**TRIPOTO (TRIP EXPENSE MANAGER)**

A

Mini Project Report

Submitted in partial fulfilment of the Requirements for the award of the Degree of

## BACHELOR OF ENGINEERING

**IN**

## INFORMATION TECHNOLOGY

**By:**

**GOPI SATHWIK (1602-20-737-013)**

**P.HARSHITH (1602-20-737-016)**

**D.VAMSHIDHAR (1602-20-737-057)**



## Department of Information Technology Vasavi College of Engineering

## (Autonomous)

## (Affiliated to Osmania University)

**Ibrahimbagh, Hyderabad-31 2022**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University) Hyderabad-500 031**

**Department of Information Technology**



**DECLARATION BY THE CANDIDATE**

We, **GOPI SATHWIK**, **P.HARSHITH**, and **D.VAMSHIDHAR**, bearing hall ticket numbers, **1602-20-737-013**, **1602-20-737-016** and **1602-20-737-0157**, hereby declare that the project report entitled **“TRIPOTO”** is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology.

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

**GOPI SATHWIK**

**1602-20-737-013**

**P.HARSHITH**

**1602-20-737-016**

**D.VAMSHIDHAR**

**1602-20-737-057**

(Faculty In-Charge) **Dr.K.Ram Mohan Rao,**

**Professor & HOD,**

**Dept. of IT.**

## ACKNOWLEDGEMENT

We extend our sincere thanks to Dr. S. V. Ramana, Principal, Vasavi College of Engineering for his encouragement. We express our sincere gratitude to Dr. K. Ram Mohan Rao, Professor & Head, Department of Information Technology, Vasavi College of Engineering, for introducing the Mini-Project module in our curriculum, and also for his suggestions, motivation, and co-operation for the successful completion of our Mini Project. We also want to thank and convey our gratitude towards our mini project coordinators Ms. Lingineni Divya and Sriramoju Rajyalaxmi, for guiding us in understanding the process of project development & giving us timely suggestions at every phase. We would also like to sincerely thank the project reviewers for their valuable inputs and suggestions.

**Table of Contents**

1. ABSTRACT 5
2. INTRODUCTION 6
   1. MOTIVATION
   2. PROBLEM DEFINITION
   3. OBJECTIVE OF THE PROJECT
3. SYSTEM REQUIRNMENTS ..7
   1. HARDWARE REQUIREMENT
   2. SOFTWARE REQUIREMNET
4. PROPOSED WORK 8
   1. DESIGN………………………………………………8
   2. IMPLEMENTATION 10
   3. TESTING……………………….

5. GIT HUB LINK…………………………………………….26

6. RESULTS…………………………………………………...26

7. OUTCOMES APART FROM CURRICULUM…………….30

8. CONCLUSION & FURTHER WORK……………………..30

9. REFERENCES……………………………………………...33

**1. ABSTRACT**

The main objective of the project is to manage the trip expenses during a trip and to know weather conditions of the destination place. This project is useful for trip people. It is a GUI based application.

In this application, at maximum we can enter 5 trip traveler details and we can add their trip expense details i.e. a traveller can enter details for the what purpose he/she spent at specific time and date. We are allowed to track the expenses at any point of time.

We can know the exact weather conditions like current temperature, humidity, minimum temperature, maximum temperature and graticule location of destination place.

**2. INTRODUCTION**

TRIPOTO is an app which can be used by all trip people. The main objective of this app is to assist the users during their trip. The app mainly consists of two objectives , the first objective is to show the current weather conditions of the destination place and the second objective is to track the expenses of the team when they are on their tour.

**a) MOTIVATION**

Few months ago, we wanted to keep a bachelor room. We faced many problems to calculate the expenses and share the expenses among us. After a month, we vacated the bachelor room.

From the experience of the bachelor room we three of us (Vamshi, Harshith and Gopi) got an idea to create an app that can track the expenses and keep note of every single transaction including the time, date and purpose.

To improvise this play we extended to make an app i.e. TRIPOTO application which can track the expenses of travellers who are travelling in a trip.

We also added a new feature to our app that shows weather conditions at the destination place. Eventually this has also became a part of this app.

**b) PROBLEM DEFINITION**

This program is an application to find the weather conditions like current temperature, minimum temperature, humidity, graticule location of any place in the world. It will be difficult to travellers to keep on tracking the expenses when they are on a trip. They will not be able to note every single transaction. They also face problem in analysing their expenses based on the travellers and the purpose what did they spent about.

**c) OBJECTIVE OF THE PROJECT**

The Objective of the project is to track expenses of the trip of every single traveller and to know the weather conditions of the destination place.

**3. TECHNOLOGY**

**a) Hardware Requirements:**

* Minimum Ram required: 512 mb
* Minimum Disk Space required: 50 mb
* Processor: core i3 or above
* Input devices: Mouse
* Output devices: Monitor

## b) Software Requirements:

* Python 3.9 Latest Build
* Windows Desktop version (7/8/10/11)
* PyCharm IDE for windows devices
* Pydroid3 for Android devices

