbladdons",
        "SELECT * FROM tblprice"};

    private JFrame frame;
    private CardLayout cardLayout;
    private JPanel cardPanel;
    private JPanel selectionPanel;
    private JPanel buttonPanel;

    // Customer details
    private int adultGuests, childGuests;
    private String checkInDate = "", checkOutDate = "";
    private String selectedDestination = "None";
    private String selectedRoomType = "Standard";
    private String selectedAddOns = "None";
    private String selectedAmenities = "None";
```

```

//varia
int numAdultGuest = 0;
int numChildGuest = 0;
int numAllGuest = 0;

private Connection connection;

public HotelGUI4() {
    try {
        connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/dbhotelmanagement",
"your_user", "");
        System.out.println("Connected to the database!");
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(frame, "Failed to connect to the database!",
"Database Error", JOptionPane.ERROR_MESSAGE);
    }
    try (Connection connection = DriverManager.getConnection(url, user, password);
        Statement statement = connection.createStatement()) {
        for (String query : queries) {
            try (ResultSet resultSet = statement.executeQuery(query)) {
                // Process results
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

initialize();
}

private void connectToDatabase() {
    try {
        // Load the MySQL JDBC driver
        Class.forName("com.mysql.cj.jdbc.Driver");

        // Establish the connection
        try (Connection connection = DriverManager.getConnection(url, user, password)) {
            System.out.println("Connected to the database!");

            // Execute each query and process the results
            Statement statement = connection.createStatement();
            for (String query : queries) {
                ResultSet resultSet = statement.executeQuery(query);
                System.out.println("Results for query: " + query);
                int columnCount = resultSet.getMetaData().getColumnCount();
            }
        }
    }
}

```

```

        while (resultSet.next()) {
            for (int i = 1; i <= columnCount; i++) {
                System.out.print(resultSet.getString(i) + " ");
            }
            System.out.println();
        }

        resultSet.close(); // Close ResultSet after processing each query
    }

    statement.close();
}

} catch (Exception e) {
    e.printStackTrace();
}
}

```

```

public void initialize() {
    // Main Frame
    frame = new JFrame("Sparking Hotel Booking System");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(800, 500);

    // Card Panel
    cardPanel = new JPanel();
    cardLayout = new CardLayout();
    cardPanel.setLayout(cardLayout);

    // Adding panels
    cardPanel.add(createFrontPagePanel(), "1");
    cardPanel.add(createSignInPanel(), "2");
    cardPanel.add(createDestinationPanel(), "3");
    cardPanel.add(createCheckInAndOutPanel(), "4");
    cardPanel.add(createGuestPanel(), "5");
    cardPanel.add(createRoomSuggestionPanel(), "6");
    cardPanel.add(createAmenitiesPanel(), "7");
    cardPanel.add(createAddonsPanel(), "8");
    cardPanel.add(createConfirmationPanel(), "9");

    frame.add(cardPanel, BorderLayout.CENTER);
    frame.setVisible(true);
}

```

```

// new buttons style

```

```

private JButton createStyledButton(String text)
{
    JButton button = new JButton(text);
    button.setFont(new Font("Serif", Font.BOLD, 18));
    button.setBackground(new Color(70, 130, 180)); // Set background color
    button.setForeground(Color.WHITE);           // Set text color
    return button;
}

private JPanel createFrontPagePanel()
{
    JPanel panel = new JPanel();
    panel.setLayout(new BorderLayout());
    panel.setBackground(new Color(245, 245, 245)); // Light gray background

    // Title Panel
    JPanel titlePanel = new JPanel();
    titlePanel.setBackground(new Color(70, 130, 180)); // Steel blue background
    JLabel title = new JLabel("Welcome to SPARKING HOTEL", SwingConstants.CENTER);
    title.setFont(new Font("Serif", Font.BOLD, 28));
    title.setForeground(Color.WHITE);
    title.setBorder(BorderFactory.createEmptyBorder(20, 10, 20, 10));
    titlePanel.add(title);

    panel.add(titlePanel, BorderLayout.NORTH);

    // Center Panel - Welcome message
    JPanel centerPanel = new JPanel(new BorderLayout());
    centerPanel.setBackground(new Color(245, 245, 245));
    JLabel welcomeMessage = new JLabel(
        "<html><div style='text-align: center;'"
        + "<h2>Plan your perfect getaway</h2>"
        + "<p>Choose from our local and international destinations, luxurious amenities,"
and convenient booking process.</p>"
        + "</div></html>",
        SwingConstants.CENTER
    );
    welcomeMessage.setFont(new Font("Serif", Font.PLAIN, 18));
    welcomeMessage.setBorder(BorderFactory.createEmptyBorder(20, 50, 20, 50));
    centerPanel.add(welcomeMessage, BorderLayout.CENTER);

    panel.add(centerPanel, BorderLayout.CENTER);

    // Navigation Buttons
    JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
    buttonPanel.setBackground(new Color(245, 245, 245));
    JButton nextButton = new JButton("Next");
    nextButton.setFont(new Font("Serif", Font.BOLD, 16));

```

```

        nextButton.setBackground(new Color(70, 130, 180));
        nextButton.setForeground(Color.WHITE);

        nextButton.addActionListener(e -> cardLayout.show(cardPanel, "2")); // Navigate to Sign
In

        buttonPanel.add(nextButton);
        panel.add(buttonPanel, BorderLayout.SOUTH);
        return panel;
    }

    private JPanel createSignInPanel()
    {
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
        panel.setBackground(new Color(245, 245, 245)); // Light gray background

        // Title Section
        JLabel title = new JLabel("User", SwingConstants.CENTER);
        title.setFont(new Font("Serif", Font.BOLD, 28));
        title.setForeground(new Color(70, 130, 180));
        title.setAlignmentX(Component.CENTER_ALIGNMENT);
        title.setBorder(BorderFactory.createEmptyBorder(20, 10, 20, 10));
        panel.add(title);

        panel.add(Box.createRigidArea(new Dimension(0, 20))); // Spacer

        // Form Section
        JPanel formPanel = new JPanel();
        formPanel.setLayout(new BoxLayout(formPanel, BoxLayout.Y_AXIS));
        formPanel.setBackground(new Color(245, 245, 245));
        formPanel.setAlignmentX(Component.CENTER_ALIGNMENT);
        formPanel.setBorder(BorderFactory.createEmptyBorder(20, 50, 20, 50)); // Fixed
padding

        // Username Field
        JPanel usernamePanel = new JPanel(new BorderLayout());
        usernamePanel.setMaximumSize(new Dimension(400, 40)); // Fixed size
        usernamePanel.setBackground(new Color(245, 245, 245));
        JLabel usernameLabel = new JLabel("Username: ");
        usernameLabel.setFont(new Font("Serif", Font.PLAIN, 18));
        JTextField usernameField = new JTextField(15);
        usernameField.setFont(new Font("Serif", Font.PLAIN, 18));

        usernamePanel.add(usernameLabel, BorderLayout.WEST);
        usernamePanel.add(usernameField, BorderLayout.CENTER);

        //Password Field

```

```

JPanel passwordPanel = new JPanel(new BorderLayout());
passwordPanel.setMaximumSize(new Dimension(400, 40)); // Fixed size
passwordPanel.setBackground(new Color(245, 245, 245));
JLabel passwordLabel = new JLabel("Password: ");
passwordLabel.setFont(new Font("Serif", Font.PLAIN, 18));
JPasswordField passwordField = new JPasswordField(15);
passwordField.setFont(new Font("Serif", Font.PLAIN, 18));

passwordPanel.add(passwordLabel, BorderLayout.WEST);
passwordPanel.add(passwordField, BorderLayout.CENTER);

// Add Fields to Form Panel
formPanel.add(usernamePanel);
formPanel.add(Box.createRigidArea(new Dimension(0, 10))); // Spacer
formPanel.add(passwordPanel);

panel.add(formPanel);

panel.add(Box.createRigidArea(new Dimension(0, 20))); // Spacer

// Navigation Buttons
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
buttonPanel.setBackground(new Color(245, 245, 245));

