

**Tourist site**

An

Object-Oriented Programming Concepts through Java Course Project Report in partial fulfillment of the degree

#### Bachelor of Technology

in

**Computer Science &Engineering**

**By**

**2103A52055 M.Dheeraj**

**2103A52118 P. Aashish**

**2103A52128 CH.Rohith**

**2103A52136 G. Suraj**

Under the Guidance of

**Sravan sir**

School of Computer Science and AI

**Submitted to**

**School of CS & AI**





# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## CERTIFICATE

This is to certify that the Object Oriented Programming through Java-Course Project Report entitled **“TOURIST SITE”** is a record of bonafide work carried out by the students M. Dheeraj, P.Aashish ,CH.Rohith, G.Surajbearing RollNo(s) 203A52055, 203A52118,2103A52128,2103A52136 during the academic year 2022-2023in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the SR University,Hasanparthy,Warangal.

**Lab In-charge** **Head of the Department**

# ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our project guide SRAVAN SIR **,**as well as Head of the CSE Department **Dr . M. Sheshikala**, Associate Professorfor guiding us from the beginning through the end of the Capstone Project with theirintellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which always constructive and encouraging and ultimately drove us to the right direction .We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Principal, **Dr.V.MAHESH,** for his continuous support and guidance to complete this project.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely supp

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TABLE OF CONTENTS**   |  |  |  | | --- | --- | --- | | **S.No** | **TITLE** | **Pg.no** | | **1** | Abstract | 5 | | **2** | Objective | 6 | | **3** | Definition of elements used in project | 7-8 | | **4** | Designs | 9 | | **5** | Implementation | 10-15 | | **6** | Result Screens | 16-18 | | **7** | Conclusions | 19 | |

# ABSTRACT

Our project is a Tourist interface. We have created a interface useful to the tourist to make their work easy and plan their schedules within their requirements.Basically, tourists plan their schedules in the weekends and in holidays.On this basis, we have created a interface consisting of weekend plans and long trip plans.Planning a trip is not easier in present situations based on the busy schedules. So, in our interface we will provide each element that is required to plan a trip based on their choice.Our interface consist of the details as days (weekend days )to be selected, number of persons to be visited and affordable budget .Based on this, tour packages and the total amount of the tour will be obtained as output based on the user’s requirements of holidays .Through this interface ,tourists can schedule their tour packages in a user-friendly and in a cost-effective way.

# OBJECTIVE

Our main aim is to focus on the people who want to travel to the different places. A tour package with a less number of days and budget-friendly package is not available. In our site, we suggest input to be given as budget, number of days, number of persons,location of their choice that would be affordable to the users. We will be providing the tour packages they can be visited based on the budget and the number of days using GUI(swings),if-else conditional statements, and exception handling concepts. So, we have created a user-friendly tourist interface to reach out people easily.

# 3.DEFINITIONS OF THE ELEMENTS USED IN

# PROJECT

# About Swings:

# Swing is a Java GUI (Graphical User Interface) library that provides a set of components and tools for building interactive desktop applications. It is a part of the Java Foundation Classes (JFC) and offers a rich set of graphical components like buttons, labels, text fields, and more, allowing developers to create user-friendly interfaces. Swing follows the Model-View-Controller (MVC) architecture, separating the logic, presentation, and user interaction components. It is platform-independent and provides a consistent look and feel across different operating systems, making it a powerful choice for cross-platform application development. Swing is widely used in Java desktop applications for its flexibility and ease of use.Java swing is a part of Java foundation classes ( JFC) that is used to create window based applications It is built on the top of Awt ( abstract window tool kit)API and entirely in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

**About AWT:**

Java AWT (Abstract Window Toolkit) is an API to develop Graphical User Interface (GUI) or windows-based applications in Java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavy weight i.e. its components are using the resources of underlying operating system (OS).

The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

The AWT will help the user to understand Java GUI programming in simple and easy steps.

1. **JTextField** :The object of a JTextField class is a text component that allows the editing of a single line text. It inherits JTextComponent class.
2. **JButton**:TheJButton class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed. It inherits AbstractButton class.
3. **ActionListener:** The Java Action Listener is notified whenever you click on the button or menu item. Itis notified against Action Event. The Action Listener interface is found injava.awt.event[package.](https://www.javatpoint.com/package)Ithasonlyone method: action Performed().

