"TRAVEL BUDDY"

SMART TRAVEL RECOMMENDATION AND TOURISM SUPPORT MOBILE BASED SYSTEM

Project ID : 2023-308

SUPERVISION PERSONALITIES

Supervisor:

Mrs. Thamali Dassanayaka

Lecturer

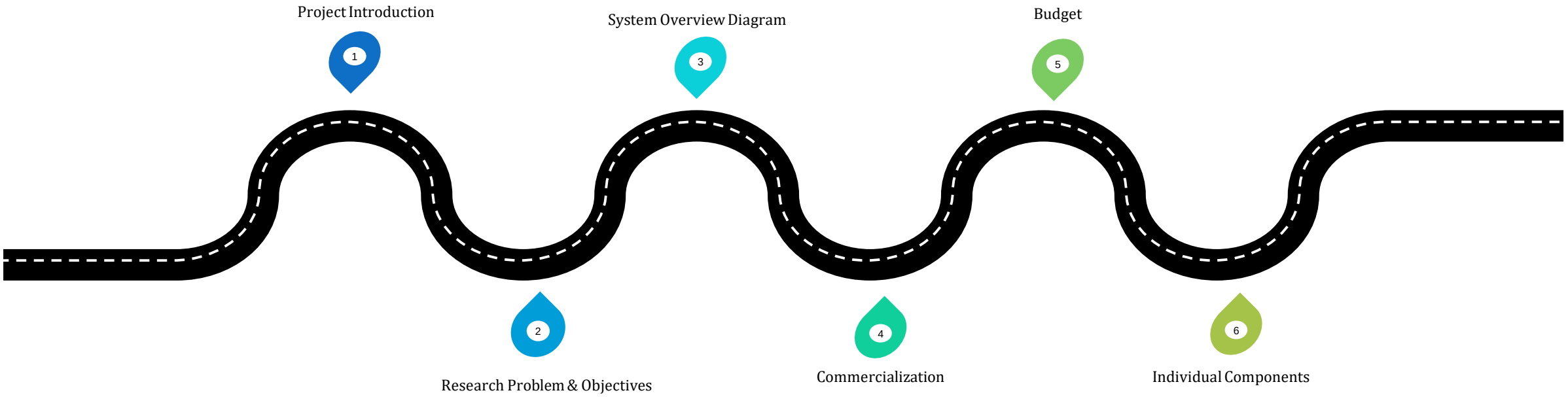
Faculty of Computing | Information Technology

Co-Supervisor:

Dr. Samantha Rajapaksha

Head | Department of Information Technology

ROAD MAP


Project Introduction
1
System Overview Diagram
3
Budget
5
2
Research Problem & Objectives
4
Commercialization
6
Individual Components

INTRODUCTION

- The Smart Travel Recommendation and tourism Support Mobile-Based System is a technology-driven solution that provides users with individualized travel recommendations and tourist support via a mobile application.

- It uses advanced technologies such as natural language processing, machine learning, image processing, emotion analysis, and facial recognition to deliver customized recommendations based on user preferences, location, and emotional state.

RESEARCH PROBLEM

RESEARCH PROBLEM

- Lack of a comprehensive mobile-based system that can provide personalized recommendations and assistance to tourists during their travel.
- Tourist frequently struggle to find acceptable locations and communication barriers.
- There is no efficient system for analyzing user preferences and emotions in order to provide personalized recommendations.
- There is no such a system that can provide information about locations by allowing the system to recognize them and helping to explore new places in an efficient way.

RESEARCH OBJECTIVES

MAIN OBJECTIVE

- The main outcome of the Smart Travel Recommendation and Tourism Support Mobile-Based System is to provide tourists with personalized and real-time recommendations, location discovery, tourist assistance, emotional analysis, and support services to enhance their travel experience in Sri Lanka.

SUB OBJECTIVES

Sub Objective 1: Provide a tourist assistant to guide the tour with emergency support, and location information, and provide correct answers to questions asked by tourists

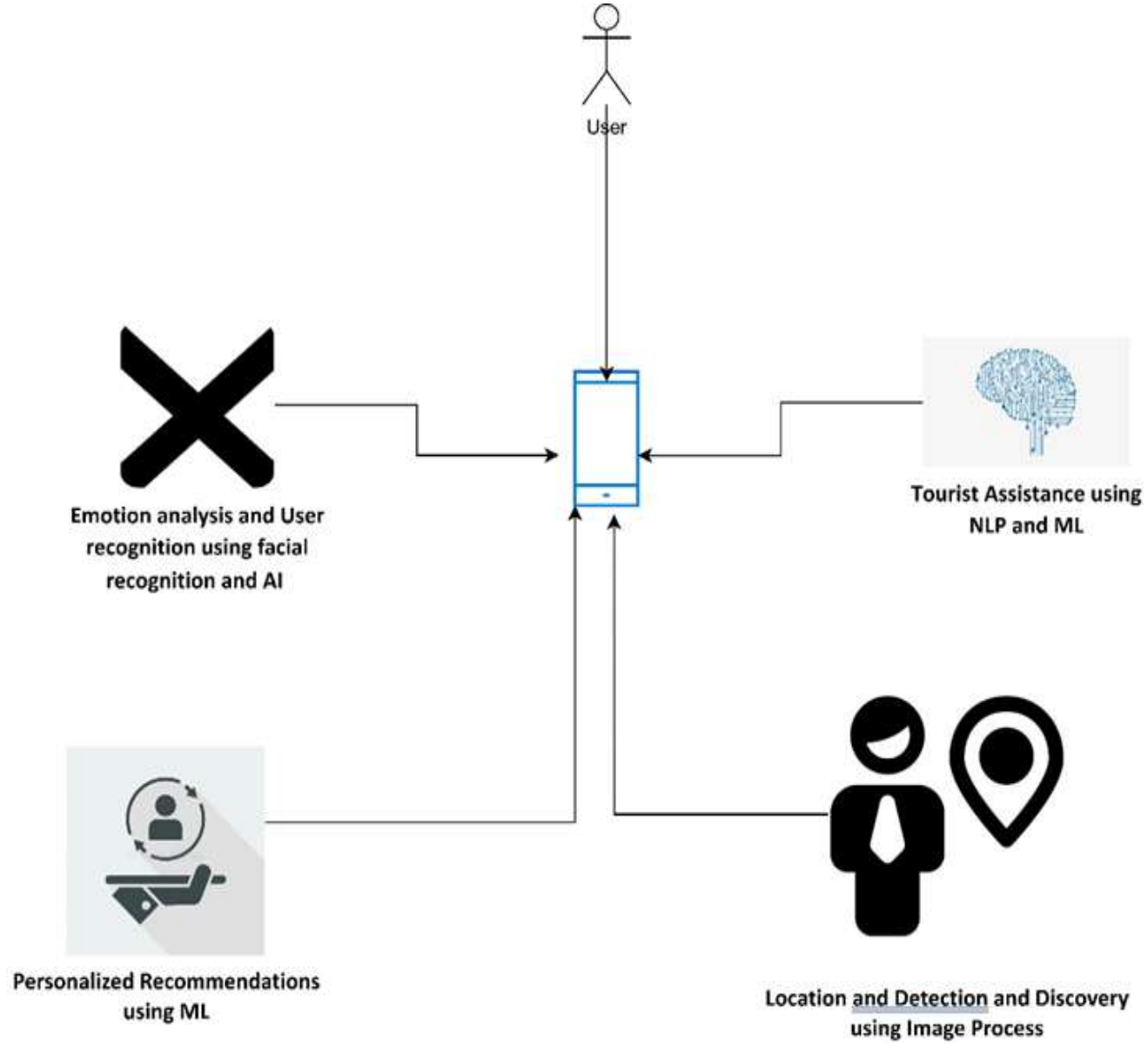
Sub Objective 2 : Provide a customized experience for tourists by providing them with relevant information and options in real-time and making the recommendations more accurate and effective.

SUB OBJECTIVES

Sub Objective 3 : Provide information about locations by allowing the system to recognize them and helping to explore new places and information in an efficient way.

Sub Objective 4 : Accurately identify the emotions of tourists through facial recognition technology and provide personalized recommendations based on the emotional state of the user.

SYSTEM DIAGRAM



COMMERCIALIZATION

LOGO

COMMERCIALIZATION PLAN

- Introduce to Tourism Agency
- Introduce to Foreign Travel Vloggers
- Introduce to the Sri Lanka Tourism Development Authority
- Promote through Social Media
- Publish on Play Store
- Make a Subscription Plan
    - Free Two Days trial period for the new users
    - After the trial period $3.99 per month

BUDGET PLAN

Resource Type	Cost Per unit	Units	**Total Cost (1$=316.00)**
Travel & Accommodation			30,000.00
Internet			10,000.00
Cloud Services			
Server (AWS)	$11 (Per Month)	12 months	48,048.00
Database (AWS RDS)	$8.5 (Per Month)	12 months	37,128.00
Domain	$11.5 (Per Year)		4,186.00
Total in Rs.			**110,362.00**

INDIVIDUAL COMPONENTS

IT20029968 | ATHUKORALA Y.J

B.Sc.(Hons) in Information Technology Specializing in Information Technology

INDIVIDUAL COMPONENT

Developing a tourist assistant chatbot that uses natural language processing and machine learning algorithms to assist tourists in their travels.