JButton backButton = new JButton("Back");
JButton nextButton = new JButton("Next");

backButton.setFont(new Font("Serif", Font.BOLD, 16));
nextButton.setFont(new Font("Serif", Font.BOLD, 16));
backButton.setBackground(new Color(100, 100, 100));
nextButton.setBackground(new Color(70, 130, 180));
backButton.setForeground(Color.WHITE);
nextButton.setForeground(Color.WHITE);

backButton.addActionListener(e -> cardLayout.show(cardPanel, "1")); // Back to Front
Page
nextButton.addActionListener(e -> cardLayout.show(cardPanel, "3")); // Go to Guest Info
Panel

buttonPanel.add(backButton);
buttonPanel.add(nextButton);
panel.add(buttonPanel, BorderLayout.SOUTH);

return panel;
}

// New Destination Panel with Local and International Buttons
private JPanel createDestinationPanel() {

```

```

// Main panel setup
JPanel panel = new JPanel(new BorderLayout());

// Create selectionPanel to hold the dynamically changing content
selectionPanel = new JPanel();
selectionPanel.setLayout(new BoxLayout(selectionPanel, BoxLayout.Y_AXIS));
selectionPanel.setBackground(new Color(245, 245, 245));

// Create Local and International buttons
JButton localButton = new JButton("Local");
JButton internationalButton = new JButton("International");

// Create a button panel to center the buttons
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new BoxLayout(buttonPanel, BoxLayout.Y_AXIS)); // Stack
vertically
buttonPanel.setBackground(new Color(245, 245, 245));
buttonPanel.add(Box.createVerticalGlue()); // Push buttons to the center vertically
buttonPanel.add(localButton);
buttonPanel.add(Box.createRigidArea(new Dimension(0, 10))); // Spacer between
buttons
buttonPanel.add(internationalButton);
buttonPanel.add(Box.createVerticalGlue()); // Push buttons to the center vertically

// Center-align the buttons
localButton.setAlignmentX(Component.CENTER_ALIGNMENT);
internationalButton.setAlignmentX(Component.CENTER_ALIGNMENT);

// Add action listeners for the buttons
localButton.addActionListener(e -> displayLocalDestinations());
internationalButton.addActionListener(e -> displayInternationalDestinations());

// Create a container to hold both buttons and dynamic destinations
JPanel container = new JPanel(new BorderLayout());
container.setBorder(BorderFactory.createTitledBorder(
    BorderFactory.createLineBorder(Color.GRAY, 2), "Destination Selection"));
// Add button panel with customized padding
buttonPanel.setBorder(BorderFactory.createEmptyBorder(20, 10, 40, 10)); // Reduce
bottom padding for closer spacing
container.add(buttonPanel, BorderLayout.NORTH); // Place buttons at the top

// Add selection panel with customized padding
selectionPanel.setBorder(BorderFactory.createEmptyBorder(30, 10, 10, 10)); // Reduce
top padding for closer spacing
container.add(selectionPanel, BorderLayout.CENTER); // Place selections below buttons
panel.add(container, BorderLayout.CENTER);

// Navigation buttons panel

```

```

JPanel navigationPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
navigationPanel.setBackground(new Color(245, 245, 245));

JButton backButton = createStyledButton("Back");
JButton nextButton = createStyledButton("Next");

backButton.setFont(new Font("Serif", Font.BOLD, 16));
nextButton.setFont(new Font("Serif", Font.BOLD, 16));
backButton.setBackground(new Color(100, 100, 100)); // Dark gray background for Back
button
nextButton.setBackground(new Color(70, 130, 180)); // Steel blue background for Next
button
backButton.setForeground(Color.WHITE); // White text on the Back button
nextButton.setForeground(Color.WHITE); // White text on the Next button

// Back button resets the panel
backButton.addActionListener(e -> resetToInitialView(buttonPanel, selectionPanel));
backButton.addActionListener(e -> cardLayout.show(cardPanel, "2"));

// Next button navigates forward
nextButton.addActionListener(
    e -> {
        if (validateDestinationSelection()) {
            cardLayout.show(cardPanel, "4"); // Navigate to Guest Information Panel
        } else {
            JOptionPane.showMessageDialog(frame, "Please select a destination before
proceeding.",
                "Validation Error", JOptionPane.ERROR_MESSAGE);
        }
    });

navigationPanel.add(backButton);
navigationPanel.add(nextButton);

panel.add(navigationPanel, BorderLayout.SOUTH);

return panel;
}

// Method to reset the panel to the initial state
private void resetToInitialView(JPanel buttonPanel, JPanel selectionPanel) {
    selectionPanel.removeAll(); // Clear any displayed options
    selectionPanel.revalidate();
    selectionPanel.repaint();

    // Ensure only the Local and International buttons are visible
    buttonPanel.setVisible(true);
}

```



```

// Method to validate the destination selection
private boolean validateDestinationSelection() {
    // Check if any of the radio buttons in the selection panel are selected
    for (Component comp : selectionPanel.getComponents()) {
        if (comp instanceof JRadioButton) {
            JRadioButton radioButton = (JRadioButton) comp;
            if (radioButton.isSelected()) {
                return true; // A destination has been selected
            }
        }
    }
    return false; // No selection made
}

// Method to display local destinations
private void displayLocalDestinations() {
    selectionPanel.removeAll(); // Remove previous content

    // Create radio buttons for local destinations
    JRadioButton baguioButton = new JRadioButton("Baguio");
    JRadioButton boracayButton = new JRadioButton("Boracay");
    JRadioButton elNidoButton = new JRadioButton("El Nido");
    JRadioButton siargaoButton = new JRadioButton("Siargao");

    ButtonGroup localGroup = new ButtonGroup();
    localGroup.add(baguioButton);
    localGroup.add(boracayButton);
    localGroup.add(elNidoButton);
    localGroup.add(siargaoButton);

    // Add buttons to selectionPanel
    selectionPanel.add(baguioButton);
    selectionPanel.add(Box.createRigidArea(new Dimension(0, 5)));
    selectionPanel.add(boracayButton);
    selectionPanel.add(Box.createRigidArea(new Dimension(0, 5)));
    selectionPanel.add(elNidoButton);
    selectionPanel.add(Box.createRigidArea(new Dimension(0, 5)));
    selectionPanel.add(siargaoButton);

    ActionListener updateDestination = e -> selectedDestination = ((JRadioButton)
e.getSource()).getText();
    baguioButton.addActionListener(updateDestination);
    boracayButton.addActionListener(updateDestination);
    elNidoButton.addActionListener(updateDestination);
    siargaoButton.addActionListener(updateDestination);

    // Align buttons

```

```

        centerAlignComponents(selectionPanel);

        selectionPanel.revalidate();
        selectionPanel.repaint();
    }

// Method to display international destinations
private void displayInternationalDestinations() {
    selectionPanel.removeAll(); // Remove previous content

    // Create radio buttons for international destinations
    JRadioButton hongKongButton = new JRadioButton("Hong Kong");
    JRadioButton japanButton = new JRadioButton("Japan");
    JRadioButton singaporeButton = new JRadioButton("Singapore");
    JRadioButton southKoreaButton = new JRadioButton("South Korea");

    ButtonGroup internationalGroup = new ButtonGroup();
    internationalGroup.add(hongKongButton);
    internationalGroup.add(japanButton);
    internationalGroup.add(singaporeButton);
    internationalGroup.add(southKoreaButton);

    // Add buttons to selectionPanel
    selectionPanel.add(hongKongButton);
    selectionPanel.add(Box.createRigidArea(new Dimension(0, 5)));
    selectionPanel.add(japanButton);
    selectionPanel.add(Box.createRigidArea(new Dimension(0, 5)));
    selectionPanel.add(singaporeButton);
    selectionPanel.add(Box.createRigidArea(new Dimension(0, 5)));
    selectionPanel.add(southKoreaButton);

    ActionListener updateDestination = e -> selectedDestination = ((JRadioButton)
e.getSource()).getText();
    hongKongButton.addActionListener(updateDestination);
    japanButton.addActionListener(updateDestination);
    singaporeButton.addActionListener(updateDestination);
    southKoreaButton.addActionListener(updateDestination);

    // Align buttons
    centerAlignComponents(selectionPanel);

    selectionPanel.revalidate();
    selectionPanel.repaint();
}

// Helper method to center-align components
private void centerAlignComponents(JPanel panel) {

```