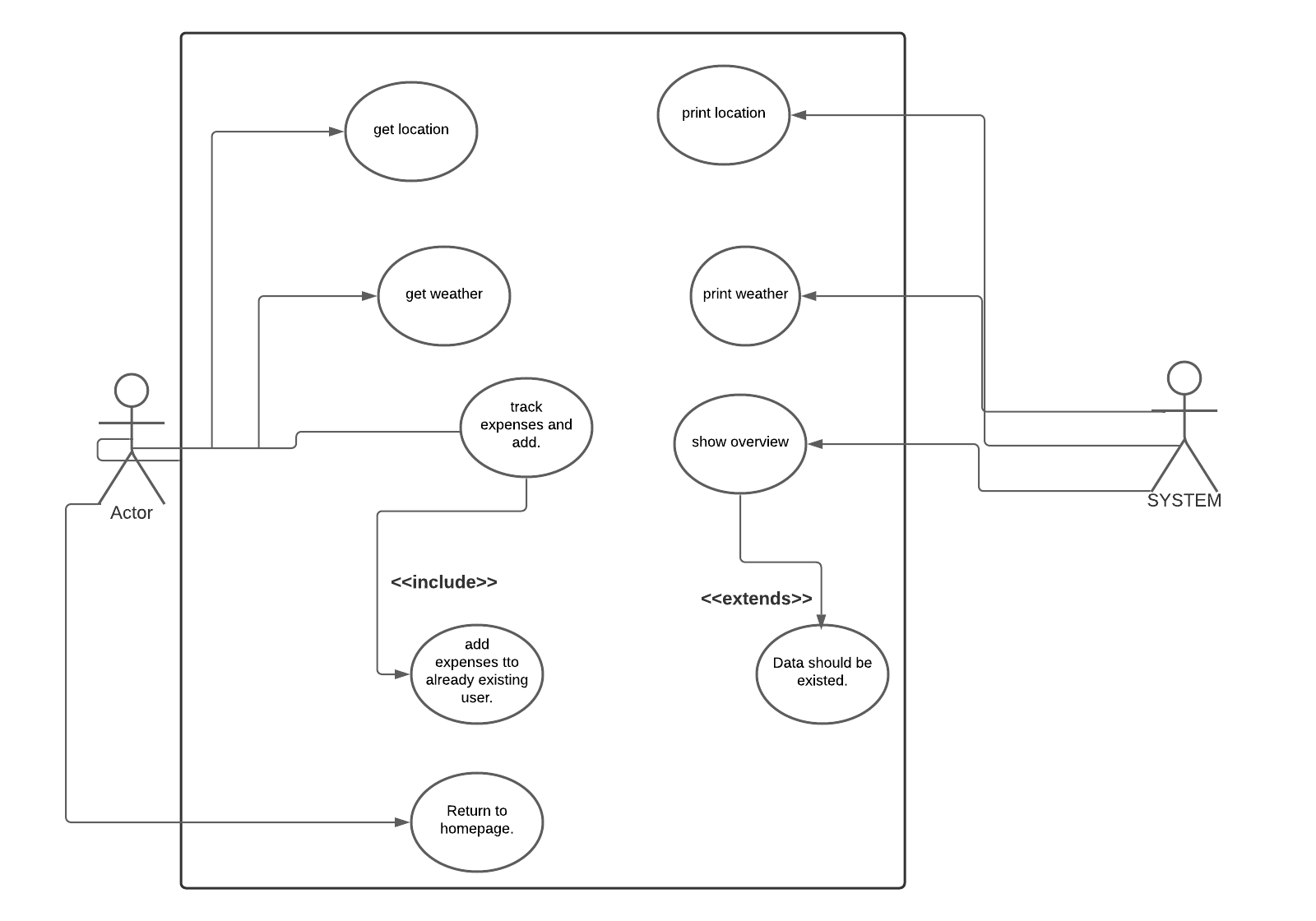
**4. PROPOSED WORK**

**a) DESIGN**

* **USE CASE DIAGRAM**

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how).

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

****

**b) USE CASE DESCRIPTION**

**User**

The user has to run the program and enter the details of travellers and destination place. User can heck the weather condition and track expenses.

**System**

The System displays weather conditions and each of expenses spent by each traveller for specific purpose at specific date and time.

**Weather conditions**

The user can get weather conditions like current temperature, humidity, maxmimum and minimum temperature.

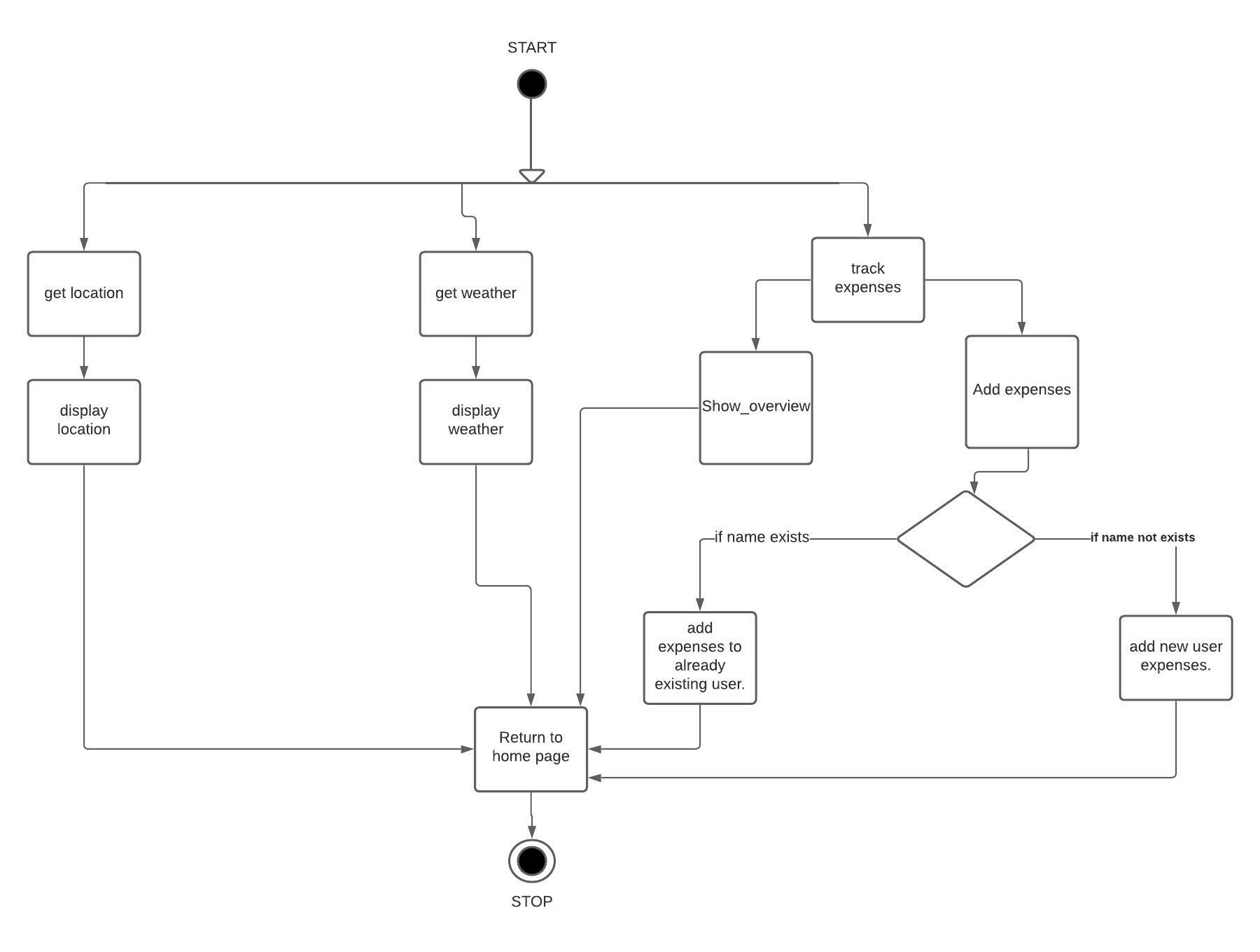
**Track Expenses**

The user can track expenses by adding the amount spent by each traveller for specific purpose at specific time and date.

**Back to home page**

The user can get back to home page after every task where he/she can choose weather conditions or track expenses.