RESEARCH QUESTION

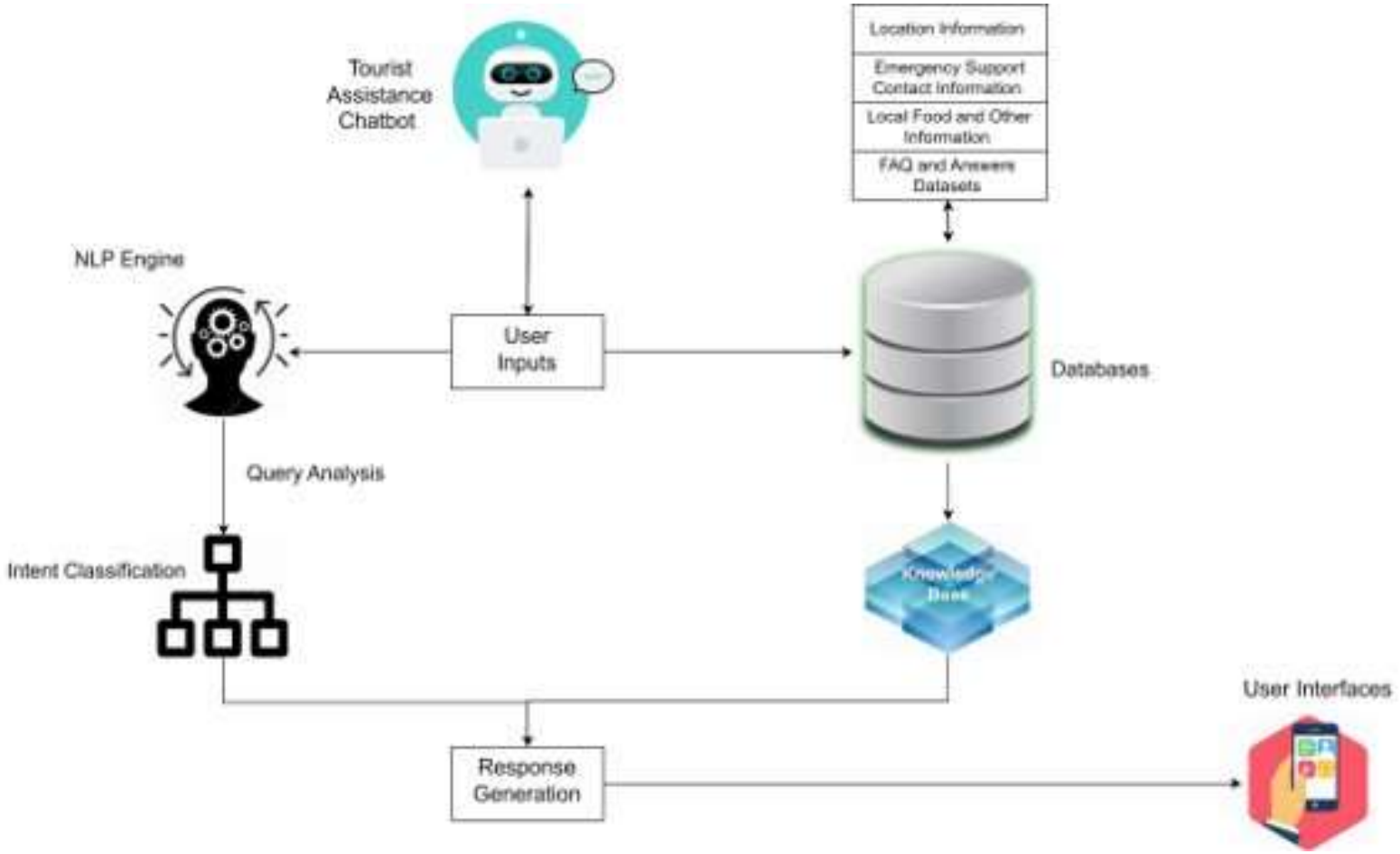
- Many travelers face problems in getting reliable and accurate information and helpful tips during their travels.
- Limited knowledge of the local language, culture, and foods, which can make it difficult for them to communicate with locals and navigate unfamiliar areas.
- Guidebooks or online travel websites, may not provide relevant information about Sri Lanka for tourists.
- Lack of knowledge about emergency support contact information, travel location information, local foods, and travel tips.

SPECIFICS AND SUB OBJECTIVES

Provide a tourist assistant to guide the tour with, emergency support contact information, location information, local food information, provide answers to tourist's frequently asked questions

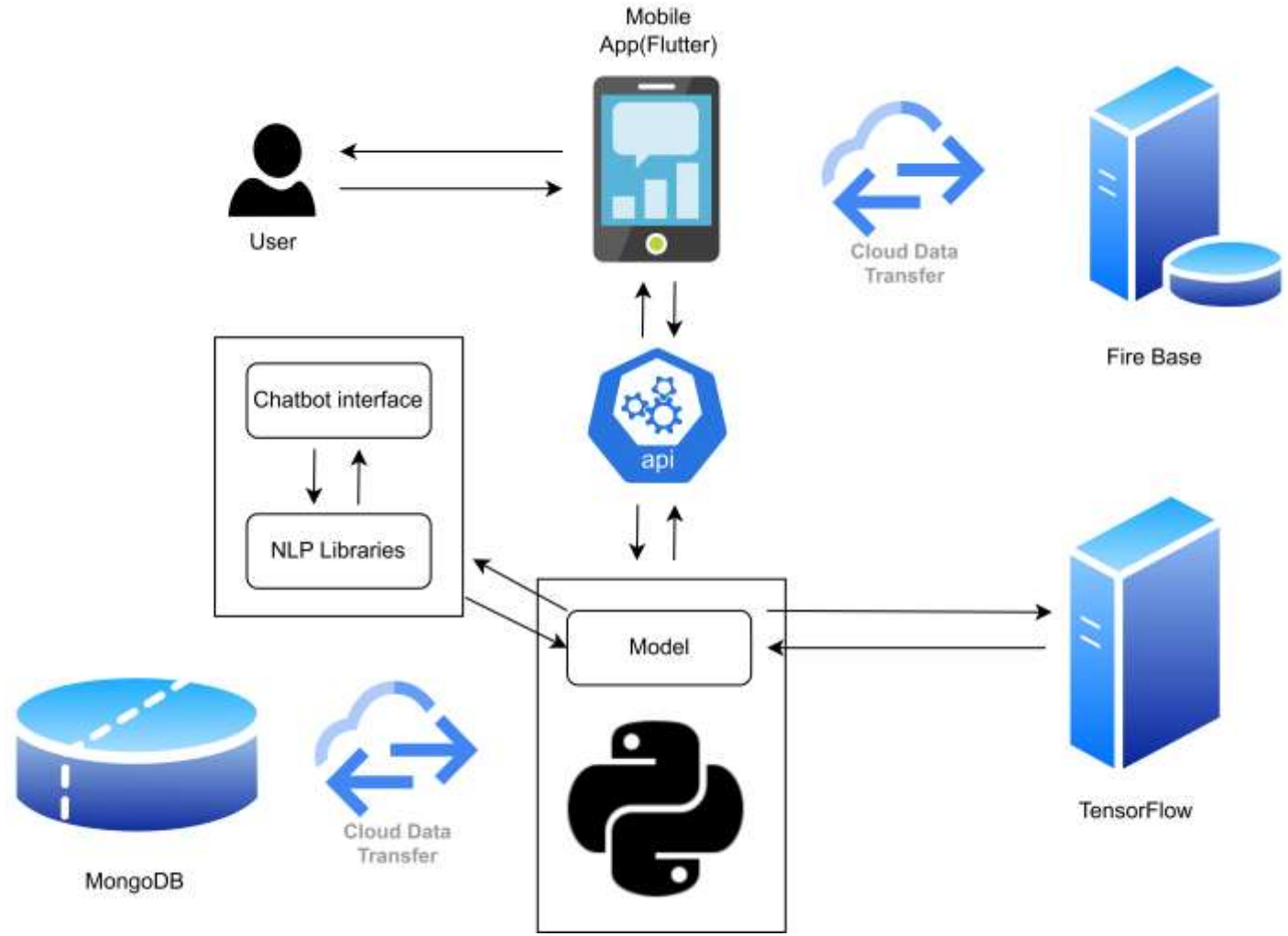
METHODOLOGY

System Diagram:



METHODOLOGY

System Overview Diagram 2 (Based on Technologies):



Methodology

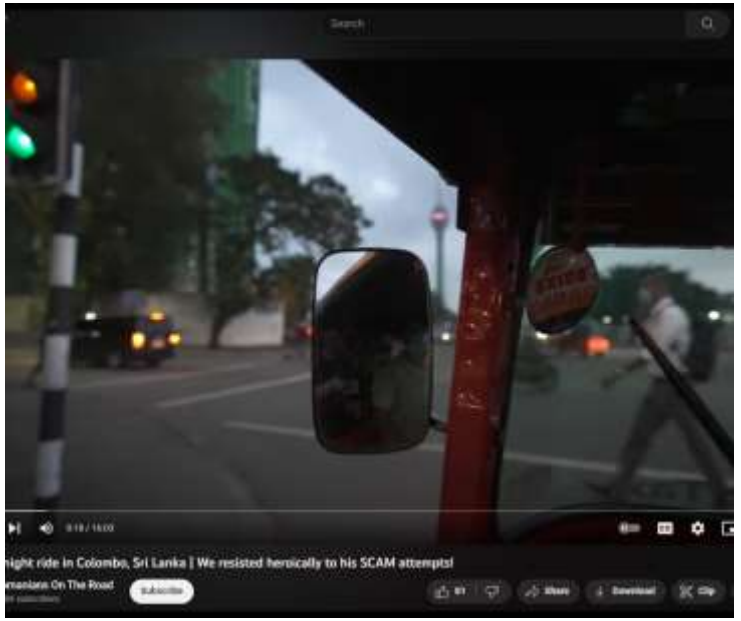
- Gather Location, food, culture, scams information
- Creating questions and relevant answers based on collected the data and information.
- Creating a training data set and testing data set
- Put sampling values to the data set
- Create and compile the model