```

    for (Component comp : panel.getComponents()) {
        if (comp instanceof JRadioButton) {
            ((JRadioButton) comp).setAlignmentX(Component.CENTER_ALIGNMENT);
        }
    }
}

private JPanel createCheckInAndOutPanel() {
    JPanel panel = new JPanel();
    panel.setLayout(new BorderLayout());
    panel.setBackground(new Color(245, 245, 245)); // Light gray background

    // Title
    JLabel title = new JLabel("Check-In and Check-Out Information",
        SwingConstants.CENTER);
    title.setFont(new Font("Serif", Font.BOLD, 22));
    title.setForeground(new Color(70, 130, 180));
    title.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
    panel.add(title, BorderLayout.NORTH);

    // Details Panel
    JPanel detailsPanel = new JPanel();
    detailsPanel.setLayout(new GridBagLayout()); // Align components
    detailsPanel.setBorder(BorderFactory.createTitledBorder("Booking Details"));
    detailsPanel.setBackground(new Color(245, 245, 245));

    GridBagConstraints gbc = new GridBagConstraints();
    gbc.insets = new Insets(5, 10, 5, 10); // Spacing
    gbc.anchor = GridBagConstraints.WEST;

    // Check-in Date
    gbc.gridx = 0;
    gbc.gridy = 0;
    detailsPanel.add(new JLabel("Check-in Date (MM-DD-YYYY): "), gbc);

    JTextField checkInField = new JTextField(10);
    checkInField.setHorizontalAlignment(SwingConstants.RIGHT); // Right-align text inside
the text field
    gbc.gridx = 1;
    gbc.anchor = GridBagConstraints.EAST;
    detailsPanel.add(checkInField, gbc);

    // Check-out Date
    gbc.gridx = 0;
    gbc.gridy = 1;
    gbc.anchor = GridBagConstraints.WEST;
    detailsPanel.add(new JLabel("Check-out Date (MM-DD-YYYY): "), gbc);

```

```

    JTextField checkOutField = new JTextField(10);
    checkOutField.setHorizontalAlignment(SwingConstants.RIGHT); // Right-align text inside
the text field
    gbc.gridx = 1;
    gbc.anchor = GridBagConstraints.EAST;
    detailsPanel.add(checkOutField, gbc);

// Center the Details Panel
    JPanel centerPanel = new JPanel(new GridBagLayout()); // Center alignment
    centerPanel.setBackground(new Color(245, 245, 245));
    centerPanel.add(detailsPanel);

    panel.add(centerPanel, BorderLayout.CENTER);

// Navigation Buttons
    JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
    JButton backButton = new JButton("Back");
    JButton nextButton = new JButton("Next");
    backButton.setFont(new Font("Serif", Font.BOLD, 16));
    nextButton.setFont(new Font("Serif", Font.BOLD, 16));
    backButton.setBackground(new Color(100, 100, 100));
    nextButton.setBackground(new Color(70, 130, 180));
    backButton.setForeground(Color.WHITE);
    nextButton.setForeground(Color.WHITE);

// Back button action
    backButton.addActionListener(e -> cardLayout.show(cardPanel, "3")); // Replace "1" with
the previous panel key

// Next button action with validation
    nextButton.addActionListener(
        e -> {
            String checkIn = checkInField.getText().trim();
            String checkOut = checkOutField.getText().trim();

// Validate dates
            try {
                SimpleDateFormat dateFormat = new SimpleDateFormat("MM-dd-yyyy");
                dateFormat.setLenient(false);

                Date currentDate = new Date();
                Date checkInDate = dateFormat.parse(checkIn);
                Date checkOutDate = dateFormat.parse(checkOut);

```

```

        if (checkInDate.before(currentDate)) {
            JOptionPane.showMessageDialog(panel, "Check-in date cannot be in the past.",
"Validation Error", JOptionPane.ERROR_MESSAGE);
            return;
        }

        if (!checkOutDate.after(checkInDate)) {
            JOptionPane.showMessageDialog(panel, "Check-out date must be after the
check-in date.", "Validation Error", JOptionPane.ERROR_MESSAGE);
            return;
        }
    } catch (ParseException ex) {
        JOptionPane.showMessageDialog(panel, "Please input valid dates in the format
MM-DD-YYYY.", "Validation Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    // Validation passed
    checkInDate = checkIn;
    checkOutDate = checkOut;

    // Navigate to the next panel
    cardLayout.show(cardPanel, "5"); // Replace "2" with the destination panel key
});

buttonPanel.add(backButton);
buttonPanel.add(nextButton);

panel.add(buttonPanel, BorderLayout.SOUTH);

return panel;
}

```

```

// Guest Number
private JPanel createGuestPanel() {
    JPanel panel = new JPanel();
    panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
    panel.setBackground(new Color(245, 245, 245));
}

```

```

// Title
JLabel title = new JLabel("Guest Information", SwingConstants.CENTER);
title.setFont(new Font("Serif", Font.BOLD, 24));
title.setForeground(new Color(70, 130, 180));
title.setAlignmentX(Component.CENTER_ALIGNMENT);
title.setBorder(BorderFactory.createEmptyBorder(20, 10, 20, 10));

```

```

panel.add(title);

panel.add(Box.createRigidArea(new Dimension(0, 20))); // Spacer

// Guest Info Panel
JPanel guestInfoPanel = new JPanel(new GridBagLayout());
guestInfoPanel.setBackground(new Color(245, 245, 245));
guestInfoPanel.setBorder(BorderFactory.createTitledBorder(
    BorderFactory.createLineBorder(Color.GRAY, 2),
    "Enter Guest Details",
    TitledBorder.LEFT,
    TitledBorder.TOP,
    new Font("Serif", Font.BOLD, 18)
));

GridBagConstraints gbc = new GridBagConstraints();
gbc.insets = new Insets(10, 10, 10, 10);
gbc.fill = GridBagConstraints.HORIZONTAL;

// Number of Adults
JLabel adultLabel = new JLabel("Number of Adults:");
adultLabel.setFont(new Font("Serif", Font.PLAIN, 16));
gbc.gridx = 0;
gbc.gridy = 0;
guestInfoPanel.add(adultLabel, gbc);

JTextField adultField = new JTextField(5);
gbc.gridx = 1;
gbc.gridy = 0;
adultField.setHorizontalAlignment(SwingConstants.RIGHT);
guestInfoPanel.add(adultField, gbc);

// Number of Children
JLabel childLabel = new JLabel("Number of Children:");
childLabel.setFont(new Font("Serif", Font.PLAIN, 16));
gbc.gridx = 0;
gbc.gridy = 1;
guestInfoPanel.add(childLabel, gbc);

JTextField childField = new JTextField(5);
gbc.gridx = 1;
gbc.gridy = 1;
childField.setHorizontalAlignment(SwingConstants.RIGHT);
guestInfoPanel.add(childField, gbc);

panel.add(guestInfoPanel);
panel.add(Box.createRigidArea(new Dimension(0, 20))); // Spacer

```

```

// Navigation Buttons
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
buttonPanel.setBackground(new Color(245, 245, 245));

JButton backButton = new JButton("Back");
JButton nextButton = new JButton("Next");

backButton.setFont(new Font("Serif", Font.BOLD, 18));
nextButton.setFont(new Font("Serif", Font.BOLD, 18));
backButton.setBackground(new Color(100, 100, 100));
nextButton.setBackground(new Color(70, 130, 180));
backButton.setForeground(Color.WHITE);
nextButton.setForeground(Color.WHITE);

backButton.addActionListener(e -> cardLayout.show(cardPanel, "4")); // Back to Sign-In
Panel

nextButton.addActionListener(
    e -> {
        try {

            //vars
            numAdultGuest = Integer.parseInt(adultField.getText().trim());
            numChildGuest = Integer.parseInt(childField.getText().trim());
            numAllGuest = numAdultGuest + numChildGuest;

            // Parse the input values
            int numAdultGuest = Integer.parseInt(adultField.getText().trim());
            int children = Integer.parseInt(childField.getText().trim());

            // Validation: Number of adults must be greater than 0
            if (numAdultGuest == 0) {
                JOptionPane.showMessageDialog(panel, "The number of adults must be at least
1.", "Validation Error", JOptionPane.ERROR_MESSAGE);
                return; // Stop further execution
            }