**b) ACTIVITY DIAGRAM**

****

**c) IMPLEMENTATION**

* **MODULE WISE CODE**

import json

import kivy

from kivy.lang import Builder

from kivymd.app import MDApp

from kivy.core.window import Window

from kivy.uix.screenmanager import ScreenManager, Screen

from kivymd.uix.screen import MDScreen

from kivymd.uix.card import MDCard

import requests

from bs4 import BeautifulSoup

from kivymd.uix.snackbar import Snackbar

from datetime import datetime

class HelloScreen(MDScreen):

pass

class HomePage(MDScreen):

pass

class PostLogin(MDScreen):

pass

class Weather(MDScreen):

pass

class Overview(MDScreen):

pass

class Transactions(MDScreen):

pass

class AddTransaction(MDScreen):

pass

class TCard(MDCard):

pass

kvstring = '''<HelloScreen>:

Image:

source:'pic.png'

size\_hint\_x: 1

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

MDRaisedButton:

text: "Continue to APP"

font\_size: 20

pos\_hint: {"center\_x": 0.5, "center\_y": 0.15}

on\_press: app.continue\_to\_app()

<HomePage>:

MDLabel:

text: "WELCOME TO TRIPOTO"

font\_size: 35

pos\_hint: {"center\_x": 0.5, "center\_y": 0.95}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDLabel:

text: "Enter details of Teammates"

font\_size: 25

pos\_hint: {"center\_x": 0.5, "center\_y": 0.89}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDCard:

size\_hint: 0.95, 0.8

pos\_hint: {"center\_x": 0.5, "center\_y": 0.45}

elevation: 10

padding: 25

spacing: 25

orientation: 'vertical'

MDTextField:

id: phn1

hint\_text: "Person 1 Phone"

size\_hint\_x: 0.4

font\_size: 25

max\_text\_length: 10

pos\_hint: {"center\_x": 0.75, "center\_y": 0.65}

helper\_text\_mode: "on\_error"

helper\_text: "Enter phone number"

MDTextField:

id: p1

hint\_text: "Person 1"

size\_hint\_x: 0.4

helper\_text\_mode: "on\_error"

helper\_text: "Enter a name"

font\_size: 25

pos\_hint: {"center\_x": 0.25, "center\_y": 0.65}

MDTextField:

id: p2

hint\_text: "Person 2"

size\_hint\_x: 0.4

font\_size: 25

pos\_hint: {"center\_x": 0.25, "center\_y": 0.55}

helper\_text\_mode: "on\_error"

helper\_text: "Enter a name"

MDTextField:

id: phn2

hint\_text: "Person2 phone"

size\_hint\_x: 0.4

font\_size: 25

pos\_hint: {"center\_x": 0.75, "center\_y": 0.55}

max\_text\_length: 10

helper\_text\_mode: "on\_error"

helper\_text: "enter 10 digit phone number"

MDTextField:

id: p3

hint\_text: "Person 3"

size\_hint\_x: 0.4

font\_size: 25

pos\_hint: {"center\_x": 0.25, "center\_y": 0.45}

helper\_text\_mode: "on\_error"

helper\_text: "enter a name"

MDTextField:

id: phn3

hint\_text: "Person 3 phone"

size\_hint\_x: 0.4

font\_size: 25

pos\_hint: {"center\_x": 0.75, "center\_y": 0.45}

max\_text\_length: 10

helper\_text\_mode: "on\_error"

helper\_text: "enter 10 digit phone number"

MDTextField:

id: p4

hint\_text: "Person 4"

size\_hint\_x: 0.4

font\_size: 25

pos\_hint: {"center\_x": 0.25, "center\_y": 0.35}

helper\_text\_mode: "on\_error"

helper\_text: "Enter a name"

MDTextField:

id: phn4

hint\_text: "Person 4 phone"

size\_hint\_x: 0.4

font\_size: 25

pos\_hint: {"center\_x": 0.75, "center\_y": 0.35}

max\_text\_length: 10

helper\_text\_mode: "on\_error"

helper\_text: "enter 10 digit phone number"

MDTextField:

id: p5

hint\_text: "Person 5"

size\_hint\_x: 0.4

font\_size: 25

pos\_hint: {"center\_x": 0.25, "center\_y": 0.25}

helper\_text\_mode: "on\_error"

helper\_text: "enter a name"

MDTextField:

id: phn5

hint\_text: "Person 5 phone"

size\_hint\_x: 0.4

font\_size: 25

pos\_hint: {"center\_x": 0.75, "center\_y": 0.25}

helper\_text\_mode: "on\_error"

helper\_text: "Enter 10-digit phone number"

MDTextField:

id: destination

hint\_text: "Destination"

size\_hint\_x: 0.9

font\_size: 25

pos\_hint: {"center\_x": 0.5, "center\_y": 0.75}

required: True

helper\_text\_mode: "on\_error"

helper\_text: "Enter any place"

MDRoundFlatButton:

text: "Submit"

font\_size: 25

pos\_hint: {"center\_x": 0.5, "center\_y": 0.12}

size\_hint\_x: 0.7

on\_press: app.submit(p1.text,p2.text,p3.text,p4.text,p5.text,phn1.text,phn2.text,phn3.text,phn4.text,phn5.text,destination.text)

<PostLogin>:

MDLabel:

text: "Planning for a Trip this week?"

font\_size: 18

pos\_hint: {"center\_x": 0.5, "center\_y": 0.75}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDRoundFlatButton:

text: "Check Weather condition at your destination"

font\_size: 20

pos\_hint: {"center\_x": 0.5, "center\_y": 0.65}

on\_press: app.weather()

MDLabel:

text: "Trip Started?"

font\_size: 18

pos\_hint: {"center\_x": 0.5, "center\_y": 0.45}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDRoundFlatButton:

text: "Track expenses"

font\_size: 20

pos\_hint: {"center\_x": 0.5, "center\_y": 0.35}

on\_press: app.track()

<Weather>:

MDLabel:

text: "The weather details at "+app.destination+" is: "

font\_size: 30

pos\_hint: {"center\_x": 0.5, "center\_y": 0.90}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDLabel:

text: 'Current Temperature:'+str(app.current\_temperature)

font\_size: 25

pos\_hint: {"center\_x": 0.5, "center\_y": 0.80}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDLabel:

text: 'Humidity:'+str(app.humidity)

font\_size: 18

pos\_hint: {"center\_x": 0.5, "center\_y": 0.70}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDLabel:

text: 'Minimum Temperature:'+str(app.tempmin)

font\_size: 18

pos\_hint: {"center\_x": 0.5, "center\_y": 0.65}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDLabel:

text: 'Maximum Temperature:'+str(app.tempmax)

font\_size: 18

pos\_hint: {"center\_x": 0.5, "center\_y": 0.60}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDLabel:

text: "The location details of "+app.destination+" is :"

font\_size: 30

pos\_hint: {"center\_x": 0.5, "center\_y": 0.50}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDLabel:

text: 'Latitude:'+str(app.latitude)

font\_size: 18

pos\_hint: {"center\_x": 0.5, "center\_y": 0.40}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDLabel:

text: 'Longitude:'+str(app.longitude)

font\_size: 18

pos\_hint: {"center\_x": 0.5, "center\_y": 0.35}

halign: 'center'

size\_hint\_y: None

padding\_y: 15

MDRoundFlatButton:

text: "Go to Home Page"

font\_size: 20

pos\_hint: {"center\_x": 0.5, "center\_y": 0.15}

on\_press: app.goback()

<Overview>:

MDRaisedButton:

text: "Overview"

font\_size: 18

md\_bg\_color: 1, 0, 1, 1

text\_color: 0, 0, 1, 1

pos\_hint: {"center\_x": 0.25, "center\_y": 0.85}

size\_hint: 0.5,0.075

on\_press: app.overview()

border: 'yellow'

MDFlatButton:

text: "Transactions"

font\_size: 18

pos\_hint: {"center\_x": 0.75, "center\_y": 0.85}

on\_press: app.transactions()

size\_hint: 0.5,0.075

MDCard:

size\_hint: 0.48, 0.32

pos\_hint: {"center\_x": 0.27, "center\_y": 0.65}

elevation: 10

padding: 25

spacing: 25

orientation: 'vertical'