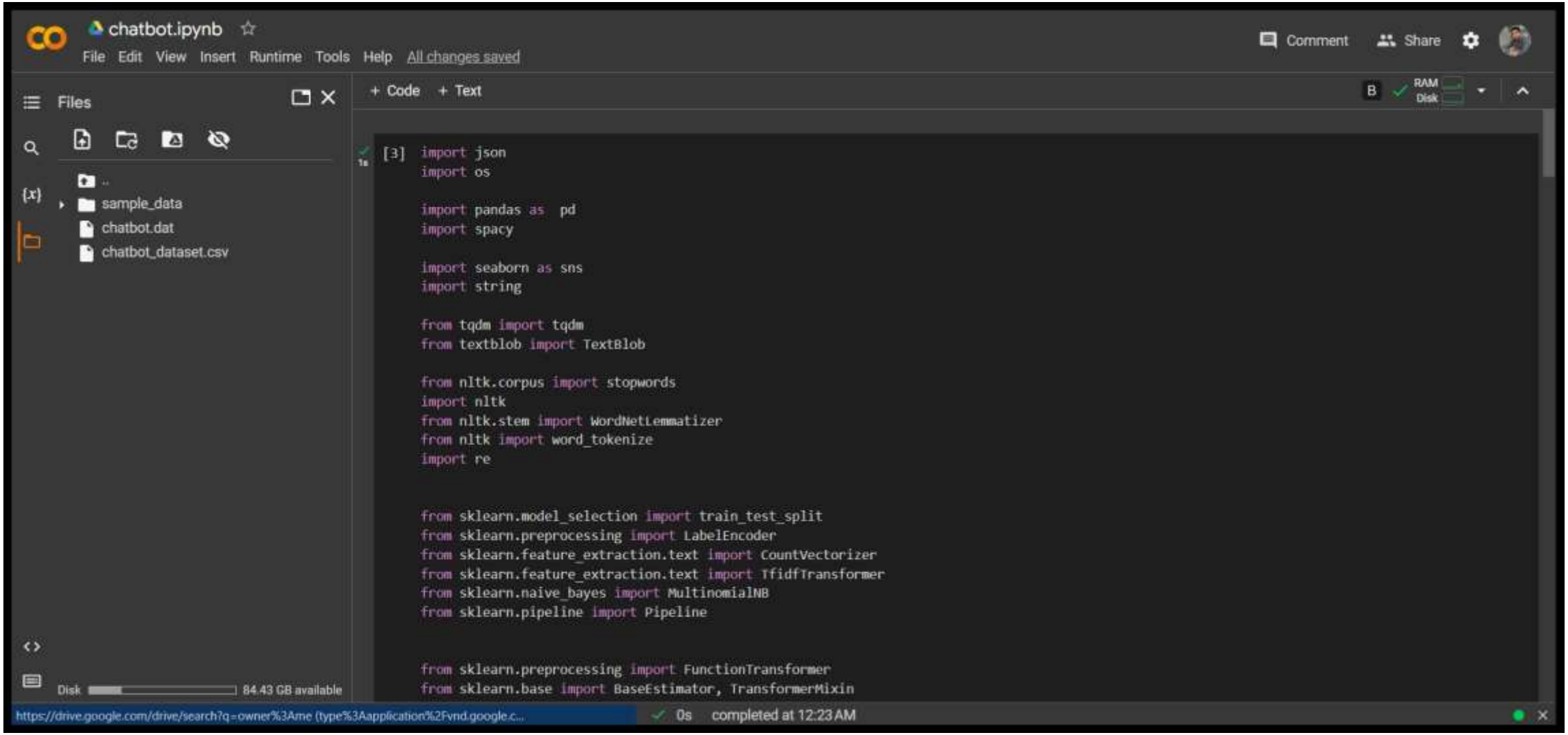
Technology Stack

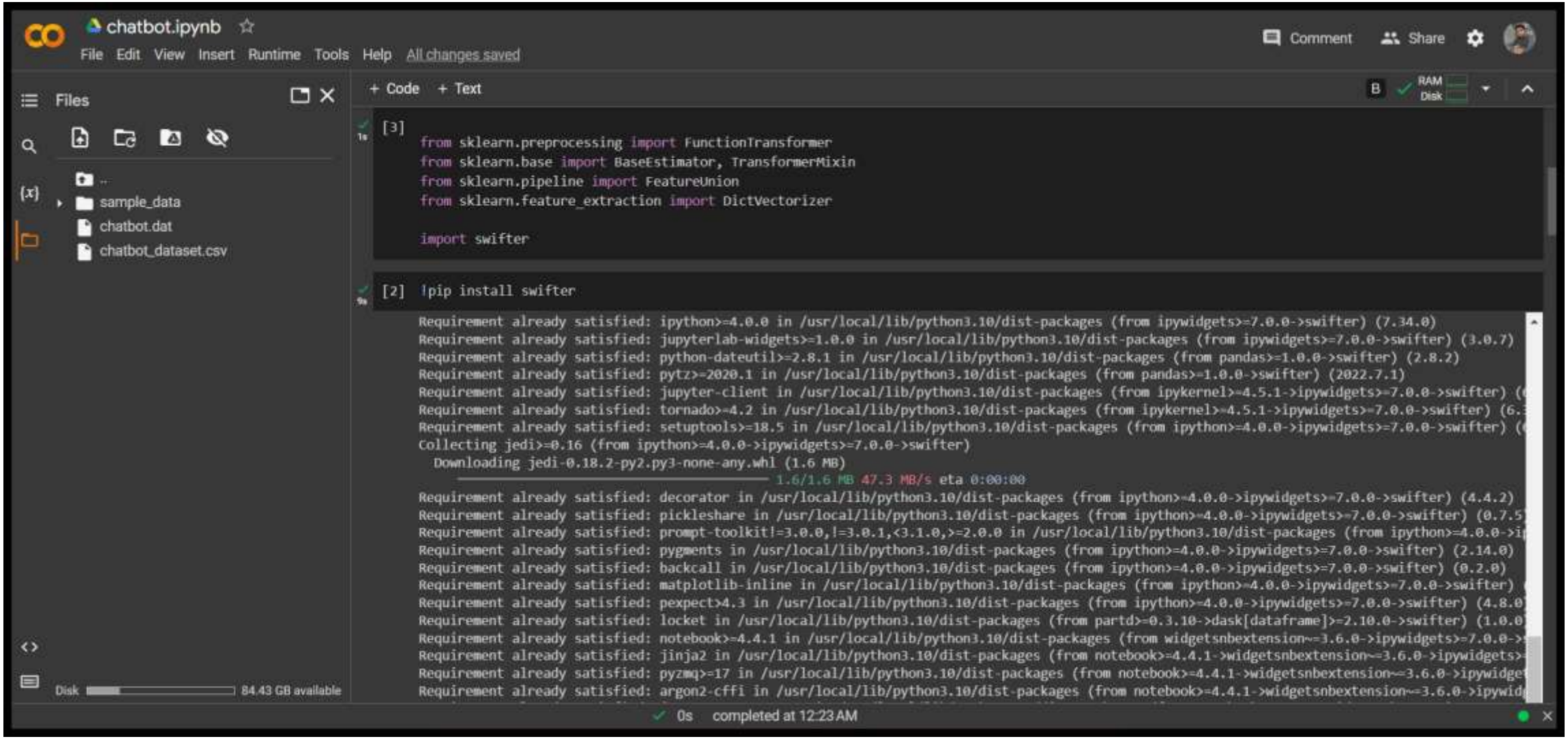
- Python
- Flutter
- Firebase
- MongoDB
- TensorFlow

DEMONSTRATION

BACKGROUND STUDY AND DATA COLLECTION

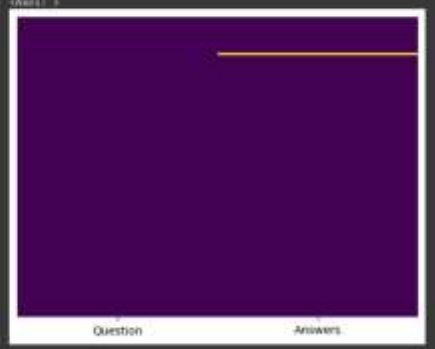



chatbot.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code
+ Text
RAM
Disk
sample_data
chatbot.dat
chatbot_dataset.csv
[3] import json
import os
import pandas as pd
import spacy
import seaborn as sns
import string
from tqdm import tqdm
from textblob import TextBlob
from nltk.corpus import stopwords
import nltk
from nltk.stem import WordNetLemmatizer
from nltk import word_tokenize
import re
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import FunctionTransformer
from sklearn.base import BaseEstimator, TransformerMixin
Disk
84.43 GB available
0s completed at 12:23 AM


chatbot.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
RAM
Disk
Files
+ Code
+ Text
sample_data
chatbot.dat
chatbot_dataset.csv
[3]
from sklearn.preprocessing import FunctionTransformer
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.pipeline import FeatureUnion
from sklearn.feature_extraction import DictVectorizer
import swifter
[2] !pip install swifter
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets>=7.0.0->swifter) (7.34.0)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets>=7.0.0->swifter) (3.0.7)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->swifter) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->swifter) (2022.7.1)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.10/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter)
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.10/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter)
Collecting jedi>=0.16 (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter)
Downloading jedi-0.18.2-py2.py3-none-any.whl (1.6 MB)
1.6/1.6 MB 47.3 MB/s eta 0:00:00
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (0.7.5)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (2.14.0)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (0.2.0)
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (4.8.0)
Requirement already satisfied: locket in /usr/local/lib/python3.10/dist-packages (from partd>=0.3.10->dask[dataframe]>=2.10.0->swifter) (1.0.0)
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.10/dist-packages (from widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->swifter)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->swifter)
Requirement already satisfied: pyzmq>=17 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->swifter)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->swifter)
Disk
84.43 GB available
0s completed at 12:23 AM

chatbot.ipynb

```
[5] df = pd.read_csv('chatbot_dataset.csv', encoding='latin-1')

[6] sns.heatmap(df.isnull(), yticklabels=False, cbar=False, cmap='viridis')
```




```
[7] df=df.fillna("null_value")
```

chatbot.ipynb

```
[7] df=df.fillna("null_value")

[8] df.head()
```

	Question	Answers
0	Hi	Hi, How can I help you?
1	Hello	Hi Buddy, How can I help you?
2	I'm in colombo now	Nice. There are many amazing places you can vi...
3	What are the best places I can visit in colombo?	Gangaramaya Temple, Galle Face Green, Mount La...
4	What are the places I can visit in galle?	Galle Fort, Sea Turtle Farm Galle Mahamodara, ...

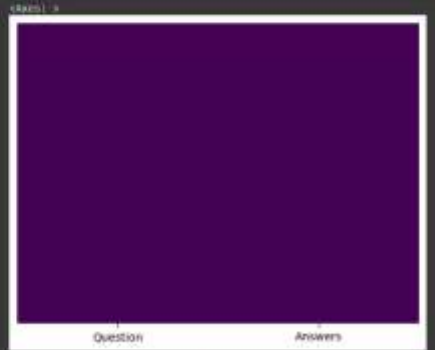
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 116 entries, 0 to 115
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Question  116 non-null    object
 1   Answers   116 non-null    object
dtypes: object(2)
memory usage: 1.9+ KB
```