            // If children is 0, proceed directly
            if (children == 0) {
                cardLayout.show(cardPanel, "6"); // Proceed to the next panel
            } else if (children > 0) {
                // Prompt for children's ages
                boolean agesConfirmed = askForChildrenAges(children);
                if (agesConfirmed) {
                    cardLayout.show(cardPanel, "6"); // Proceed only if user confirms
                } else {
                    System.out.println("Child age input canceled. Staying on the current page.");
                }
            }
        }
    }
);

```

```

    }
    } catch (NumberFormatException ex) {
        // Handle invalid input
        JOptionPane.showMessageDialog(panel, "Please enter valid numbers for adults
and children.", "Invalid Input", JOptionPane.ERROR_MESSAGE);
    }
});

```

```

buttonPanel.add(backButton);
buttonPanel.add(nextButton);
panel.add(buttonPanel);

```

```

return panel;
}

```

```

private boolean askForChildrenAges(int numChildGuest) {
    JPanel agePanel = new JPanel();
    agePanel.setLayout(new BoxLayout(agePanel, BoxLayout.Y_AXIS));
    agePanel.setBackground(new Color(245, 245, 245));

    JLabel instructionLabel = new JLabel("Please enter the age of each child (0-7):");
    instructionLabel.setFont(new Font("Serif", Font.PLAIN, 16));
    agePanel.add(instructionLabel);

    JTextField[] ageFields = new JTextField[numChildGuest];
    for (int i = 0; i < numChildGuest; i++) {
        JPanel rowPanel = new JPanel(new FlowLayout(FlowLayout.LEFT));
        rowPanel.setBackground(new Color(245, 245, 245));

        JLabel ageLabel = new JLabel("Child " + (i + 1) + " Age:");
        ageLabel.setFont(new Font("Serif", Font.PLAIN, 16));

        rowPanel.add(ageLabel);

        ageFields[i] = new JTextField(5);

        rowPanel.add(ageFields[i]);

        agePanel.add(rowPanel);
    }
}

```

```

int result = JOptionPane.showConfirmDialog(
    null,
    agePanel,
    "Child Ages",

```



```

        JOptionPane.OK_CANCEL_OPTION,
        JOptionPane.PLAIN_MESSAGE
    );

    if (result == JOptionPane.OK_OPTION) {
        try {
            for (int i = 0; i < numChildGuest; i++) {
                int age = Integer.parseInt(ageFields[i].getText());

                if (age < 0)
                {
                    JOptionPane.showMessageDialog(null, "Invalid age. Please enter a non-negative
number.");
                    return askForChildrenAges(numChildGuest);
                }
                else if (age >= 8)
                {
                    numAllGuest++;
                }
                else if (age > 7)
                {
                    JOptionPane.showMessageDialog(null, "Child age must be between 0 and 7.");
                    return askForChildrenAges(numChildGuest);
                }

                // Store or process valid age
                System.out.println("Child " + (i + 1) + " Age: " + age);
            }
            return true;
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(null, "Invalid input. Please enter numbers only.");
            return askForChildrenAges(numChildGuest);
        }
    }

    return false; // Cancel or close the dialog
}

```

```

private JPanel createRoomSuggestionPanel() {
    JPanel panel = new JPanel();
    panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
    panel.setBackground(new Color(245, 245, 245)); // Light gray background
    panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20)); // Padding around
the panel

```

```

// Title
JLabel title = new JLabel("Room Suggestions", SwingConstants.CENTER);

```

```

title.setFont(new Font("Serif", Font.BOLD, 20)); // Adjusted font size
title.setAlignmentX(Component.CENTER_ALIGNMENT);
title.setBorder(BorderFactory.createEmptyBorder(10, 0, 20, 0));
panel.add(title);

panel.add(Box.createRigidArea(new Dimension(0, 20))); // Spacer

//text
JLabel text = new JLabel("Suggested Room type is Double", SwingConstants.LEFT);
title.setFont(new Font("Serif", Font.BOLD, 20)); // Adjusted font size
title.setAlignmentX(Component.LEFT_ALIGNMENT);
title.setBorder(BorderFactory.createEmptyBorder(10, 0, 20, 0));
panel.add(text);

panel.add(Box.createRigidArea(new Dimension(0, 20))); // Spacer

// Button Group for Room Selection
ButtonGroup roomGroup = new ButtonGroup();

// Room Types and Capacities
String[][] roomDetails = {
    {"Standard", "Capacity: 1 guest with 1 extra bed"},
    {"Deluxe", "Capacity: 2 guests with 1 extra bed"},
    {"Quadruple", "Capacity: 4 guests with 1 extra bed"},
    {"Family", "Capacity: 6 guests with 1 extra bed"},
    {"Suite", "Capacity: 4 guests"}
};

for (String[] room : roomDetails) {
    JPanel roomPanel = new JPanel();
    roomPanel.setLayout(new BorderLayout());
    roomPanel.setBackground(new Color(245, 245, 245));
    roomPanel.setBorder(
        BorderFactory.createCompoundBorder(
            BorderFactory.createTitledBorder(room[0]), // Room type as border title
            BorderFactory.createEmptyBorder(5, 10, 5, 10) // Padding within the border
        )
    );

    JRadioButton radioButton = new JRadioButton(room[1]);
    radioButton.setFont(new Font("Serif", Font.PLAIN, 14)); // Adjusted font size for radio
    button text
    radioButton.setBackground(new Color(245, 245, 245));
    roomGroup.add(radioButton);

    roomPanel.add(radioButton, BorderLayout.WEST);
}

```

```

        panel.add(roomPanel);
        panel.add(Box.createRigidArea(new Dimension(0, 10))); // Spacer between room
panels
    }

    panel.add(Box.createRigidArea(new Dimension(0, 20))); // Spacer before buttons

// Navigation Buttons
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
buttonPanel.setBackground(new Color(245, 245, 245));

JButton backButton = new JButton("Back");
JButton nextButton = new JButton("Next");

backButton.setFont(new Font("Serif", Font.BOLD, 14)); // Adjusted button font
nextButton.setFont(new Font("Serif", Font.BOLD, 14)); // Adjusted button font
backButton.setBackground(new Color(100, 100, 100));
nextButton.setBackground(new Color(70, 130, 180));
backButton.setForeground(Color.WHITE);
nextButton.setForeground(Color.WHITE);

// Action Listeners for navigation
backButton.addActionListener(e -> cardLayout.show(cardPanel, "5")); // Back to
Destination Panel
nextButton.addActionListener(
    e -> {
        // Validation to ensure a room is selected
        if (roomGroup.getSelection() == null) {
            JOptionPane.showMessageDialog(panel, "Please select a room type before
proceeding.", "Validation Error", JOptionPane.ERROR_MESSAGE);
            return;
        }

        // If validation passes, navigate to the next panel
        cardLayout.show(cardPanel, "7"); // Go to Amenities Panel
    });

buttonPanel.add(backButton);
buttonPanel.add(nextButton);
panel.add(buttonPanel, BorderLayout.SOUTH);

return panel;
}

private JPanel createAmenitiesPanel() {
    JPanel panel = new JPanel();
    panel.setLayout(new BorderLayout());

```

```

panel.setBackground(new Color(245, 245, 245)); // Light gray background
panel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2)); // Border around the
panel
panel.setPreferredSize(new Dimension(500, 400)); // Fixed size

// Title
JLabel title = new JLabel("Select Amenities", SwingConstants.CENTER);
title.setFont(new Font("Serif", Font.BOLD, 24));
title.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
panel.add(title, BorderLayout.NORTH);
title.setForeground(new Color(70, 130, 180));
title.setAlignmentX(Component.CENTER_ALIGNMENT);

// Add-ons Panel with GridBagLayout
JPanel addonsPanel = new JPanel();
addonsPanel.setLayout(new GridBagLayout());
addonsPanel.setBorder(BorderFactory.createTitledBorder(
    BorderFactory.createLineBorder(Color.GRAY, 2), "Available Amenities"));
addonsPanel.setBackground(new Color(245, 245, 245));