MDLabel:

text: app.persons[0]

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: app.persons[1]

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: app.persons[2]

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: app.persons[3]

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: app.persons[4]

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDCard:

size\_hint: 0.48, 0.32

pos\_hint: {"center\_x": 0.73, "center\_y": 0.65}

elevation: 10

padding: 25

spacing: 25

orientation: 'vertical'

MDLabel:

text: str(app.spends[app.persons[0]])

font\_size: 15

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: str(app.spends[app.persons[1]])

font\_size: 15

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: str(app.spends[app.persons[2]])

font\_size: 15

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: str(app.spends[app.persons[3]])

font\_size: 15

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: str(app.spends[app.persons[4]])

font\_size: 15

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDCard:

size\_hint: 0.48, 0.4

pos\_hint: {"center\_x": 0.27, "center\_y": 0.25}

elevation: 10

padding: 25

spacing: 25

orientation: 'vertical'

MDLabel:

text: 'Food'

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: "Utilities"

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: "Travelling"

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: "Parties"

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: "Others"

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDCard:

size\_hint: 0.48, 0.4

pos\_hint: {"center\_x": 0.73, "center\_y": 0.25}

elevation: 10

padding: 25

spacing: 25

orientation: 'vertical'

MDLabel:

text: str(app.cwisespends["Food"])

font\_size: 15

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: str(app.cwisespends["Utilities"])

font\_size: 15

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: str(app.cwisespends["Travelling"])

font\_size: 15

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: str(app.cwisespends["Parties"])

font\_size: 15

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDLabel:

text: str(app.cwisespends["Others"])

font\_size: 18

halign: left

pos\_hint: {"center\_x": 0.5, "center\_y": 0.5}

halign: 'center'

MDFloatingActionButtonSpeedDial:

data: app.data

rotation\_root\_button: True

callback: app.plus

Widget:

size\_hint\_y: None

height: 10

MDNavigationLayout:

ScreenManager:

MDScreen:

MDToolbar:

title: "Track Expenses"

elevation: 10

pos\_hint: {"top": 1}

left\_action\_items:

[['keyboard-backspace', lambda x: app.goback()]]

<Transactions>:

MDRaisedButton:

text: "Transactions"

font\_size: 18

md\_bg\_color: 1, 0, 1, 1

text\_color: 0, 0, 1, 1

pos\_hint: {"center\_x": 0.75, "center\_y": 0.85}

size\_hint: 0.5,0.075

on\_press: app.transactions()

border: 'yellow'

MDFlatButton:

text: "Overview"

font\_size: 18

pos\_hint: {"center\_x": 0.25, "center\_y": 0.85}

on\_press: app.track()

size\_hint: 0.5,0.075

ScrollView:

id: scroll

do\_scroll\_x: False

do\_scroll\_y: True

size\_hint: 1,0.80

pos\_hint: {"center\_x": 0.5275, "y": 0}

MDList:

id: box

spacing: 20

MDFloatingActionButtonSpeedDial:

data: app.data

rotation\_root\_button: True

callback: app.plus

Widget:

size\_hint\_y: None

height: 10

MDNavigationLayout:

ScreenManager:

MDScreen:

MDToolbar:

title: "Track Expenses"

elevation: 10

pos\_hint: {"top": 1}

left\_action\_items:

[['keyboard-backspace', lambda x: app.goback()]]

<AddTransaction>

MDCard:

size\_hint: 0.95, 0.8

pos\_hint: {"center\_x": 0.5, "center\_y": 0.45}

elevation: 10

padding: 25

spacing: 25

orientation: 'vertical'

MDTextField:

id: amount

hint\_text: "Amount"

size\_hint\_x: 0.7

required: True

helper\_text\_mode: "on\_error"

helper\_text: "Enter Amount"

width: 180

font\_size: 25

pos\_hint: {"center\_x": 0.50, "center\_y": 0.75}

MDTextField:

id: purpose

hint\_text: "Purpose"

size\_hint\_x: 0.7

width: 180

font\_size: 25

pos\_hint: {"center\_x": 0.50, "center\_y": 0.65}

required: True

helper\_text\_mode: "on\_error"

helper\_text: "Enter purpose"

MDLabel:

text: "Food"

font\_size: 20

pos\_hint: {"center\_x": 0.70, "center\_y": 0.5}

size\_hint\_x: 0.30

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .8, 'center\_y': .5}

on\_active: app.on\_checkbox\_active('Food')

MDLabel:

text: "Utilities"

font\_size: 20

pos\_hint: {"center\_x": 0.70, "center\_y": 0.42}

size\_hint\_x: 0.30

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .8, 'center\_y': .42}

on\_active: app.on\_checkbox\_active('Utilities')

MDLabel:

text: "Travelling"

font\_size: 20

pos\_hint: {"center\_x": 0.7, "center\_y": 0.35}

size\_hint\_x: 0.3

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .8, 'center\_y': .35}

on\_active: app.on\_checkbox\_active('Travelling')

MDLabel:

text: "Parties"

font\_size: 20

pos\_hint: {"center\_x": 0.7, "center\_y": 0.27}

size\_hint\_x: 0.3

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .8, 'center\_y': .27}

on\_active: app.on\_checkbox\_active('Parties')

MDLabel:

text: "Others"

font\_size: 20

pos\_hint: {"center\_x": 0.7, "center\_y": 0.20}

size\_hint\_x: 0.3

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .8, 'center\_y': .20}

on\_active: app.on\_checkbox\_active('Others')

MDLabel:

text: app.persons[0]

font\_size: 20

pos\_hint: {"center\_x": 0.30, "center\_y": 0.5}

size\_hint\_x: 0.30

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .45, 'center\_y': .5}

on\_active: app.on\_person\_select(app.persons[0])

MDLabel:

text: app.persons[1]

font\_size: 20

pos\_hint: {"center\_x": 0.30, "center\_y": 0.42}

size\_hint\_x: 0.30

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .45, 'center\_y': .42}

on\_active: app.on\_person\_select(app.persons[1])

MDLabel:

text: app.persons[2]

font\_size: 20

pos\_hint: {"center\_x": 0.30, "center\_y": 0.35}

size\_hint\_x: 0.3

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .45, 'center\_y': .35}

on\_active: app.on\_person\_select(app.persons[2])

MDLabel:

text: app.persons[3]

font\_size: 20

pos\_hint: {"center\_x": 0.30, "center\_y": 0.27}

size\_hint\_x: 0.3

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .45, 'center\_y': .27}

on\_active: app.on\_person\_select(app.persons[3])

MDLabel:

text: app.persons[4]

font\_size: 20

pos\_hint: {"center\_x": 0.30, "center\_y": 0.20}

size\_hint\_x: 0.3

halign: 'left'

MDCheckbox:

size\_hint: None, None

size: "48dp", "48dp"

pos\_hint: {'center\_x': .45, 'center\_y': .20}

on\_active: app.on\_person\_select(app.persons[4])

MDRaisedButton:

text: "Add"

font\_size: 25

pos\_hint: {"center\_x": 0.5, "center\_y": 0.12}

size\_hint\_x: 0.7

on\_press: app.addSubmit(amount.text,purpose.text)