```
[10] df.columns
```

chatbot.ipynb

```
[10] df.columns

Index(['Question', 'Answers'], dtype='object')

sns.heatmap(df.isnull(), yticklabels=False, cbar=False, cmap='viridis')
```



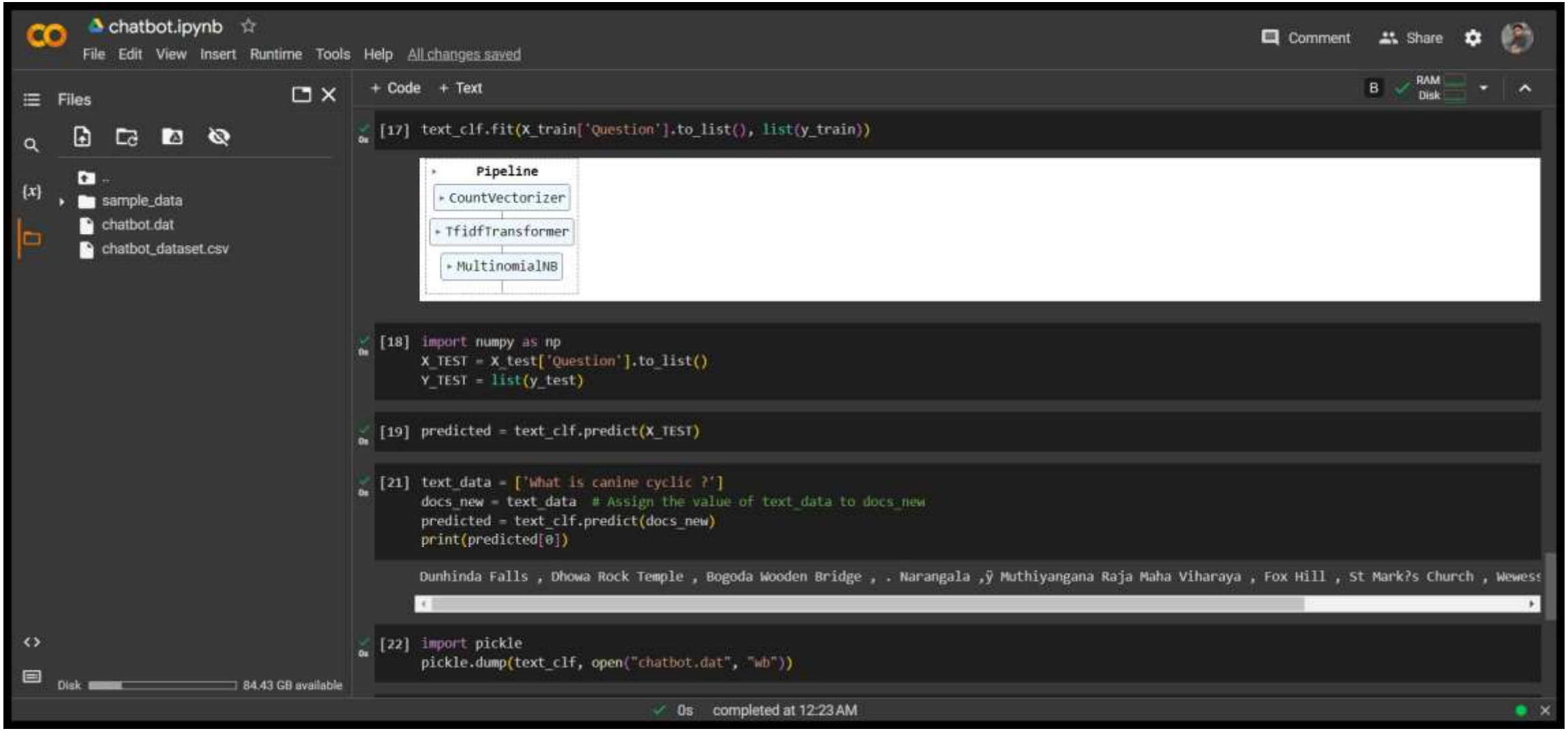
chatbot.ipynb

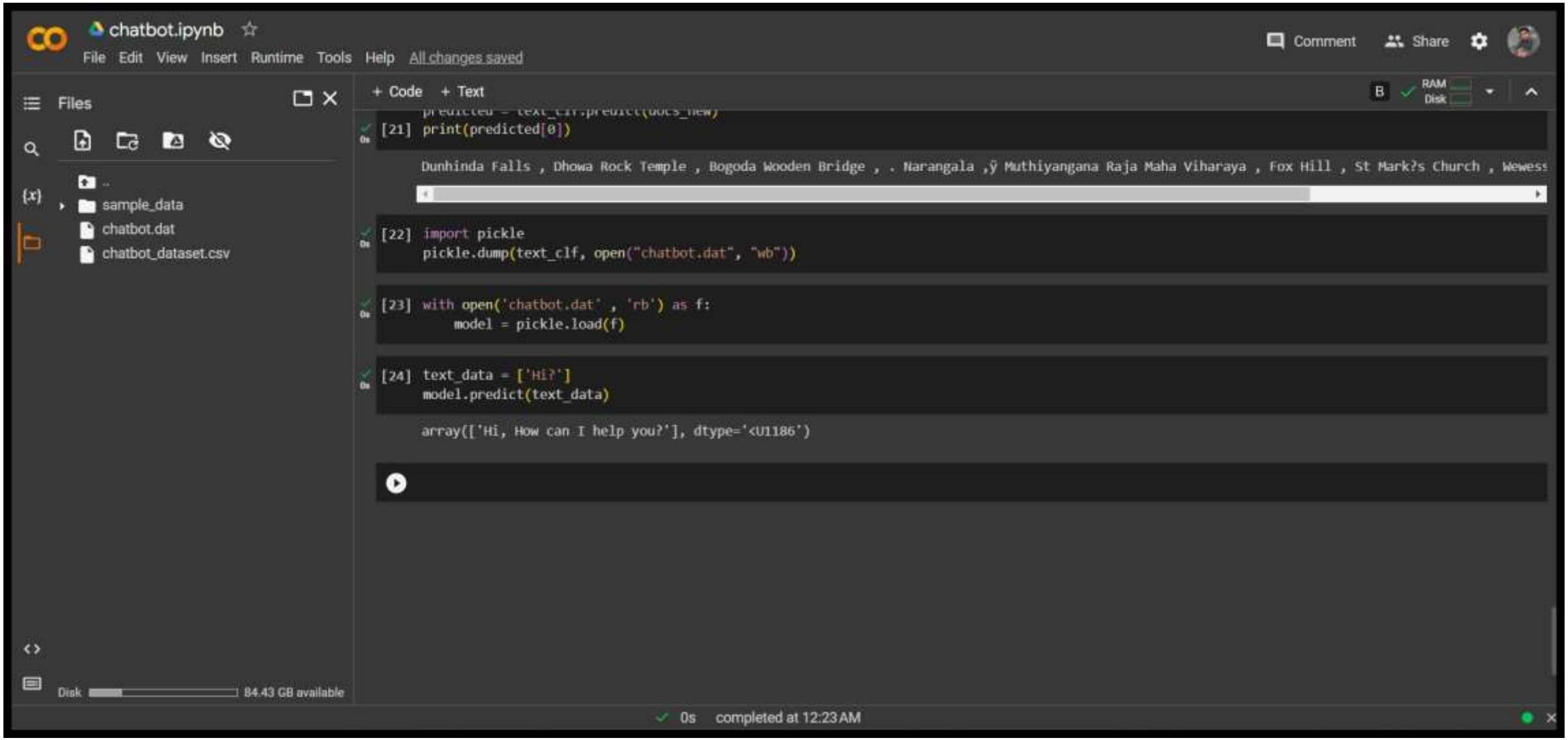
```
[12] df.isna().sum()
```

```
Question    0
Answers     0
dtype: int64
```

```
[13] X = df.drop('Answers', axis=1)
     y = df['Answers']

[14] from sklearn.model_selection import train_test_split

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)

[15] from sklearn.feature_extraction.text import TfidfVectorizer
     tf = TfidfVectorizer()

[16] text_clf = Pipeline([
       ('vect', CountVectorizer(analyzer='word', stop_words='english')),
       ('tfidf', TfidfTransformer(use_idf=True)),
       ('clf', MultinomialNB(alpha=.01)),
     ])