GridBagConstraints gbc = new GridBagConstraints();
gbc.insets = new Insets(10, 10, 10, 10); // Padding around components
gbc.fill = GridBagConstraints.HORIZONTAL; // Ensure components fill the grid cells

// Add-ons with Quantities
JLabel thaiMassageLabel = new JLabel("Thai Massage (₱1540):");
JLabel aromaFacialLabel = new JLabel("Aroma Facial Massage (₱1045):");
JLabel poolAccessLabel = new JLabel("Access to Pool (₱300):");
JLabel gymAccessLabel = new JLabel("Gym Access (₱500):");
JLabel footSpaLabel = new JLabel("Foot Spa Services (₱825):");

JSpinner thaiMassageSpinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1)); //
Quantity for Thai Massage
JSpinner aromaFacialSpinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1)); //
Quantity for Aroma Facial
JSpinner poolAccessSpinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1)); //
Quantity for Pool Access
JSpinner gymAccessSpinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1)); //
Quantity for Gym Access
JSpinner footSpaSpinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1)); //
Quantity for Foot Spa

JLabel thaiMassageSubtotal = new JLabel("₱0.00");
JLabel aromaFacialSubtotal = new JLabel("₱0.00");
JLabel poolAccessSubtotal = new JLabel("₱0.00");
JLabel gymAccessSubtotal = new JLabel("₱0.00");
JLabel footSpaSubtotal = new JLabel("₱0.00");

```

```
// Thai Massage
gbc.gridx = 0;
gbc.gridy = 0;
addonsPanel.add(thaiMassageLabel, gbc);

gbc.gridx = 1;
addonsPanel.add(thaiMassageSpinner, gbc);

gbc.gridx = 2;
addonsPanel.add(thaiMassageSubtotal, gbc);

// Aroma Facial Massage
gbc.gridx = 0;
gbc.gridy = 1;
addonsPanel.add(aromaFacialLabel, gbc);

gbc.gridx = 1;
addonsPanel.add(aromaFacialSpinner, gbc);

gbc.gridx = 2;
addonsPanel.add(aromaFacialSubtotal, gbc);

// Access to Pool
gbc.gridx = 0;
gbc.gridy = 2;
addonsPanel.add(poolAccessLabel, gbc);

gbc.gridx = 1;
addonsPanel.add(poolAccessSpinner, gbc);

gbc.gridx = 2;
addonsPanel.add(poolAccessSubtotal, gbc);

// Gym Access
gbc.gridx = 0;
gbc.gridy = 3;
addonsPanel.add(gymAccessLabel, gbc);

gbc.gridx = 1;
addonsPanel.add(gymAccessSpinner, gbc);

gbc.gridx = 2;
addonsPanel.add(gymAccessSubtotal, gbc);

// Foot Spa Services
gbc.gridx = 0;
gbc.gridy = 4;
addonsPanel.add(footSpaLabel, gbc);
```

```

gbc.gridx = 1;
addonsPanel.add(footSpaSpinner, gbc);

gbc.gridx = 2;
addonsPanel.add(footSpaSubtotal, gbc);

// Total Panel
JPanel totalPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
totalPanel.setBackground(new Color(245, 245, 245));

JLabel totalLabel = new JLabel("Total: ₱0.00");
totalLabel.setFont(new Font("Serif", Font.BOLD, 16));
totalPanel.add(totalLabel);

// Add change listeners to update subtotals and total price
addChangeListener(thaiMassageSpinner, 1540.00, thaiMassageSubtotal, totalLabel,
thaiMassageSpinner, aromaFacialSpinner, poolAccessSpinner, gymAccessSpinner,
footSpaSpinner);
addChangeListener(aromaFacialSpinner, 1045.00, aromaFacialSubtotal, totalLabel,
thaiMassageSpinner, aromaFacialSpinner, poolAccessSpinner, gymAccessSpinner,
footSpaSpinner);
addChangeListener(poolAccessSpinner, 300.00, poolAccessSubtotal, totalLabel,
thaiMassageSpinner, aromaFacialSpinner, poolAccessSpinner, gymAccessSpinner,
footSpaSpinner);
addChangeListener(gymAccessSpinner, 500.00, gymAccessSubtotal, totalLabel,
thaiMassageSpinner, aromaFacialSpinner, poolAccessSpinner, gymAccessSpinner,
footSpaSpinner);
addChangeListener(footSpaSpinner, 825.00, footSpaSubtotal, totalLabel,
thaiMassageSpinner, aromaFacialSpinner, poolAccessSpinner, gymAccessSpinner,
footSpaSpinner);

panel.add(addonsPanel, BorderLayout.CENTER);
panel.add(totalPanel, BorderLayout.SOUTH);

// Navigation Buttons
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
JButton backButton = new JButton("Back");
JButton nextButton = new JButton("Next");
backButton.setFont(new Font("Serif", Font.BOLD, 16));
nextButton.setFont(new Font("Serif", Font.BOLD, 16));
backButton.setBackground(new Color(100, 100, 100));
nextButton.setBackground(new Color(70, 130, 180));
backButton.setForeground(Color.WHITE);
nextButton.setForeground(Color.WHITE);

// Back button action to navigate to the previous panel

```

```

        backButton.addActionListener(e -> cardLayout.show(cardPanel, "6"));

// Next button action to navigate to the next panel
nextButton.addActionListener(
    e -> {
        int thaiMassageQty = (int) thaiMassageSpinner.getValue();
        int aromaFacialQty = (int) aromaFacialSpinner.getValue();
        int poolAccessQty = (int) poolAccessSpinner.getValue();
        int gymAccessQty = (int) gymAccessSpinner.getValue();
        int footSpaQty = (int) footSpaSpinner.getValue();

        // Calculate the total price for each amenity
        int thaiMassagePrice = thaiMassageQty * 1540;
        int aromaFacialPrice = aromaFacialQty * 1045;
        int poolAccessPrice = poolAccessQty * 300;
        int gymAccessPrice = gymAccessQty * 500;
        int footSpaPrice = footSpaQty * 825;
        int amenityTotalPrice = thaiMassagePrice + aromaFacialPrice + poolAccessPrice +
gymAccessPrice + footSpaPrice;

        // Output the details
        System.out.println("Selected Amenities:");
        System.out.println("Thai Massage: " + thaiMassageQty + " x ₱1540 = ₱" +
thaiMassagePrice);
        System.out.println("Aroma Facial Massage: " + aromaFacialQty + " x ₱1045 = ₱" +
aromaFacialPrice);
        System.out.println("Access to Pool: " + poolAccessQty + " x ₱300 = ₱" +
poolAccessPrice);
        System.out.println("Gym Access: " + gymAccessQty + " x ₱500 = ₱" +
gymAccessPrice);
        System.out.println("Foot Spa Services: " + footSpaQty + " x ₱825 = ₱" +
footSpaPrice);
        System.out.println("Total Amenity Price: ₱" + amenityTotalPrice);
        System.out.println("Total: " + totalLabel.getText());

        // Navigate to the confirmation panel
        cardLayout.show(cardPanel, "8"); // Replace "9" with the confirmation panel key
    });

    buttonPanel.add(backButton);
    buttonPanel.add(nextButton);

    panel.add(buttonPanel, BorderLayout.SOUTH);

    return panel;
}

```

```

private void addChangeListener(JSpinner spinner, double pricePerUnit, JLabel
subtotalLabel, JLabel totalLabel, JSpinner... otherSpinners) {
    spinner.addChangeListener(
        e -> {
            // Update the subtotal for the specific spinner
            int quantity = (int) spinner.getValue();
            double subtotal = quantity * pricePerUnit;
            subtotalLabel.setText(String.format("₱%.2f", subtotal));

            // Recalculate the total price
            double total = 0.0;
            for (JSpinner otherSpinner : otherSpinners) {
                int otherQuantity = (int) otherSpinner.getValue();
                double otherPricePerUnit = 0.0;