### **Packages**: Collection of similar types of classes, interfaces and sub packages. The package is both a naming and a visibility control mechanism.

### **Action Event:** An action event occurs, whenever an action is performed by the user. Examples: When the user clicks a button, chooses a menu item, presses Enter in a text field. The result is that an action Performed message is sent to all action listeners that are registered on the relevant component

### **Exceptional Handling:**Exception handling is the process of responding to unwanted or unexpected events when a computer program runs. Exception handling deals with these events to avoid the program or system crashing, and without this process, exceptions would disrupt the normal operation of a program.

### **User defined exception:**User defined exceptions are also referred as custom exceptions. The exceptions which are created as per our use case and thrown using **throw** keyword are user defined exceptions, such exceptions are derived classes of Exception class from java.lang package.

### **Try and catch keywords:** The try statement allows you to define a block of code to be tested for errors while it is being executed. The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

# 4.DESIGN

# 4.1 SCREENS

# 

# 5.IMPLEMENTATION

**CODE:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class NoBudgetException extends Exception {

public NoBudgetException() {

super("Budget not sufficient! Not available");

}

}

public class TouristGUI {

private JFrame frame;

private JPanel panel;

private JComboBox<String> choiceComboBox;

private JTextField daysField;

private JTextField budgetField;

private JTextField locationField;

private JTextField membersField;

private JButton calculateButton;

public TouristGUI() {

frame = new JFrame("Tourist Planner");

panel = new JPanel();

panel.setLayout(new GridLayout(7, 2));

String[] choices = {"1 Day", "2 Days", "3 Days"}; // Removed "1 Week"

choiceComboBox = new JComboBox<>(choices);

daysField = new JTextField();

budgetField = new JTextField();

locationField = new JTextField();

membersField = new JTextField();

calculateButton = new JButton("Calculate");

panel.add(new JLabel("Choose Duration:"));

panel.add(choiceComboBox);

panel.add(new JLabel("Enter Days (Sunday Monday Saturday):"));

panel.add(daysField);

panel.add(new JLabel("Enter Budget:"));

panel.add(budgetField);

panel.add(new JLabel("Enter Location (kerala,Tamil Nadu,Varanasi,Jammu,Delhi):"));

panel.add(locationField);

panel.add(new JLabel("Enter Number of Members:"));

panel.add(membersField);

panel.add(new JLabel("")); // Empty label for spacing

panel.add(calculateButton);

calculateButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

calculate();

} catch (NoBudgetException ex) {

JOptionPane.showMessageDialog(frame, ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

});

frame.add(panel);

frame.setSize(400, 400); // Increased the height for a more spacious look

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

}

private void calculate() throws NoBudgetException {

String choiceString = (String) choiceComboBox.getSelectedItem();

int choice = Integer.parseInt(choiceString.split(" ")[0]);

String days = daysField.getText().toLowerCase();

int budget = Integer.parseInt(budgetField.getText());

String location = locationField.getText().toLowerCase();

int numberOfMembers = Integer.parseInt(membersField.getText());

if (choice < 1 || choice > 3) { // Adjusted the choice range

throw new IllegalArgumentException("Invalid choice");

}

String placeToVisit = "";

int totalAmount = 0;

switch (choice) {

case 1:

if (budget < 1000 || budget > 2000) {

throw new NoBudgetException();

}

if (days.equals("sunday")) {

placeToVisit = "Bhogatha Waterfalls";

totalAmount = budget \* numberOfMembers;

} else if (days.equals("saturday")) {

placeToVisit = "Pakal";

totalAmount = budget \* numberOfMembers;

} else if (days.equals("monday")) {

placeToVisit = "Bhimneni Waterfalls";

totalAmount = budget \* numberOfMembers;

} else {

throw new IllegalArgumentException("Invalid day input");

}

break;

case 2:

String[] dayArray = days.split(" ");

if (dayArray.length != 2) {

throw new IllegalArgumentException("Invalid days input");

}

if ((dayArray[0].equals("saturday") && dayArray[1].equals("sunday")) ||

(dayArray[1].equals("saturday") && dayArray[0].equals("sunday"))) {

if (budget >= 3000 && budget <= 5000) {

placeToVisit = "Vizag";

totalAmount = budget \* numberOfMembers;