[17] text_clf.fit(X_train['Question'].to_list(), list(y_train))
```

```
Pipeline
  CountVectorizer
```

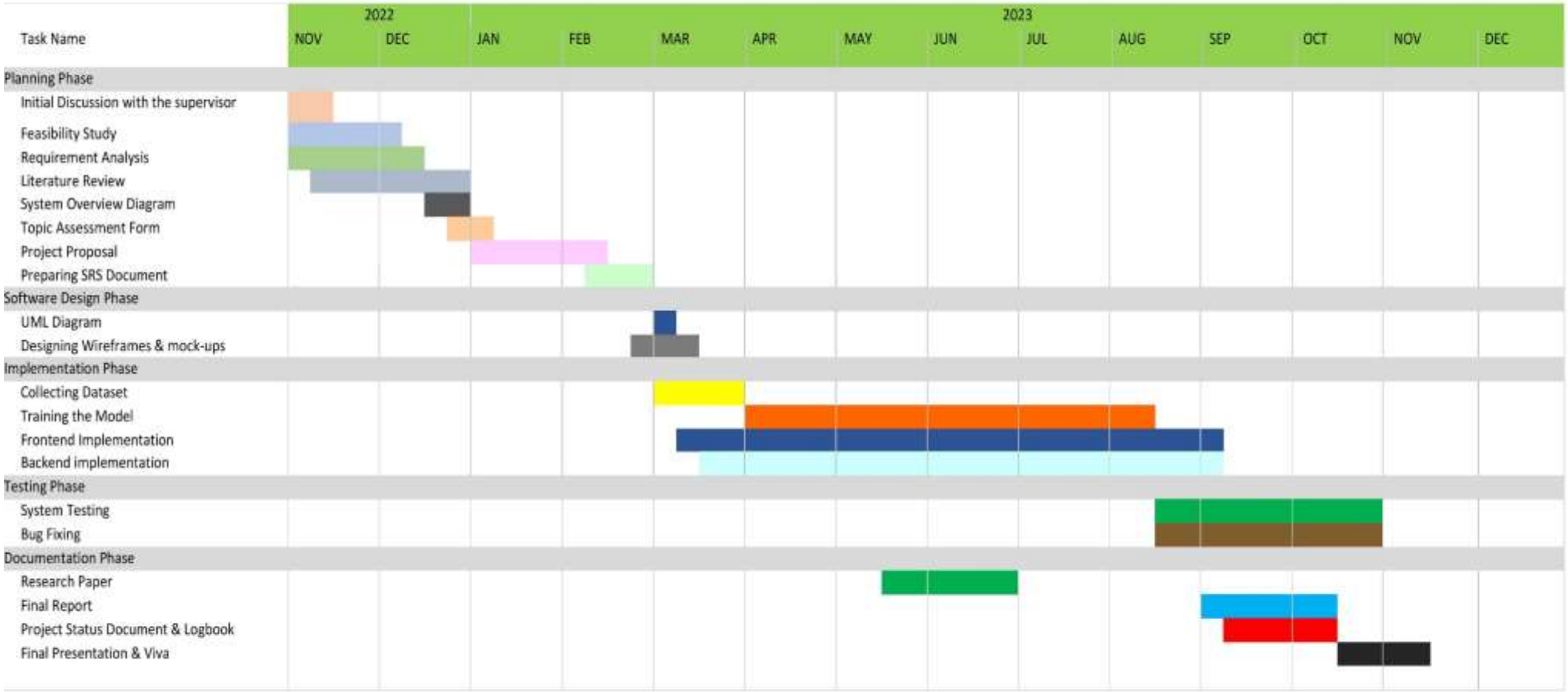

chatbot.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code
+ Text
RAM
Disk
sample_data
chatbot.dat
chatbot_dataset.csv
[17] text_clf.fit(X_train['Question'].to_list(), list(y_train))
Pipeline
CountVectorizer
TfidfTransformer
MultinomialNB
[18] import numpy as np
X_TEST = X_test['Question'].to_list()
Y_TEST = list(y_test)
[19] predicted = text_clf.predict(X_TEST)
[21] text_data = ['What is canine cyclic ?']
docs_new = text_data # Assign the value of text_data to docs_new
predicted = text_clf.predict(docs_new)
print(predicted[0])
Dunhinda Falls , Dhowa Rock Temple , Bogoda Wooden Bridge , . Narangala ,ỹ Muthiyangana Raja Maha Viharaya , Fox Hill , St Mark?s Church , Wewess
[22] import pickle
pickle.dump(text_clf, open("chatbot.dat", "wb"))
Disk
84.43 GB available
0s completed at 12:23 AM


chatbot.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code
+ Text
RAM
Disk
[21] print(predicted[0])
Dunhinda Falls , Dhowa Rock Temple , Bogoda Wooden Bridge , . Narangala ,ÿ Muthiyangana Raja Maha Viharaya , Fox Hill , St Mark?s Church , Wewess
[22] import pickle
pickle.dump(text_clf, open("chatbot.dat", "wb"))
[23] with open('chatbot.dat' , 'rb') as f:
model = pickle.load(f)
[24] text_data = ['Hi?']
model.predict(text_data)
array(['Hi, How can I help you?'], dtype='<U1186')
sample_data
chatbot.dat
chatbot_dataset.csv
Disk
84.43 GB available
0s completed at 12:23 AM

TASK TO BE DONE

- Fine tune my component with other components
- Develop an mobile application with a user-friendly User Interfaces
- Add more data to datasets

GANTT CHART


Task Name
2022
2023
NOV
DEC
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
Planning Phase
Initial Discussion with the supervisor
Feasibility Study
Requirement Analysis
Literature Review
System Overview Diagram
Topic Assessment Form
Project Proposal
Preparing SRS Document
Software Design Phase
UML Diagram
Designing Wireframes & mock-ups
Implementation Phase
Collecting Dataset
Training the Model
Frontend Implementation
Backend implementation
Testing Phase
System Testing
Bug Fixing
Documentation Phase
Research Paper
Final Report
Project Status Document & Logbook
Final Presentation & Viva

REFERENCES

[1] S. L. T. D. Authority, "Sri Lanka Tourism Development Authority," [Online]. Available: https://www.sltda.gov.lk/.

[2] K.K.D.N. Dilshan, C.A.J.P. Chandranath, U.M.D.M. Parussella, Samantha Thelijjagoda, H.M.C.J. Herath and Thilini Jayalath, "JESSY: An Intelligence Travel Assistant," 2021.

[3] Chen, W., Cheng, M. and Huang, Y., "Personalized Tourism Attraction Recommendation Using Natural Language Processing and Machine Learning," no. 2019, 2019.

[4] Zhang, X., Cao and J., " Chatbot technology in tourism services: A study of voice and text-based systems," 2018.

[5] Jiang, S., Wu, B., Li and H., "Tourist chatbot: An AI-based tool for improving tourist," Journal of Travel Research, 2019.

[6] Sthapit, E., Basnet, C. R., Singh and R., "Developing a chatbot using natural language processing to assist tourists in Nepal," Journal of Tourism and Hospitality Management, 2020.