                // Assign price based on spinner (you can create a map to streamline this if needed)
                if (otherSpinner == spinner) {
                    otherPricePerUnit = pricePerUnit;
                } else if (otherSpinner == otherSpinners[0]) {
                    otherPricePerUnit = 1540.00; // Price for Thai Massage
                } else if (otherSpinner == otherSpinners[1]) {
                    otherPricePerUnit = 1045.00; // Price for Aroma Facial Massage
                } else if (otherSpinner == otherSpinners[2]) {
                    otherPricePerUnit = 300.00; // Price for Pool Access
                } else if (otherSpinner == otherSpinners[3]) {
                    otherPricePerUnit = 500.00; // Price for Gym Access
                } else if (otherSpinner == otherSpinners[4]) {
                    otherPricePerUnit = 825.00; // Price for Foot Spa
                }

                total += otherQuantity * otherPricePerUnit;
            }

            totalLabel.setText(String.format("Total: ₱%.2f", total));
        });
}

```

```

private JPanel createAddonsPanel() {
    JPanel panel = new JPanel();
    panel.setLayout(new BorderLayout());
    panel.setBackground(new Color(245, 245, 245)); // Light gray background
    panel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2)); // Border around the
panel
    panel.setPreferredSize(new Dimension(500, 400)); // Fixed size
}

```



```

// Title
JLabel title = new JLabel("Select Add-ons", SwingConstants.CENTER);
title.setFont(new Font("Serif", Font.BOLD, 24));
title.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
panel.add(title, BorderLayout.NORTH);
title.setForeground(new Color(70, 130, 180));
title.setAlignmentX(Component.CENTER_ALIGNMENT);

// Add-ons Panel with GridBagLayout
JPanel addonsPanel = new JPanel();
addonsPanel.setLayout(new GridBagLayout());
addonsPanel.setBorder(BorderFactory.createTitledBorder(
    BorderFactory.createLineBorder(Color.GRAY, 2), "Available Add-ons"));
addonsPanel.setBackground(new Color(245, 245, 245));

GridBagConstraints gbc = new GridBagConstraints();
gbc.insets = new Insets(10, 10, 10, 10); // Padding around components
gbc.fill = GridBagConstraints.HORIZONTAL; // Ensure components fill the grid cells

// Add-ons with Quantities
JLabel bedLabel = new JLabel("Bed (₱650.00/night):");
JLabel blanketLabel = new JLabel("Blanket (₱250.00/night):");
JLabel pillowLabel = new JLabel("Pillow (₱100.00/night):");
JLabel toiletriesLabel = new JLabel("Toiletries (₱200.00/set):");

JSpinner bedSpinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1)); // Quantity
for bed
JSpinner blanketSpinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1)); //
Quantity for blanket
JSpinner pillowSpinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1)); //
Quantity for pillow
JSpinner toiletriesSpinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1)); //
Quantity for toiletries

JLabel bedSubtotal = new JLabel("₱0.00");
JLabel blanketSubtotal = new JLabel("₱0.00");
JLabel pillowSubtotal = new JLabel("₱0.00");
JLabel toiletriesSubtotal = new JLabel("₱0.00");

// Bed
gbc.gridx = 0;
gbc.gridy = 0;
addonsPanel.add(bedLabel, gbc);

gbc.gridx = 1;
addonsPanel.add(bedSpinner, gbc);

gbc.gridx = 2;

```

```

addonsPanel.add(bedSubtotal, gbc);

// Blanket
gbc.gridx = 0;
gbc.gridy = 1;
addonsPanel.add(blanketLabel, gbc);

gbc.gridx = 1;
addonsPanel.add(blanketSpinner, gbc);

gbc.gridx = 2;
addonsPanel.add(blanketSubtotal, gbc);

// Pillow
gbc.gridx = 0;
gbc.gridy = 2;
addonsPanel.add(pillowLabel, gbc);

gbc.gridx = 1;
addonsPanel.add(pillowSpinner, gbc);

gbc.gridx = 2;
addonsPanel.add(pillowSubtotal, gbc);

// Toiletries
gbc.gridx = 0;
gbc.gridy = 3;
addonsPanel.add(toiletriesLabel, gbc);

gbc.gridx = 1;
addonsPanel.add(toiletriesSpinner, gbc);

gbc.gridx = 2;
addonsPanel.add(toiletriesSubtotal, gbc);

// Total Panel
JPanel totalPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
totalPanel.setBackground(new Color(245, 245, 245));

JLabel totalLabel = new JLabel("Total: ₱0.00");
totalLabel.setFont(new Font("Serif", Font.BOLD, 16));
totalPanel.add(totalLabel);

// Add change listeners to update subtotals and total price
addChangeListener(bedSpinner, 650.00, bedSubtotal, totalLabel, blanketSpinner,
pillowSpinner, toiletriesSpinner);
addChangeListener(blanketSpinner, 250.00, blanketSubtotal, totalLabel, bedSpinner,
pillowSpinner, toiletriesSpinner);

```

```

        addChangeListener(pillowSpinner, 100.00, pillowSubtotal, totalLabel, bedSpinner,
blanketSpinner, toiletriesSpinner);
        addChangeListener(toiletriesSpinner, 200.00, toiletriesSubtotal, totalLabel, bedSpinner,
blanketSpinner, pillowSpinner);

        panel.add(addonsPanel, BorderLayout.CENTER);
        panel.add(totalPanel, BorderLayout.SOUTH);

// Navigation Buttons
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
JButton backButton = new JButton("Back");
JButton nextButton = new JButton("Next");
backButton.setFont(new Font("Serif", Font.BOLD, 16));
nextButton.setFont(new Font("Serif", Font.BOLD, 16));
backButton.setBackground(new Color(100, 100, 100));
nextButton.setBackground(new Color(70, 130, 180));
backButton.setForeground(Color.WHITE);
nextButton.setForeground(Color.WHITE);

// Back button action to navigate to the previous panel
backButton.addActionListener(e -> cardLayout.show(cardPanel, "7")); // Replace "2" with
the destination panel key

// Next button action to navigate to the next panel
nextButton.addActionListener(
    e ->
    {
        int bedQty = (int) bedSpinner.getValue();
        int blanketQty = (int) blanketSpinner.getValue();
        int pillowQty = (int) pillowSpinner.getValue();
        int toiletriesQty = (int) toiletriesSpinner.getValue();

        // Calculate the total price for each amenity
        int bedPrice = bedQty * 650; // Bed price is ₱650.00 each
        int blanketPrice = blanketQty * 250; // Blanket price is ₱250.00 each
        int pillowPrice = pillowQty * 100; // Pillow price is ₱100.00 each
        int toiletriesPrice = toiletriesQty * 200; // Toiletries price is ₱200.00 each
        int amenityTotalPrice = bedPrice + blanketPrice + pillowPrice + toiletriesPrice;

        // Output the details
        System.out.println("Selected Add-ons:");
        System.out.println("Bed: " + bedQty + " x ₱650.00 = ₱" + bedPrice);
        System.out.println("Blanket: " + blanketQty + " x ₱250.00 = ₱" + blanketPrice);
        System.out.println("Pillow: " + pillowQty + " x ₱100.00 = ₱" + pillowPrice);
        System.out.println("Toiletries: " + toiletriesQty + " x ₱200.00 = ₱" + toiletriesPrice);
        System.out.println("Total Amenity Price: ₱" + amenityTotalPrice);
        System.out.println("Total: " + totalLabel.getText());
    }
);

```

```

        // Navigate to the confirmation panel
        cardLayout.show(cardPanel, "9"); // Replace "8" with the confirmation panel key
    });

    buttonPanel.add(backButton);
    buttonPanel.add(nextButton);

    panel.add(buttonPanel, BorderLayout.SOUTH);

    return panel;
}

```

```

private JPanel createConfirmationPanel() {
    JPanel panel = new JPanel();
    panel.setLayout(new GridBagLayout()); // Use GridBagLayout for centering
    panel.setBackground(new Color(245, 245, 245));

    GridBagConstraints gbc = new GridBagConstraints();
    gbc.insets = new Insets(10, 10, 10, 10); // Padding around components
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.anchor = GridBagConstraints.CENTER; // Center alignment for all components
    gbc.gridx = 0; // Single column
    gbc.weightx = 1.0; // Equal horizontal space distribution

    // Title
    JLabel title = new JLabel("Booking Confirmation", SwingConstants.CENTER);
    title.setFont(new Font("Serif", Font.BOLD, 24));
    title.setForeground(new Color(70, 130, 180));
    gbc.gridy = 0;
    gbc.gridwidth = 2; // Span across two columns
    panel.add(title, gbc);