Widget:

size\_hint\_y: None

height: 10

MDNavigationLayout:

ScreenManager:

MDScreen:

MDToolbar:

title: "Add Transaction"

elevation: 10

pos\_hint: {"top": 1}

left\_action\_items:

[['keyboard-backspace', lambda x: app.goback()]]

<TCard>:

id: tcard

orientation: "vertical"

size\_hint: 1, None

height: box\_top.height + box\_bottom.height

focus\_behavior: True

ripple\_behavior: True

pos\_hint: {"center\_x": .5, "center\_y": .5}

MDBoxLayout:

id: box\_top

spacing: "20dp"

adaptive\_height: True

MDBoxLayout:

id: text\_box

orientation: "vertical"

adaptive\_height: True

spacing: "10dp"

padding: 0, "10dp", "10dp", "10dp"

MDLabel:

text: app.money

theme\_text\_color: "Primary"

font\_style: "H5"

bold: True

adaptive\_height: True

MDLabel:

text: app.paidby+" "+app.paymenttime

adaptive\_height: True

theme\_text\_color: "Primary"

MDSeparator:

MDBoxLayout:

id: box\_bottom

adaptive\_height: True

padding: "10dp", 0, 0, 0

MDLabel:

text: app.purpose

adaptive\_height: True

pos\_hint: {"center\_y": .5}

theme\_text\_color: "Primary"

'''

Builder.load\_string(kvstring)

sm = ScreenManager()

Window.size = (700, 600)

class TripotoApp(MDApp):

def build(self):

self.data = {

'Add Transaction': 'Addtrs'

}

self.theme\_cls.theme\_style = "Light"

self.theme\_cls.primary\_palette = "Orange"

sm.add\_widget(HelloScreen(name='helloscreen'))

sm.add\_widget(HomePage(name='homepage'))

sm.current = 'helloscreen'

self.transactionsar = []

return sm

def continue\_to\_app(self):

sm.current = 'homepage'

def submit(self, p1, p2, p3, p4, p5, phn1, phn2, phn3, phn4, phn5, des):

# if p1 and p2 and p3 and p4 and p5 and phn1 and phn2 and phn3 and phn4 and phn5 and des:

self.persons = [p1, p2, p3, p4, p5]

self.phones = [phn1, phn2, phn3, phn4, phn5]

self.destination = des

sm.add\_widget(PostLogin(name='postlogin'))

sm.current = 'postlogin'

sm.remove\_widget(sm.get\_screen('homepage'))

'''else:

Snackbar(

text="Fields cannot be blank",

snackbar\_x="10dp",

snackbar\_y="10dp",

size\_hint\_x=.95

).open()'''

def goback(self):

sm.current = 'postlogin'

def weather(self):

api\_request = requests.get("https://api.openweathermap.org/data/2.5/weather?q="

+ self.destination + "&units=metric&appid=" + 'bb9c06a29f2282c6d4b8606cc12d874c')

api = json.loads(api\_request.content)

# Temperatures

y = api['main']

self.current\_temperature = y['temp']

self.humidity = y['humidity']

self.tempmin = y['temp\_min']

self.tempmax = y['temp\_max']

# Coordinates

x = api['coord']

self.longitude = x['lon']

self.latitude = x['lat']

# Country

z = api['sys']

self.country = z['country']

self.citi = api['name']

'''search = f"weather in {self.destination}"

url=f"https://www.google.com/search?&q={search}"

url2=f"https://in.search.yahoo.com/search?p=weather%20in%20hyderabad"

r=requests.get(url)

s=BeautifulSoup(r.text,"html.parser")

update=s.find("div",class\_="BNeawe").text

self.temp=update

'''

sm.add\_widget(Weather(name='weather'))

sm.current = 'weather'

def addSubmit(self, amount, purpose):

row = [self.personname, amount, purpose, self.category, datetime.now().strftime("%D %H:%M")]

self.transactionsar.append(row)

Snackbar(

text="Transaction added successfully!!",

snackbar\_x="10dp",

snackbar\_y="10dp",

size\_hint\_x=.95

).open()

try:

sm.remove\_widget(sm.get\_screen('overview'))

sm.remove\_widget(sm.get\_screen('transactions'))

except kivy.uix.screenmanager.ScreenManagerException:

pass

sm.add\_widget(Overview(name='overview'))

sm.add\_widget(Transactions(name='transactions'))

self.track()

sm.remove\_widget(sm.get\_screen('addtransaction'))

def track(self):

try:

sm.remove\_widget(sm.get\_screen('overview'))

except kivy.uix.screenmanager.ScreenManagerException:

pass

c = ['Food', 'Utilities', 'Travelling', 'Parties', 'Others']

self.cwisespends = {"Food": 0, "Utilities": 0, "Travelling": 0, "Parties": 0, "Others": 0}

self.spends = {self.persons[0]: 0, self.persons[1]: 0, self.persons[2]: 0, self.persons[3]: 0,

self.persons[4]: 0}

for i in range(len(self.transactionsar)):

for j in range(5):

if self.transactionsar[i][0] == self.persons[j]:

self.spends[self.transactionsar[i][0]] = self.spends[self.transactionsar[i][0]] + int(

self.transactionsar[i][1])

for k in c:

if self.transactionsar[i][3] == k:

self.cwisespends[k] = self.cwisespends[k] + int(self.transactionsar[i][1])

sm.add\_widget(Overview(name='overview'))

sm.current = "overview"

try:

sm.remove\_widget(sm.get\_screen('transactions'))

except kivy.uix.screenmanager.ScreenManagerException:

pass

def transactions(self):

try:

sm.remove\_widget(sm.get\_screen('transactions'))

except kivy.uix.screenmanager.ScreenManagerException:

pass

self.lenn = len(self.transactionsar) \* 90

sm.add\_widget(Transactions(name='transactions'))

sm.current = 'transactions'

try:

sm.remove\_widget(sm.get\_screen('overview'))

except kivy.uix.screenmanager.ScreenManagerException:

pass

for i in range(len(self.transactionsar)):

self.money = self.transactionsar[i][1]

self.paidby = self.transactionsar[i][0]

self.purpose = self.transactionsar[i][2]

self.category = self.transactionsar[i][3]

self.paymenttime = self.transactionsar[i][4]

l = TCard()

sm.get\_screen('transactions').ids.box.add\_widget(l)

def on\_checkbox\_active(self, category):

self.category = category

def on\_person\_select(self, personname):

self.personname = personname

def plus(self, addtrs):

sm.add\_widget(AddTransaction(name='addtransaction'))

sm.current = 'addtransaction'

TripotoApp().run()