[7] Haddara, M., Hamed and S., "Enhancing the accuracy of tourist information retrieval by natural language processing," Journal of Hospitality and Tourism Technology, 2018.

[8] S. A. Kaumalee Bogahawatte, "Intelligent Criminal Identification System," p. 06, 2013.

[9] C. S. S. Isuru Jayaweera, "Crime Analytics: Analysis of Crimes Through," p. 06. [10] M. K. R. S. O. Christian Sunday Nwankwo, "APPLICATION OF DATA ANALYTICS TECHNIQUES IN ANALYZING," p. 08, 2018.

REFERENCES

[1] S. L. T. D. Authority, "Sri Lanka Tourism Development Authority," [Online]. Available: https://www.sltda.gov.lk/.

[2] K.K.D.N. Dilshan, C.A.J.P. Chandranath, U.M.D.M. Parussella, Samantha Thelijjagoda, H.M.C.J. Herath and Thilini Jayalath, "JESSY: An Intelligence Travel Assistant," 2021.

[3] Chen, W., Cheng, M. and Huang, Y., "Personalized Tourism Attraction Recommendation Using Natural Language Processing and Machine Learning," no. 2019, 2019.

[4] Zhang, X., Cao and J., " Chatbot technology in tourism services: A study of voice and text-based systems," 2018.

[5] Jiang, S., Wu, B., Li and H., "Tourist chatbot: An AI-based tool for improving tourist," Journal of Travel Research, 2019.

[6] Sthapit, E., Basnet, C. R., Singh and R., "Developing a chatbot using natural language processing to assist tourists in Nepal," Journal of Tourism and Hospitality Management, 2020.

[7] Haddara, M., Hamed and S., "Enhancing the accuracy of tourist information retrieval by natural language processing," Journal of Hospitality and Tourism Technology, 2018.

[8] S. A. Kaumalee Bogahawatte, "Intelligent Criminal Identification System," p. 06, 2013.

[9] C. S. S. Isuru Jayaweera, "Crime Analytics: Analysis of Crimes Through," p. 06. [10] M. K. R. S. O. Christian Sunday Nwankwo, "APPLICATION OF DATA ANALYTICS TECHNIQUES IN ANALYZING," p. 08, 2018.

[10] Karl Rock (New Zealand) Catfishing Scammers in Sri Lanka: https://www.youtube.com/watch?v=TJY-vcYczjc

[11]

IT20051020 | SHAMINDA W.G.T.

B.Sc.(Hons) in Information Technology Specializing in Information Technology

INDIVIDUAL COMPONENT

Developing a system that recognize historical places, monuments, tourist attractive places, destinations through smart phone camera and provide relevant details about them. In current situations of Sri Lanka this system will replace as tourist guide service and very easy to handle

RESEARCH QUESTION

- There are many apps to identify many objects. But in tourism industry there are no accurate app to identify places, tourism attractive places.
- The assistance of a guide is essential when a foreigner wants to know more about places in Sri Lanka.
- Another problem facing foreign tourists is the lack of tour guides due to the economic crisis in Sri Lanka.
- Difficulty in providing guidance to all tourists visiting Sri Lanka is a problem for tour guides
- Is the tourist guide school 100% true? Can they be trusted?
- For these reasons, the location/places identification system shows the need.

SPECIFICS AND SUB OBJECTIVES

Specific Objective

Detect Places, and analysis them using machine learning and image processing

Sub Objective

- Collecting data sets
- Creating data sets into normalized form
- Taking photos and videos