} else {

throw new NoBudgetException();

}

} else if ((dayArray[0].equals("sunday") && dayArray[1].equals("monday")) ||

(dayArray[1].equals("sunday") && dayArray[0].equals("monday"))) {

if (budget >= 3000 && budget <= 5000) { // Use && for logical AND

placeToVisit = "Nagarjuna Sagar";

totalAmount = budget \* numberOfMembers;

} else {

throw new NoBudgetException();

}

} else {

throw new IllegalArgumentException("Invalid days input");

}

break;

case 3:

String[] dayArray3 = days.split(" ");

if (dayArray3.length != 3) {

throw new IllegalArgumentException("Invalid days input");

}

if ((dayArray3[0].equals("saturday") || dayArray3[0].equals("sunday") || dayArray3[0].equals("monday")) &&

(dayArray3[1].equals("saturday") || dayArray3[1].equals("sunday") || dayArray3[1].equals("monday")) &&

(dayArray3[2].equals("saturday") || dayArray3[2].equals("sunday") || dayArray3[2].equals("monday"))) {

if (budget >= 4000 && budget <= 7000) {

placeToVisit = "Ooty, Shirdi, or Tirupati";

totalAmount = budget \* numberOfMembers;

} else {

throw new NoBudgetException();

}

} else {

throw new IllegalArgumentException("Invalid days input");

}

**break;**

**default:**

**throw new IllegalArgumentException("Invalid choice");**

}

// Display results

System.out.println("Places to visit: " + placeToVisit);

System.out.println("Total amount: " + totalAmount);

System.out.println("Enjoy your Trip :) Visit our Page Again");

}

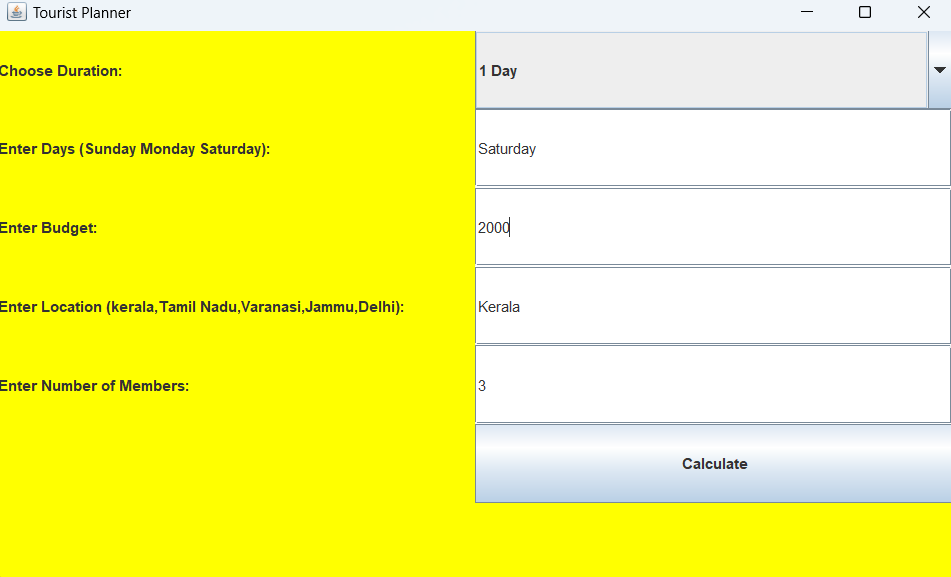
public static void main(String[] args) {

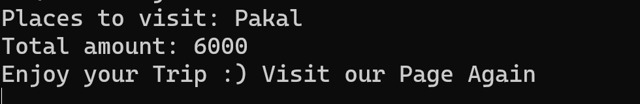
new TouristGUI();