**c) TESTING**

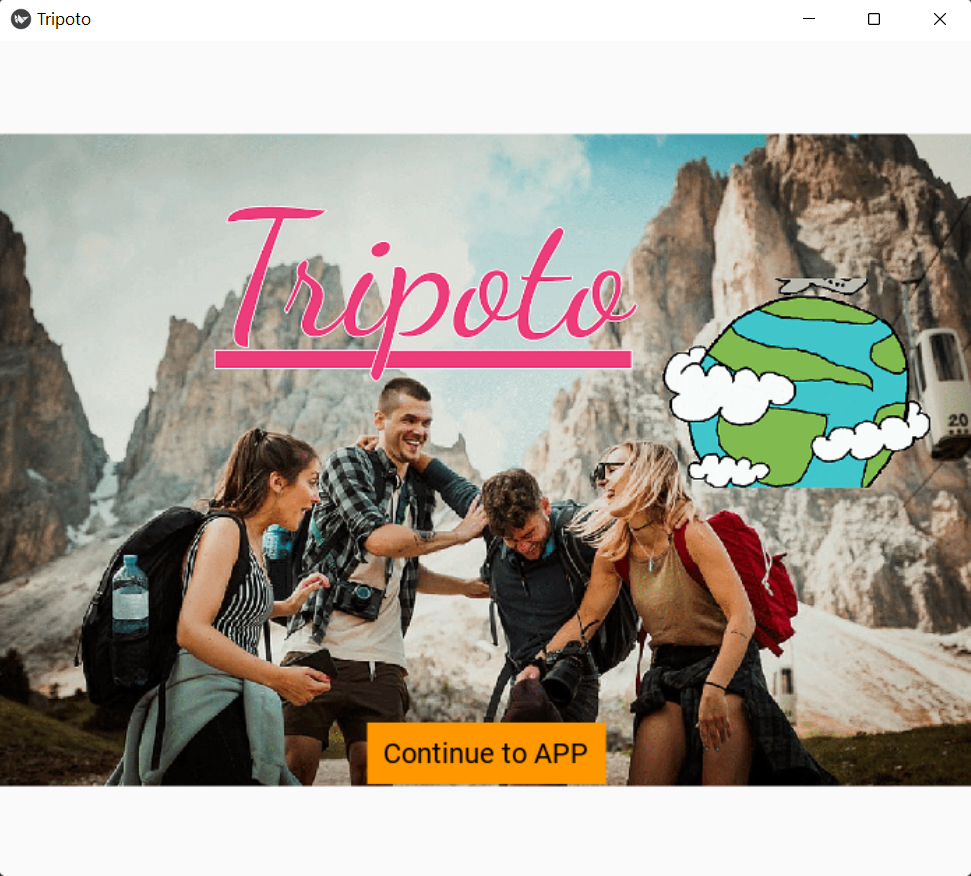
| **Test Case** | | | |
| --- | --- | --- | --- |
| **Test Case Id:** TC01 | | **Use Case ID:** UC01 | |
| **Test Case Title:** Weather | |  | |
| **Test Case Description:**  To Verify weather at different locations. | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1. open the app and enter the destination place.  2. Click on submit.  3. Click on check weather at your place. | The user gets exact weather conditions if he/she enters the correct place. | | If a wrong spelling of destination is given, then it will answer by looking for the word closest to it. |

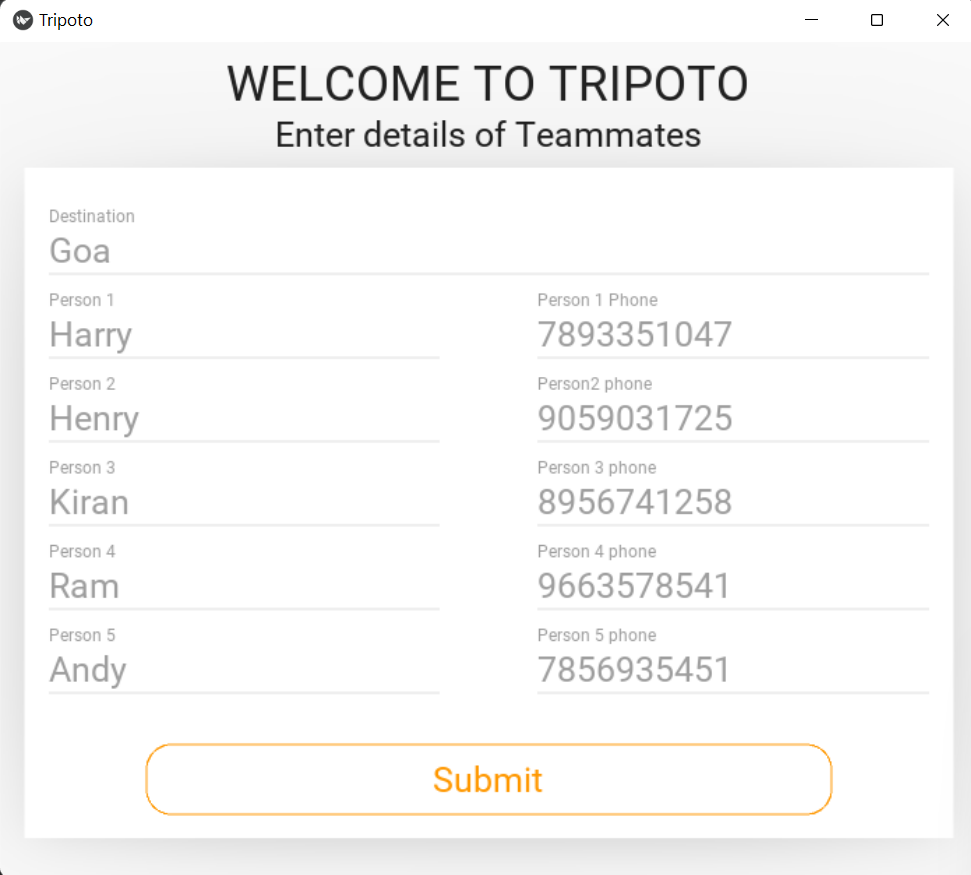
| **Test Case** | | | |
| --- | --- | --- | --- |
| **Test Case Id:** TC02 | | **Use Case ID:** UC02 | |
| **Test Case Title:** Track Expenses | |  | |
| **Test Case Description:**  To track expenses. | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.open the app and enter the destination place and details of travellers.  2.Click on submit.  3.Click on track expenses. | Expenses are added under respective travellers and purpose. | | Expenses are added only under the selected checkboxes. |