    // Summary Label
    JLabel summaryLabel = new JLabel("Booking Details:");
    summaryLabel.setFont(new Font("Serif", Font.BOLD, 18));
    gbc.gridy = 1;
    gbc.gridwidth = 1; // Revert back to single-column span
    panel.add(summaryLabel, gbc);

    // Details Area
    JTextArea detailsArea = new JTextArea();
    detailsArea.setText(generateDetailsSummary()); // Remove arguments
    detailsArea.setFont(new Font("Serif", Font.PLAIN, 16));
    detailsArea.setEditable(false);
    detailsArea.setRows(10);
    detailsArea.setColumns(40); // Set width
    detailsArea.setBorder(BorderFactory.createLineBorder(Color.GRAY, 1));
}

```

```

detailsArea.setText(generateDetailsSummary());
gbc.gridy = 2;
gbc.gridwidth = 2; // Span across two columns
panel.add(new JScrollPane(detailsArea), gbc); // Add JScrollPane for better usability


// Button Panel
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10));
buttonPanel.setBackground(new Color(245, 245, 245));


// Back Button
JButton backButton = new JButton("Back");
backButton.setFont(new Font("Serif", Font.BOLD, 16));
backButton.setBackground(new Color(100, 100, 100));
backButton.setForeground(Color.WHITE);
backButton.addActionListener(e -> cardLayout.show(cardPanel, "8")); // Back to
destination panel
buttonPanel.add(backButton);


// Confirm Button
JButton confirmButton = new JButton("Confirm");
confirmButton.setFont(new Font("Serif", Font.BOLD, 16));
confirmButton.setBackground(new Color(70, 130, 180));
confirmButton.setForeground(Color.WHITE);
confirmButton.addActionListener(
    e ->
    {
        JOptionPane.showMessageDialog(panel, "Your booking has been confirmed!",
"Confirmation", JOptionPane.INFORMATION_MESSAGE);
        System.exit(0); // Exit application
    });
buttonPanel.add(confirmButton);


gbc.gridy = 3;
gbc.gridwidth = 2; // Span across two columns
panel.add(buttonPanel, gbc);


return panel;
}


// Assemble the details into a summary string
private String generateDetailsSummary() {
    StringBuilder summary = new StringBuilder();
    //summary.append("User ").append(selectedUser).append("\n");
    summary.append("Selected Destination: ").append(selectedDestination).append("\n");
    summary.append("Check-In Date: ").append(checkInDate).append("\n");
    summary.append("Check-Out Date: ").append(checkOutDate).append("\n");
}

```

```

summary.append("Number of Adults: ").append(numAdultGuest).append("\n");
summary.append("Number of Children: ").append(numChildGuest).append("\n");
summary.append("Total Guests: ").append(numAllGuest).append("\n");

summary.append("Room Type: ").append(selectedRoomType).append("\n");
summary.append("Amenities: ").append(selectedAmenities).append("\n");
summary.append("Add-ons: ").append(selectedAddOns).append("\n");
summary.append("Thank you for booking with us!");
return summary.toString();
}

private void addChangeListener(JSpinner spinner, double price, JLabel subtotalLabel,
JLabel totalLabel,
                        JSpinner other1, JSpinner other2, JSpinner other3) {
    spinner.addChangeListener(
        e -> {
            int quantity = (int) spinner.getValue();
            double subtotal = quantity * price;
            subtotalLabel.setText(String.format("₱%.2f", subtotal));
            updateTotalPrice(totalLabel, other1, other2, other3, spinner, price);
        });
}

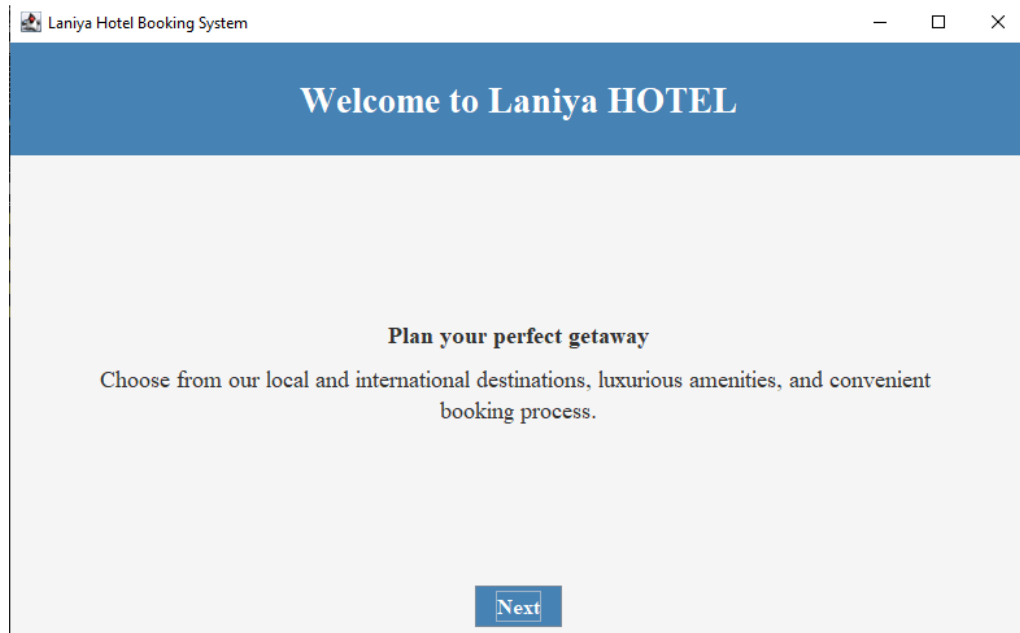
// Method to Update Total Price
private void updateTotalPrice(JLabel totalLabel, JSpinner spinner1, JSpinner spinner2,
JSpinner spinner3, JSpinner spinner4, double price) {
    double total = 0.0;
    total += (int) spinner1.getValue() * 650.00; // Bed
    total += (int) spinner2.getValue() * 250.00; // Blanket
    total += (int) spinner3.getValue() * 100.00; // Pillow
    total += (int) spinner4.getValue() * 200.00; // Toiletries
    totalLabel.setText(String.format("Total: ₱%.2f", total));
}

public static void main(String[] args)
{
    SwingUtilities.invokeLater(HotelGUI4::new);
}
}