- Identifying places
- Add relevant details to places
- Create a ML model to identify place will commit a places/locations based on get discovered datasets
- Identifying the best algorithm for place/ detection.
- Fine tuning and testing of analyzing model to increase accuracy of the results.

METHODOLOGY

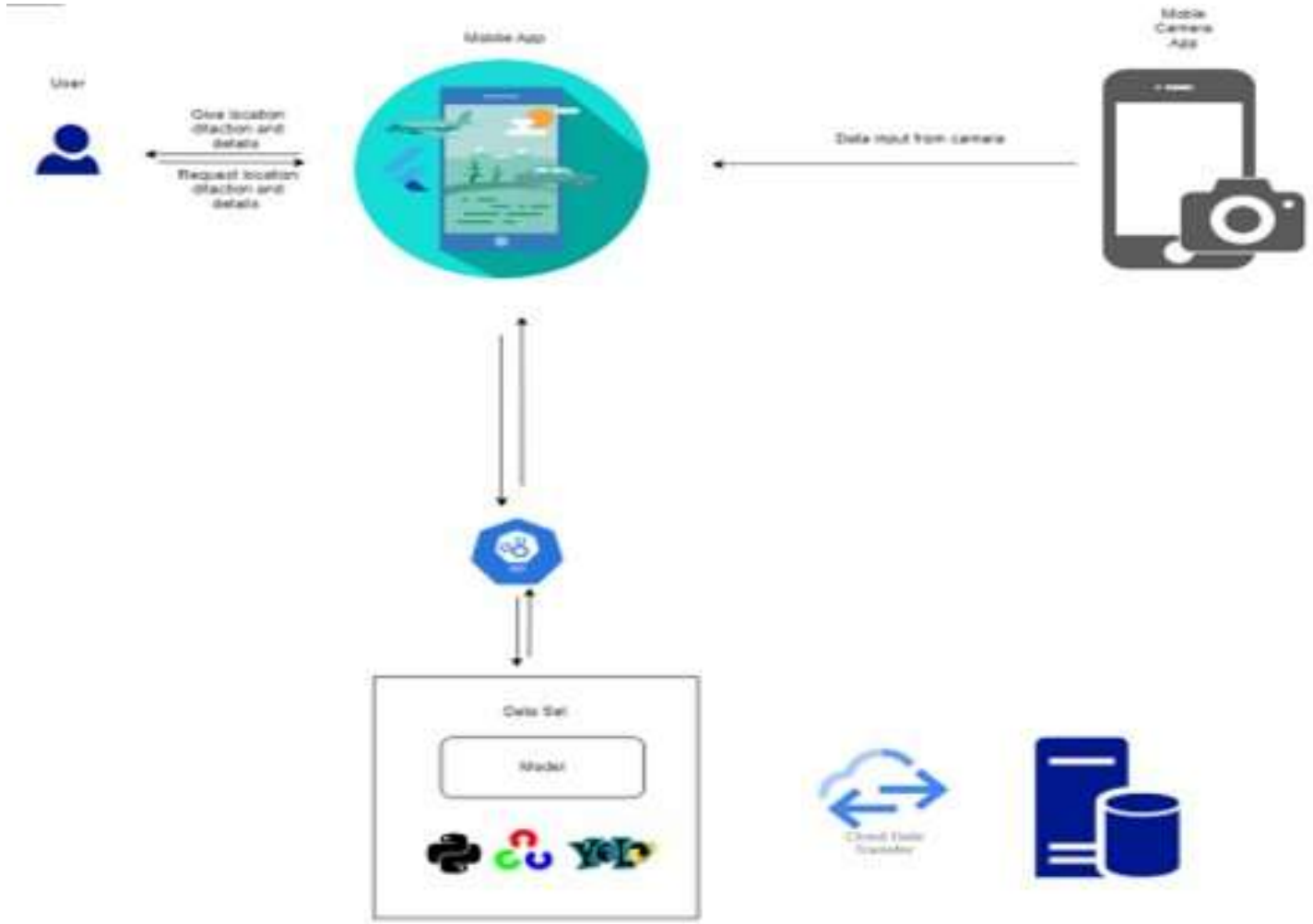
- Collect data about the historical places, monuments, and tourist destinations that the system will recognize. This can include images, video, and text descriptions. You can gather this information from various sources, such as government websites, travel blogs, and tourist guides.

- Once collected the data, need to train the system to recognize the historical places, monuments, and tourist destinations. This involves using machine learning algorithms to identify patterns and features in the images and other data. Use existing machine learning frameworks, such as TensorFlow or Keras, to train the system.

- After developed the app, need to test it thoroughly to ensure that it works as intended.

METHODOLOGY

System Diagram:

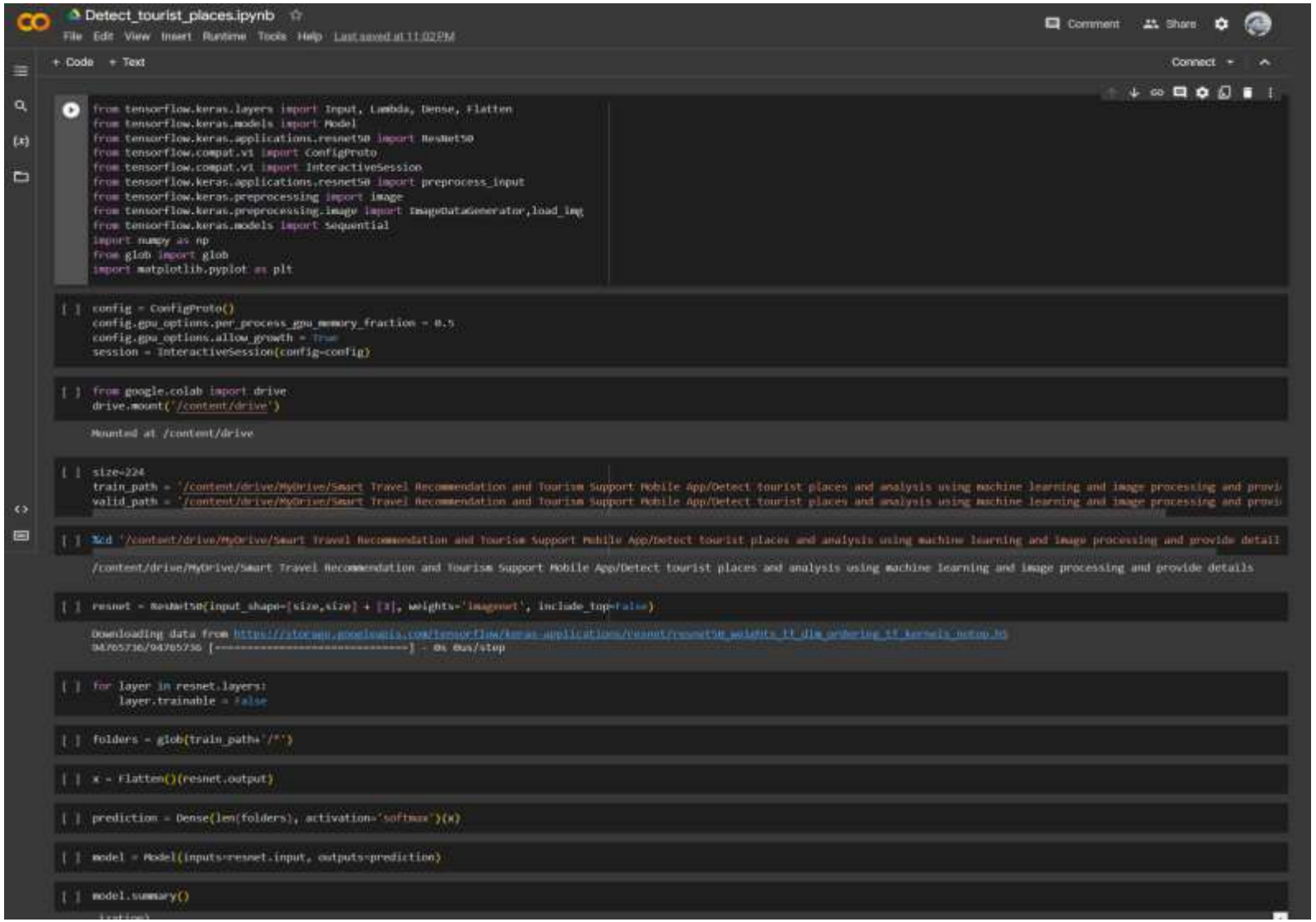


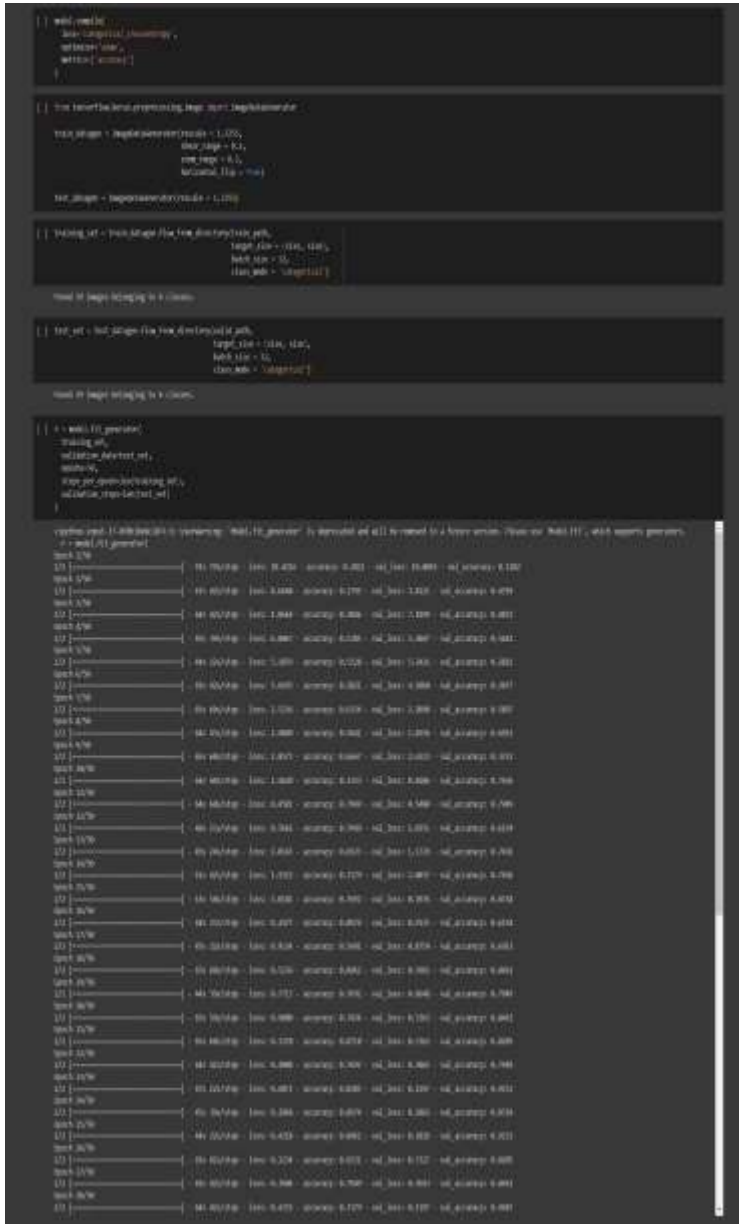
METHODOLOGY

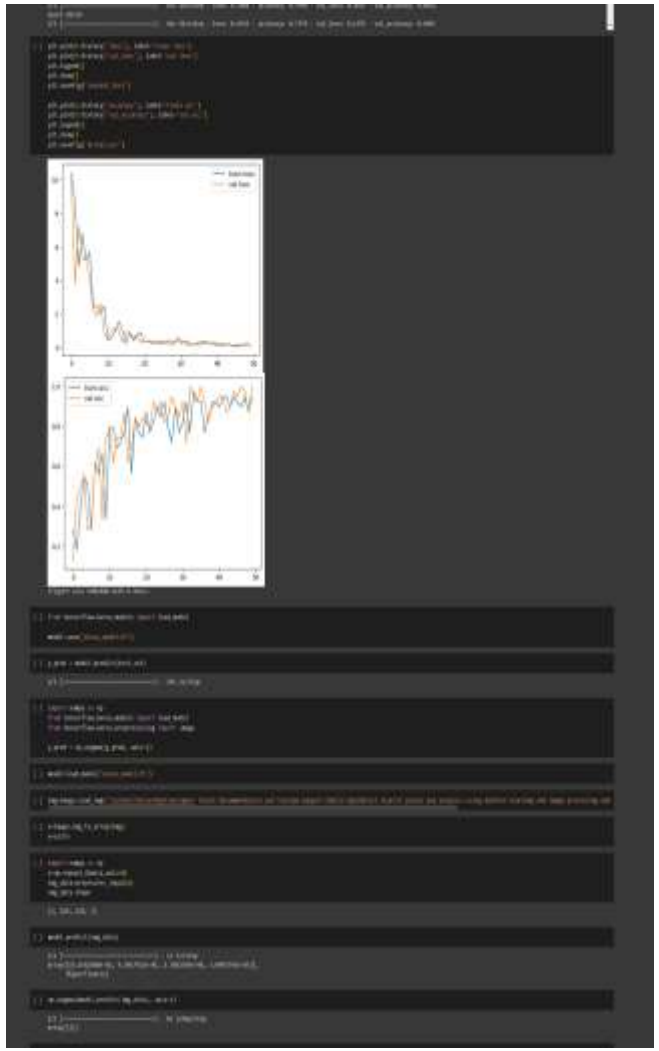
Technologies

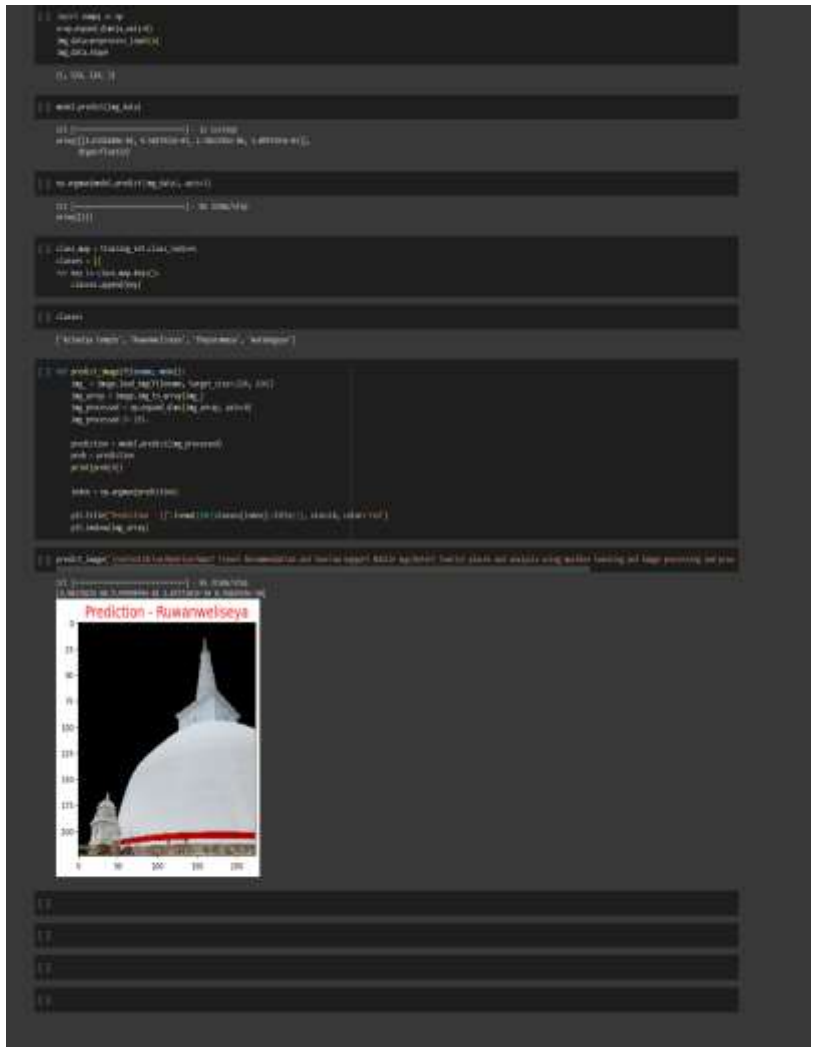
- Python
- Open CV
- Fire Base/MongoDB
- TensorFlow

DEMONSTRATION


Detect_tourist_places.ipynb
File Edit View Insert Runtime Tools Help Last saved at 11:02PM
Comment
Share
+ Code
+ Text
Connect
from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
from tensorflow.keras.models import Sequential
import numpy as np
from glob import glob
import matplotlib.pyplot as plt
config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.5
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
size=224
train_path = '/content/drive/MyDrive/Smart Travel Recommendation and Tourism Support Mobile App/Detect tourist places and analysis using machine learning and image processing and provi
valid_path = '/content/drive/MyDrive/Smart Travel Recommendation and Tourism Support Mobile App/Detect tourist places and analysis using machine learning and image processing and provi
%cd '/content/drive/MyDrive/Smart Travel Recommendation and Tourism Support Mobile App/Detect tourist places and analysis using machine learning and image processing and provide detail
/content/drive/MyDrive/Smart Travel Recommendation and Tourism Support Mobile App/Detect tourist places and analysis using machine learning and image processing and provide details
resnet = ResNet50(input_shape=[size,size] + [3], weights='imagenet', include_top=False)
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94765736/94765736 [==============================] - 0s 0us/step
for layer in resnet.layers:
layer.trainable = False
folders = glob(train_path+'/*')
x = Flatten()(resnet.output)
prediction = Dense(len(folders), activation='softmax')(x)
model = Model(inputs=resnet.input, outputs=prediction)
model.summary()



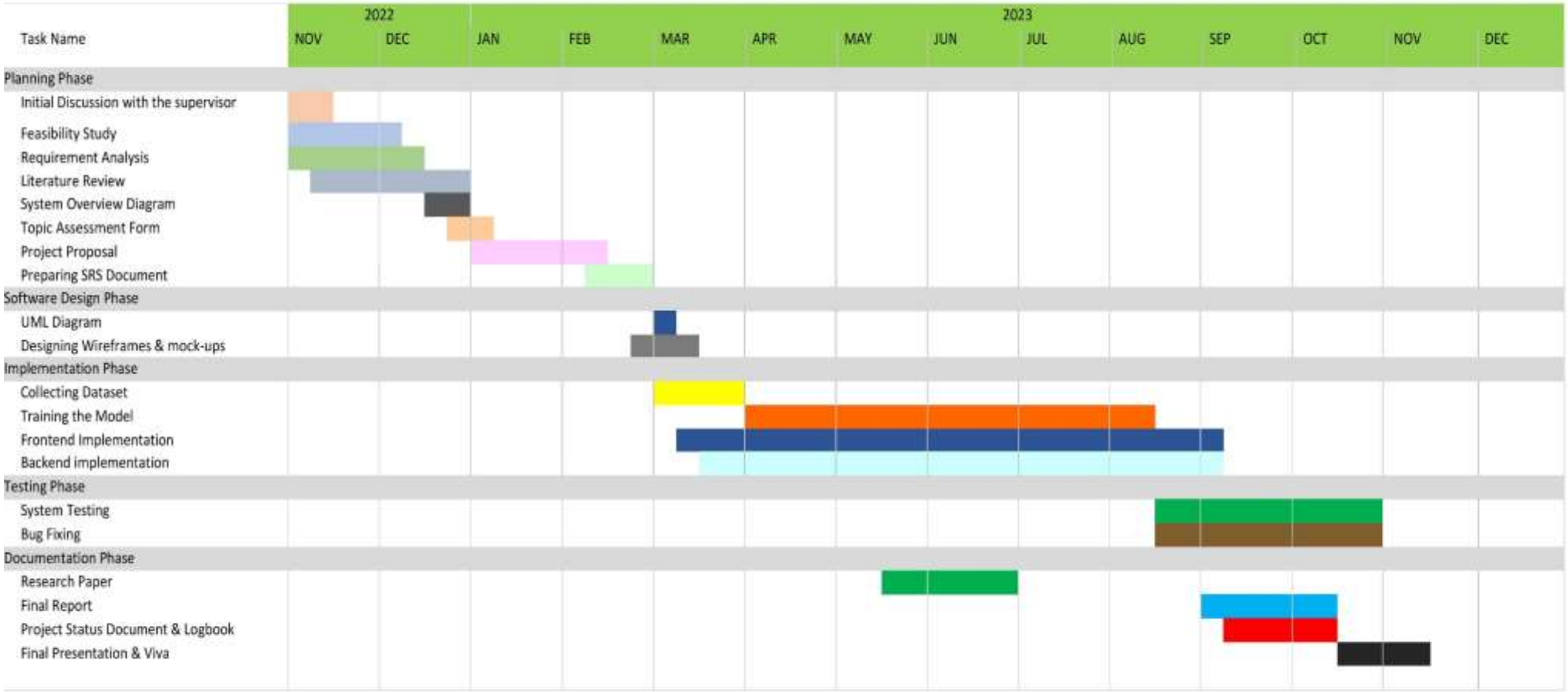



Prediction - Ruwanweliseya

TASK TO BE DONE

- Add information to relevent images
- Fine tune data set
- Add more images and information to datasets
- Develop mobile app

GANTT CHART


Task Name
2022
2023
NOV
DEC