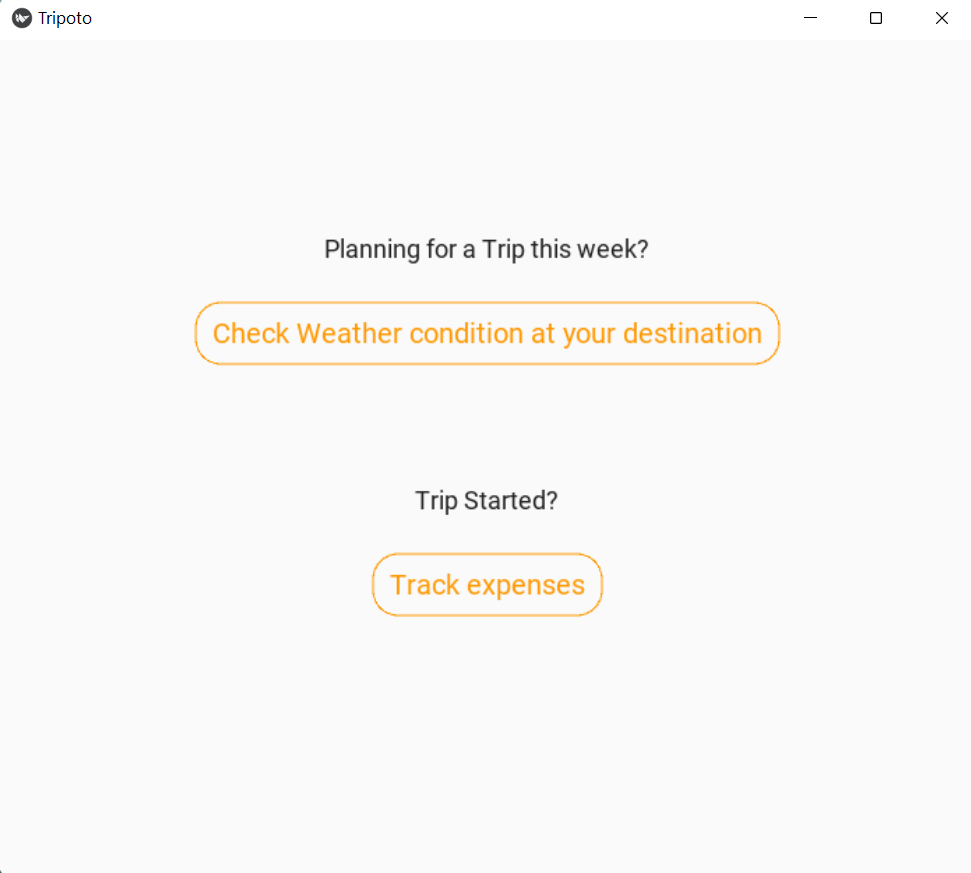
**5. GIT HUB LINK**

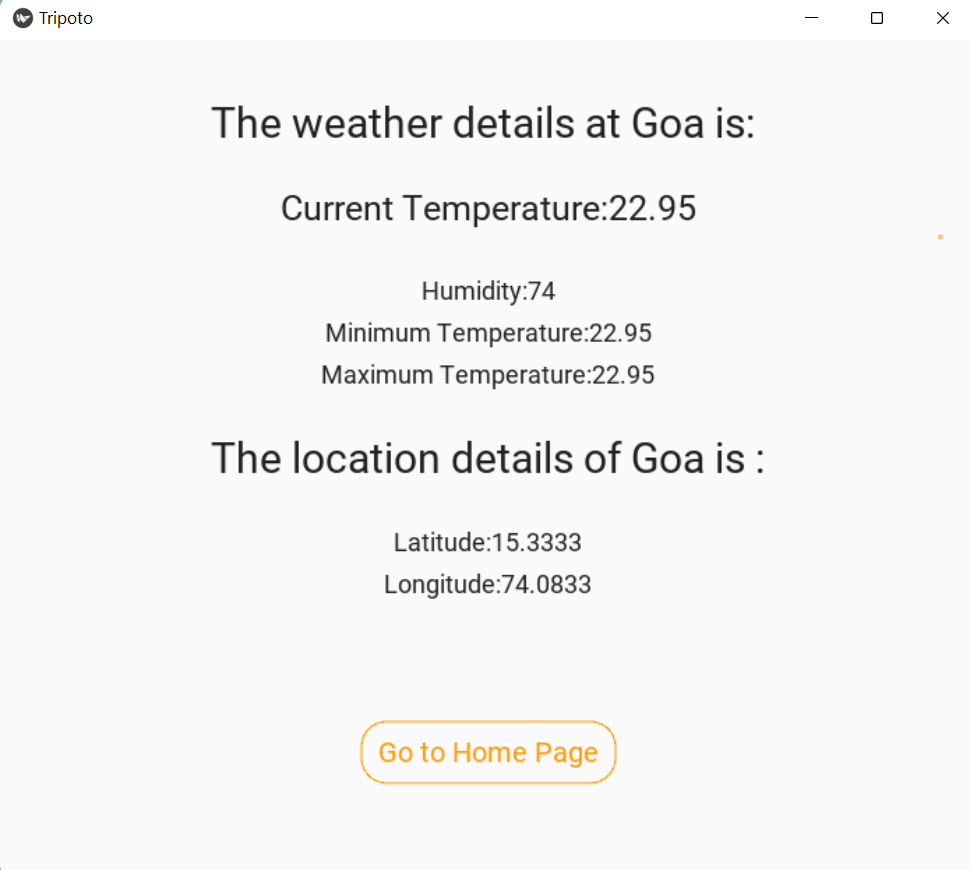
[**https://github.com/HARSHITH-7171/tripoto**](https://github.com/HARSHITH-7171/tripoto)