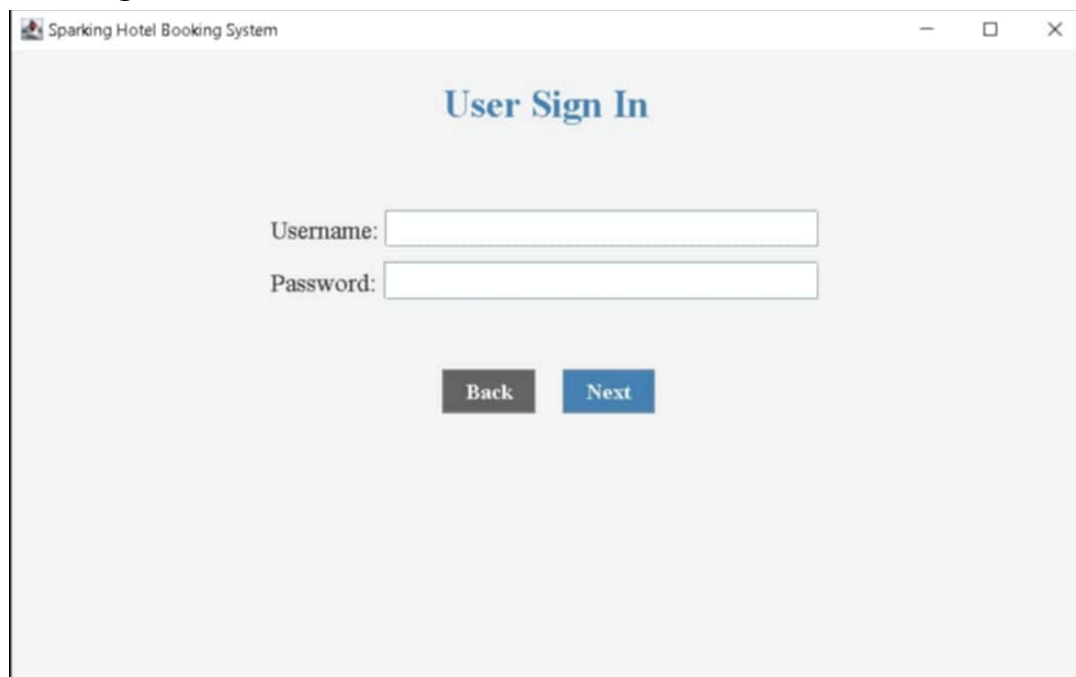
```

# JFrame

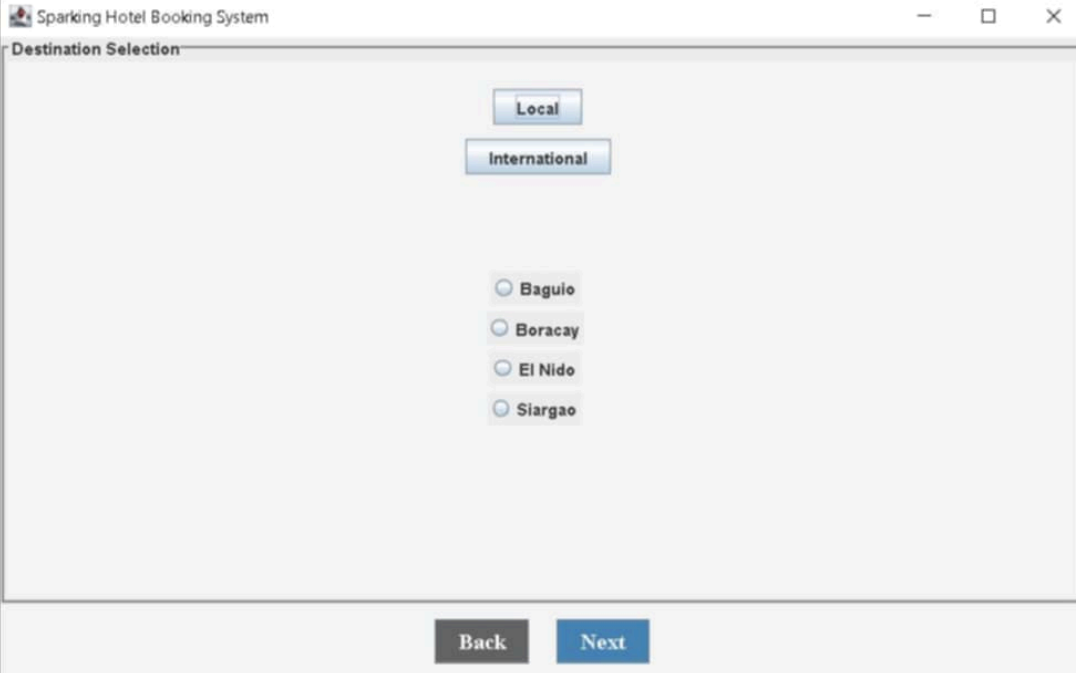
## Home page:



## User Sign in:

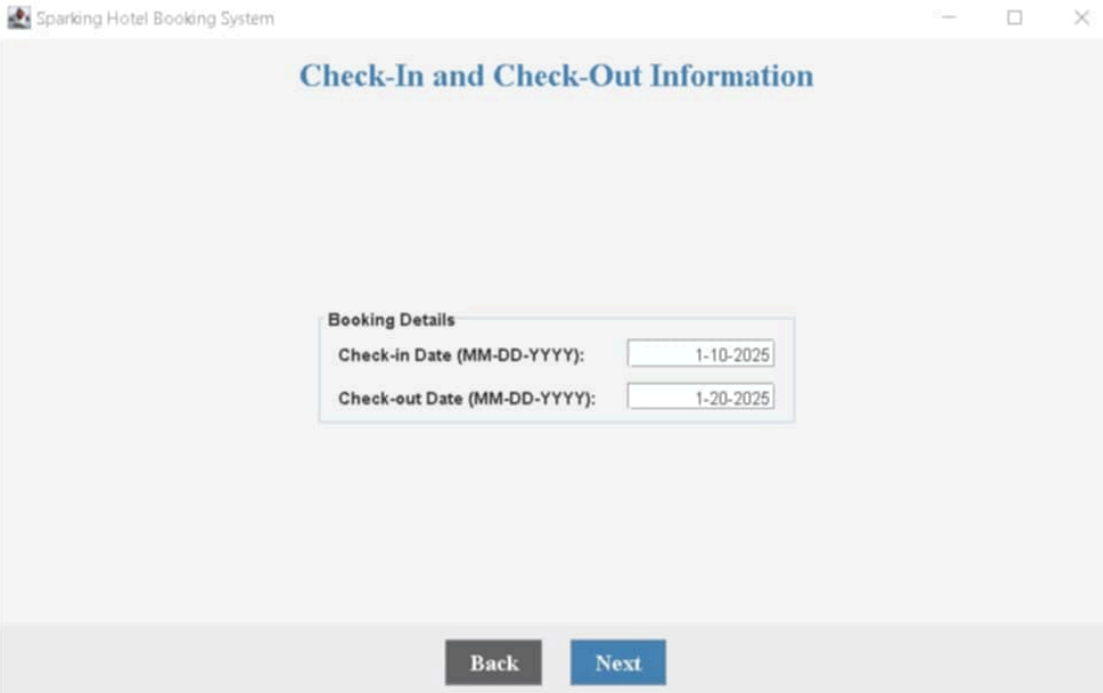


## Destination panel:



The screenshot shows a window titled "Sparking Hotel Booking System" with a subtitle "Destination Selection". Inside the window, there are two buttons: "Local" and "International". Below these buttons, there are four radio buttons with labels: "Baguio", "Boracay", "El Nido", and "Siargao". At the bottom of the window, there are two buttons: "Back" and "Next".

## Check-in Check-out:



The screenshot shows a window titled "Sparking Hotel Booking System" with a subtitle "Check-In and Check-Out Information". Inside the window, there is a section titled "Booking Details" which contains two rows of input fields. The first row is labeled "Check-in Date (MM-DD-YYYY):" and has a text box containing "1-10-2025". The second row is labeled "Check-out Date (MM-DD-YYYY):" and has a text box containing "1-20-2025". At the bottom of the window, there are two buttons: "Back" and "Next".



## Guest Panel:

Sparking Hotel Booking System

—

□

×

### Guest Information

Enter Guest Details

Number of Adults:

1

Number of Children:

2

Back

Next

## Child Age:

Child Ages

×

Please enter the age of each child (0-7):

Child 1 Age:

5

Child 2 Age:

9

OK

Cancel

## Room Suggestion Panel:

Sparking Hotel Booking System

— □ ×

### Room Suggestions

Suggested Room type is Double

Standard

☒ Capacity: 1 guest with 1 extra bed

Deluxe

☐ Capacity: 2 guests with 1 extra bed

Quadruple

☐ Capacity: 4 guests with 1 extra bed

Family

☐ Capacity: 6 guests with 1 extra bed

Suite

☐ Capacity: 4 guests

Back

Next

## Amenities Panel:

Sparking Hotel Booking System

— □ ×

### Select Amenities

☐ Thai Massage (฿1540)

☐ Aroma Facial Massage (฿1045)

☐ Access to Pool (฿300)


☐ Gym Access (฿500)

☐ Foot Spa Services (฿825)

Back

Next

## Add-ons Panel:

 Sparkling Hotel Booking System

—

□

×

Select Add-ons

Available Add-ons

Bed (P650.00/night):

0

+

−

P0.00

Blanket (P250.00/night):

0

+

−

P0.00

Pillow (P100.00/night):

0

+

−

P0.00

Toiletries (P200.00/set):

0

+


−

P0.00

Back

Next

## Create Confirmation Panel:

 Sparkling Hotel Booking System

—

□

×

Booking Confirmation

Booking Details:

Selected Destination: None

Check-In Date:

Check-Out Date:

Number of Adults: 0

Number of Children: 0

Total Guests: 0

Room Type: Standard

Amenities: None

Add-ons: None

Thank you for booking with us!

Back

Confirm