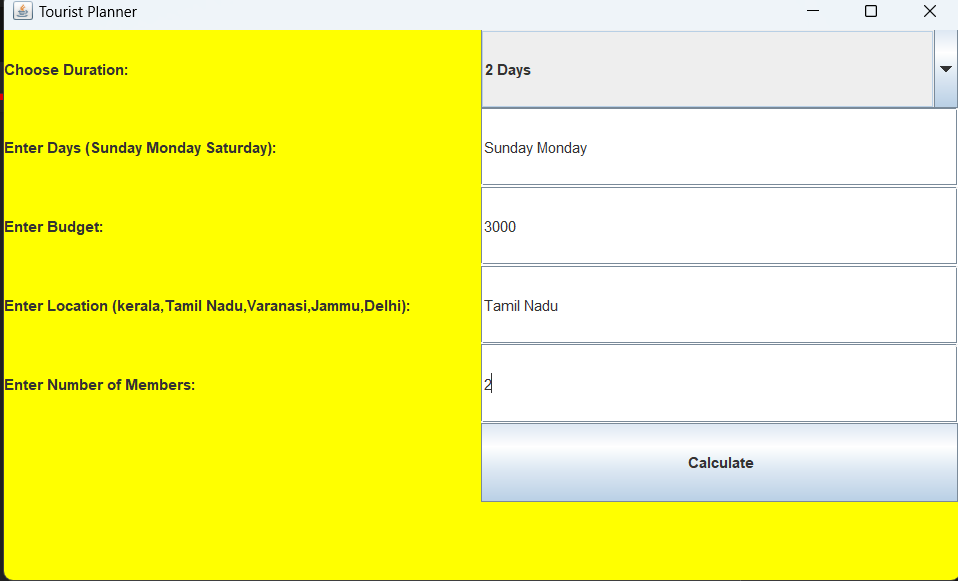
}

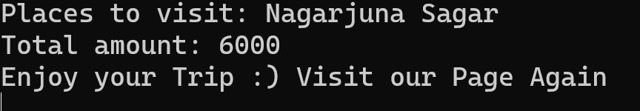
}

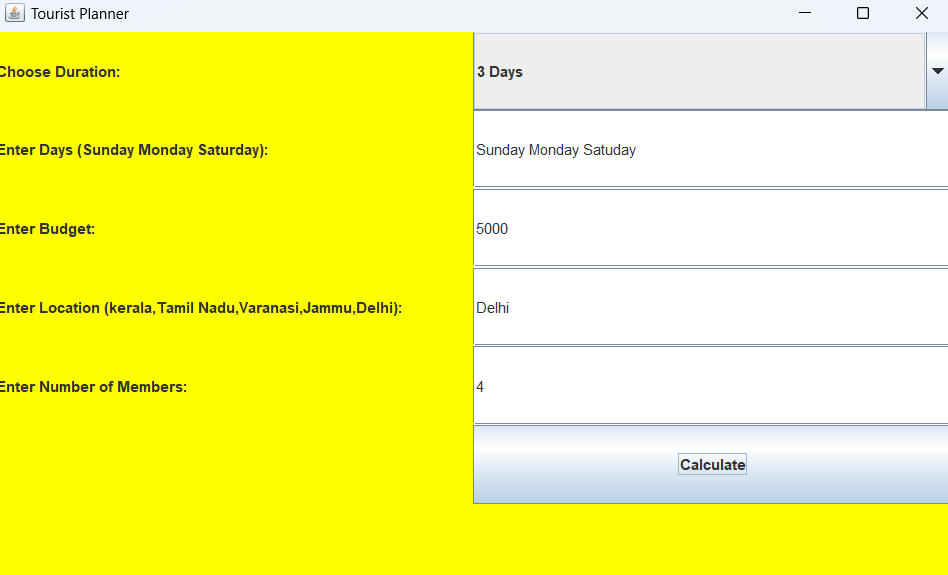
**6. RESULTS**

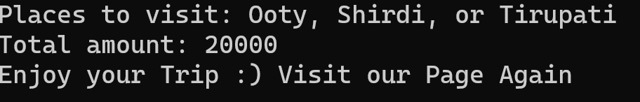












**7.CONCLUSION**

Finally, we conclude that, this project will be helpful for the people who are facing difficulties in scheduling their trips in the weekends with reasonable cost. By using this tourist site we are providing tour packages where visitors can easily use this project for travelling purpose by selecting the displayed options like number of persons, number of days they wanted to visit and at what cost they want to visit. After a while the total amount and places to visit will be suggested by the application. This may reduce the risk of searching the places for the tourists according to their budgets.

Link: <https://github.com/dheeraj0000/Tourism-site>