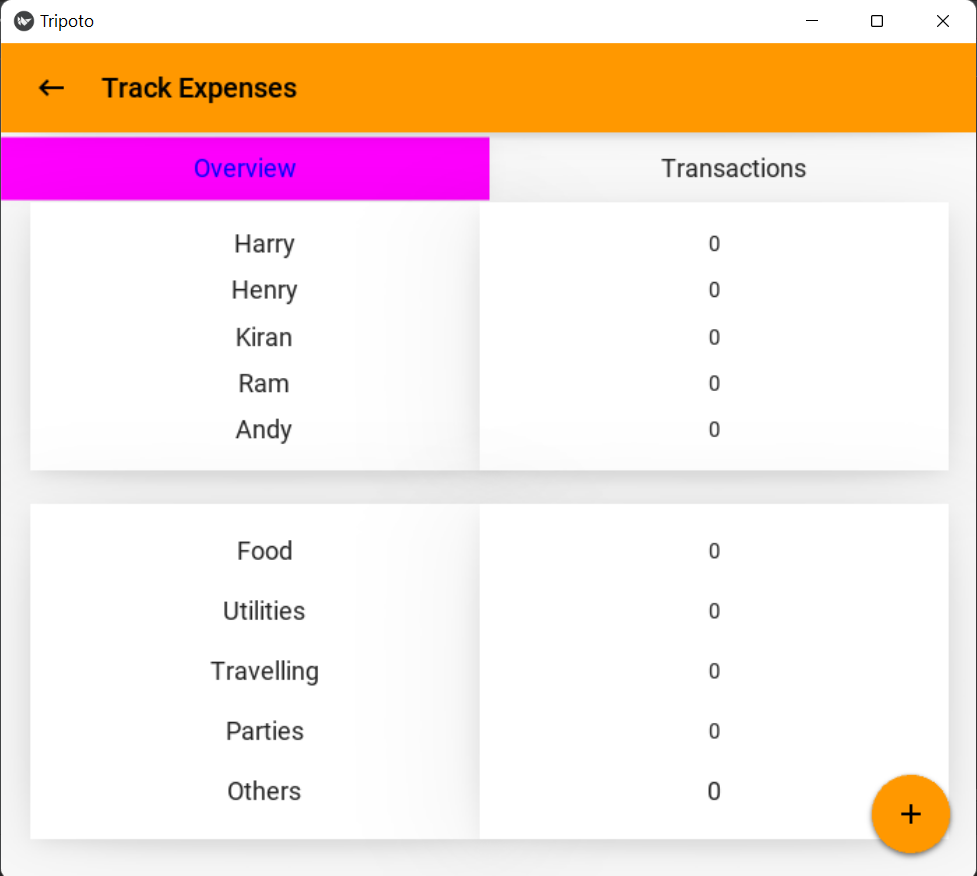
**6. RESULTS**

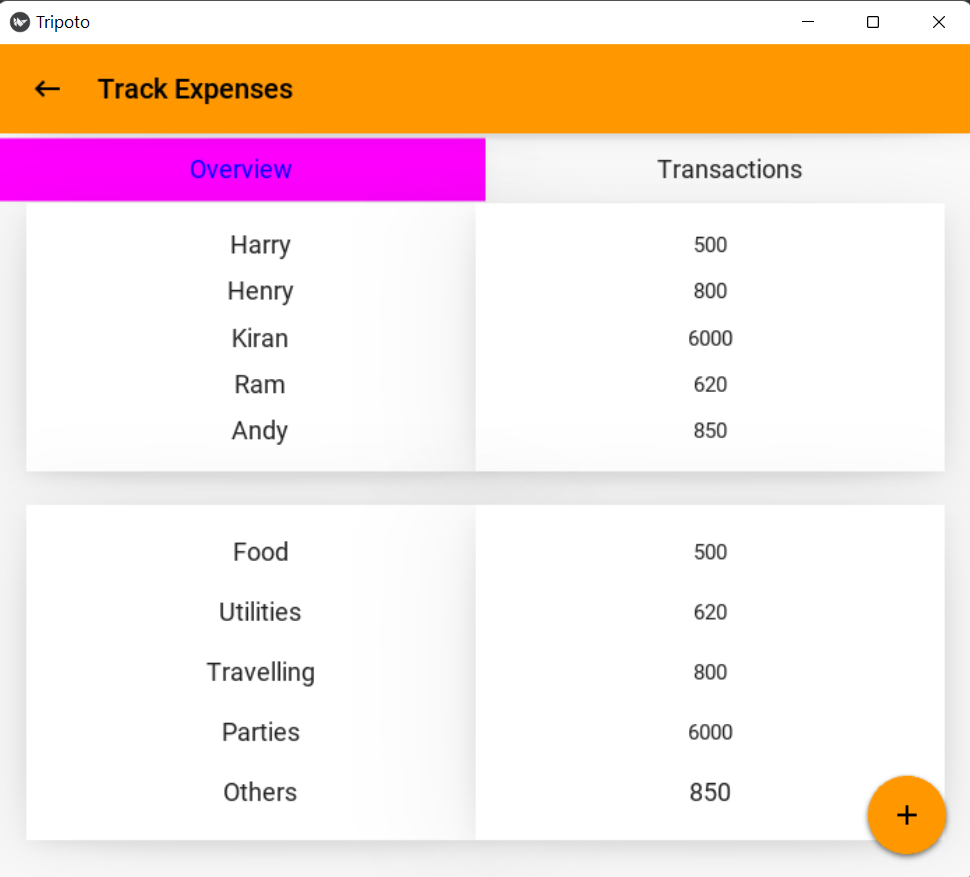
****

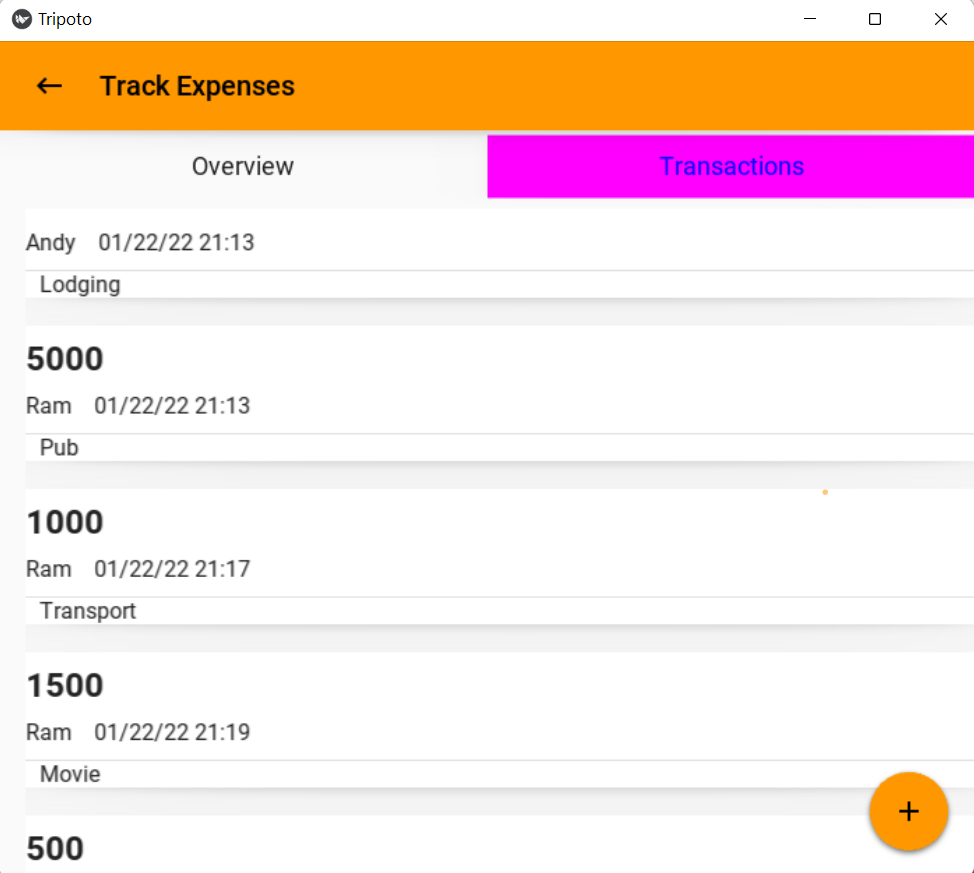
****

****

****

****

****

****

**7. OUTCOMES FROM THE PROJECT APART FROM CURRICULUM**

● We have got to know how to use different kinds of modules.

● We have discovered different kinds of libraries and modules and got to know their implementation and working.

● We have got to know about kivy,bs4,kivymd,json etc .

● We have learnt how to build GUI using kivy framework.

● We have got to know how to handle events in GUI and about the components in them.

**8 a) FUTURE SCOPE**

Reach for this application will be high, as this gives a solution for the travellers to track thie expensed during a trip and also can be used to analyse daily spends (i.e. for domestic use).

In the coming days the app can be improvised by adding some more features like:

* Integrating maps
* Artificial intelligence can be used to show the trends of people travelling and details of average money spent by the travellers.

**8 b) CONCLUSION**

Web scraping is done in order to extract and collect data from websites. Web scraping comes in handy in terms of model development, which requires data to be collected on the fly that's true, relevant to the topic, and accurate. This is desirable as it takes less time compared to implementing datasets. The data that's collected is stored in various formats, such as html, csv, xml, and more, is written to databases for later use, and is also made available online as datasets.

**9. REFERENCES**

<https://www.edureka.co/blog/web-scraping-with-python/>

<https://www.geeksforgeeks.org/create-a-gui-for-weather-forecast-using-openweathermap-api-in-python/>