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
Planning Phase
Initial Discussion with the supervisor
Feasibility Study
Requirement Analysis
Literature Review
System Overview Diagram
Topic Assessment Form
Project Proposal
Preparing SRS Document
Software Design Phase
UML Diagram
Designing Wireframes & mock-ups
Implementation Phase
Collecting Dataset
Training the Model
Frontend Implementation
Backend implementation
Testing Phase
System Testing
Bug Fixing
Documentation Phase
Research Paper
Final Report
Project Status Document & Logbook
Final Presentation & Viva

REFERENCES

[1] D. Buhalis, "Smart Tourism Destinations Enhancing Tourism Experience Through Personalisation of Services," p. 14, 2015.

[2] Santos-Júnior, Adalberto; Mendes-Filho, Luiz; Almeida García, Fernando; Manuel Simões, José;, Smart Tourism Destinations: a study based on the view of the stakeholders, p. 23, 2017.

[3] N. Godewithana, K. Jayasena, C. Nagarawaththa, P. Croos, B. Harshanath and J. Alosius, "Historical Places & Monuments Identification System," 2020.

[4] B. M. a. M. T. M. M. Etaati, "Cross Platform Web-based Smart Tourism Using Deep Monument Mining," 2019.

IT19192024 | JAYAWARDHANA E.H.K

B.Sc.(Hons) in Information Technology Specializing in Information Technology

INDIVIDUAL COMPONENT

Personalized location recommendation and service recommendation refers to the development of a system that provides relevant location and location-based service recommendations tailored to their specific travel preferences such as travel style, budget, and interests.

RESEARCH QUESTION

- There many ways to identify places and services in the tourism industry based on technology. but tourism industry not fully digitalized.
- difficult to choose the places and services according to their preferences.
- Even if there is smartphone, does not recommend relevant location and location-based services that meet preferences such as travel style, budget and interests.
- This shows a need to provide recommendations based on data from personalized style, budget, interests, preferences and behavior.

SPECIFICS AND SUB OBJECTIVES

- **SPECIFICS -** personalized locations recommendation and services recommendation is to provide the relevant location and location-based services, that are tailored to their specific travel preferences such as travel style, budget, and interests.

- **OBJECTIVE -** provide a customized experience for tourists by providing them with relevant options in real time and making the recommendations more accurate and effective.

METHODOLOGY

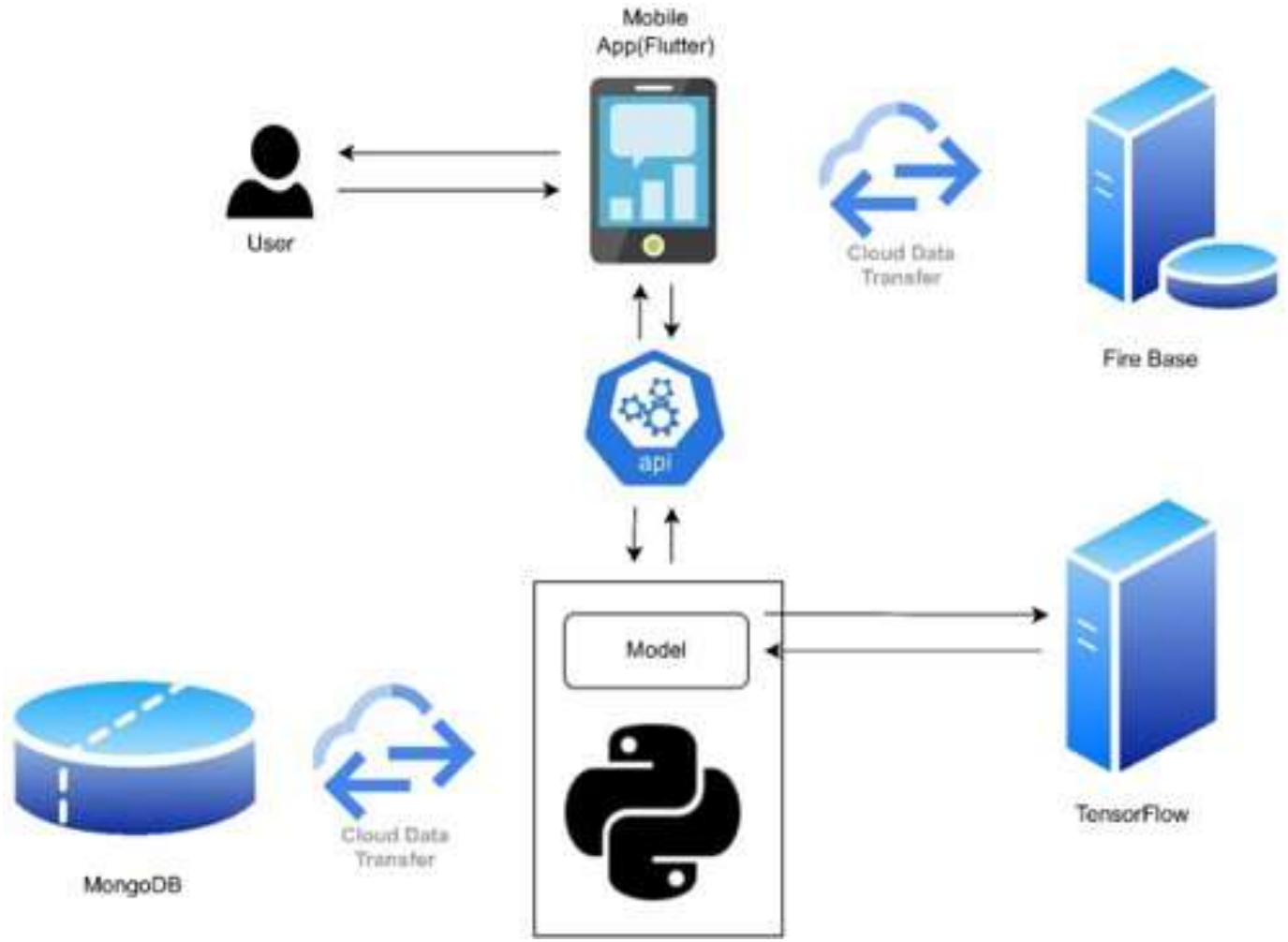
- A machine learning-based recommendation app to provide location-based services tailored to their specific travel preferences such as relevant location and travel style, budget, and interests.

- Personalized recommendations are based on data collected from previous travelers' preferences and behaviors, making them more accurate and effective.

- Collaborative filtering and content-based filtering - This technology suggests services and places that users have used or rated highly.

METHODOLOGY

System Diagram

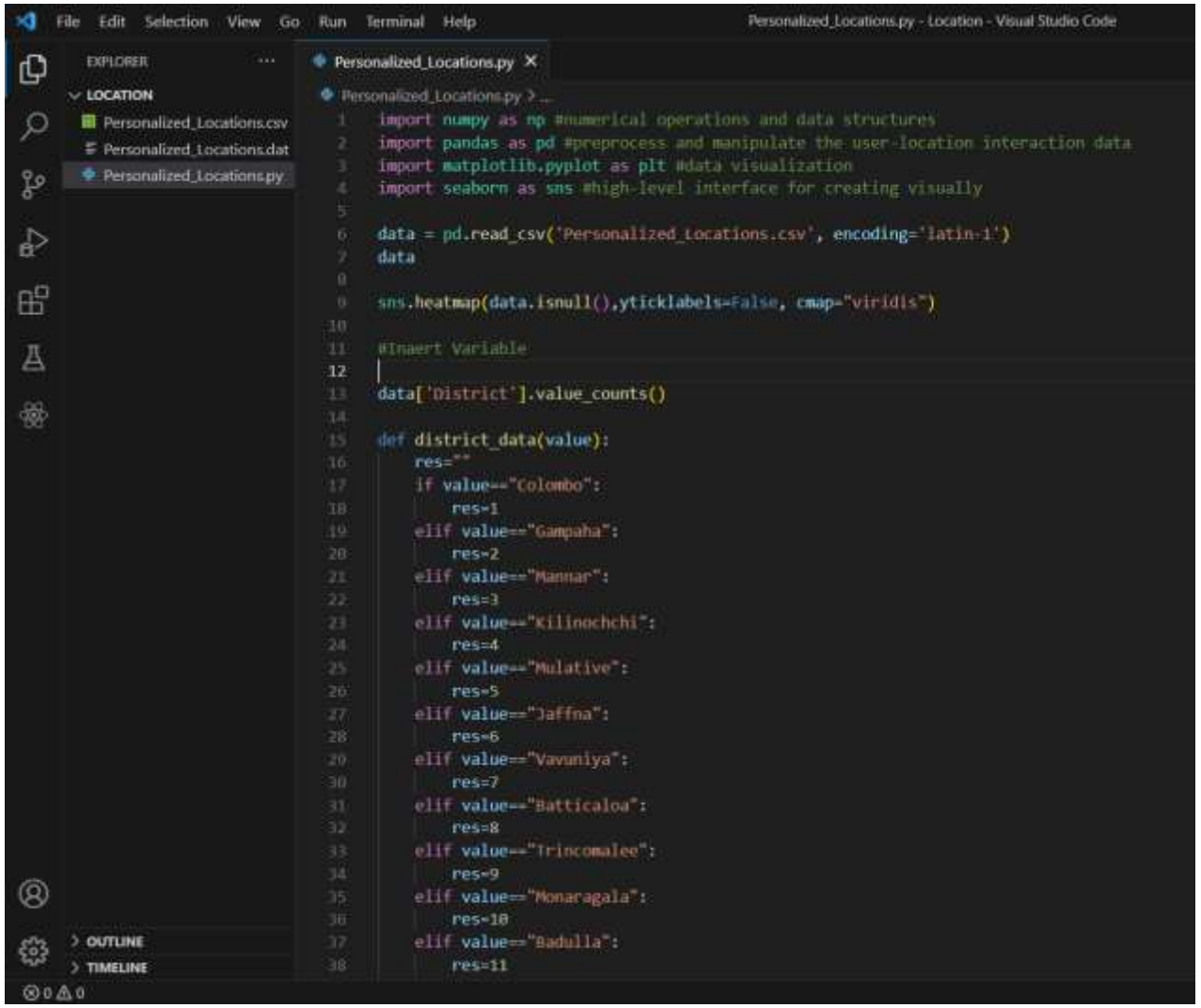


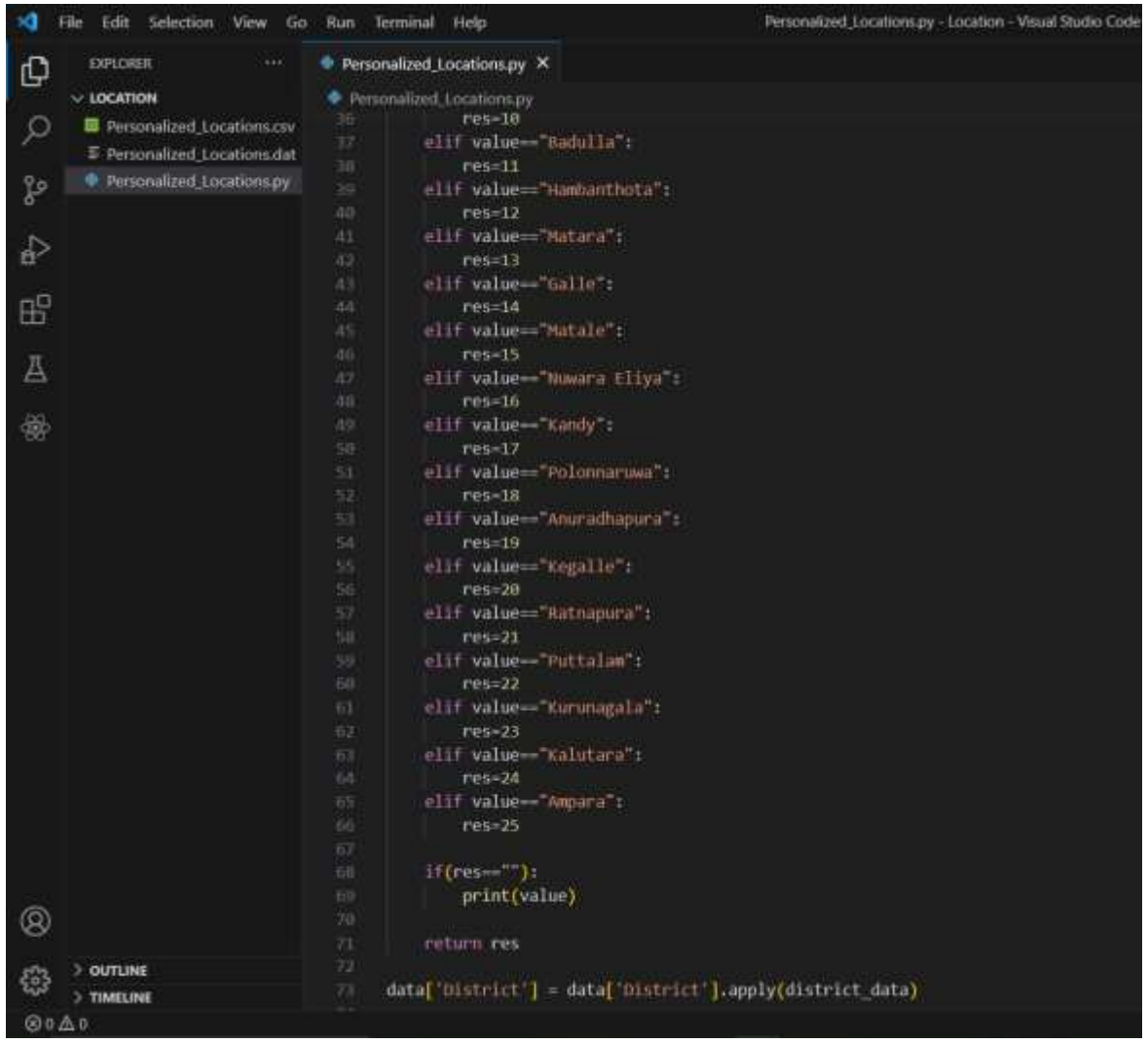
METHODOLOGY

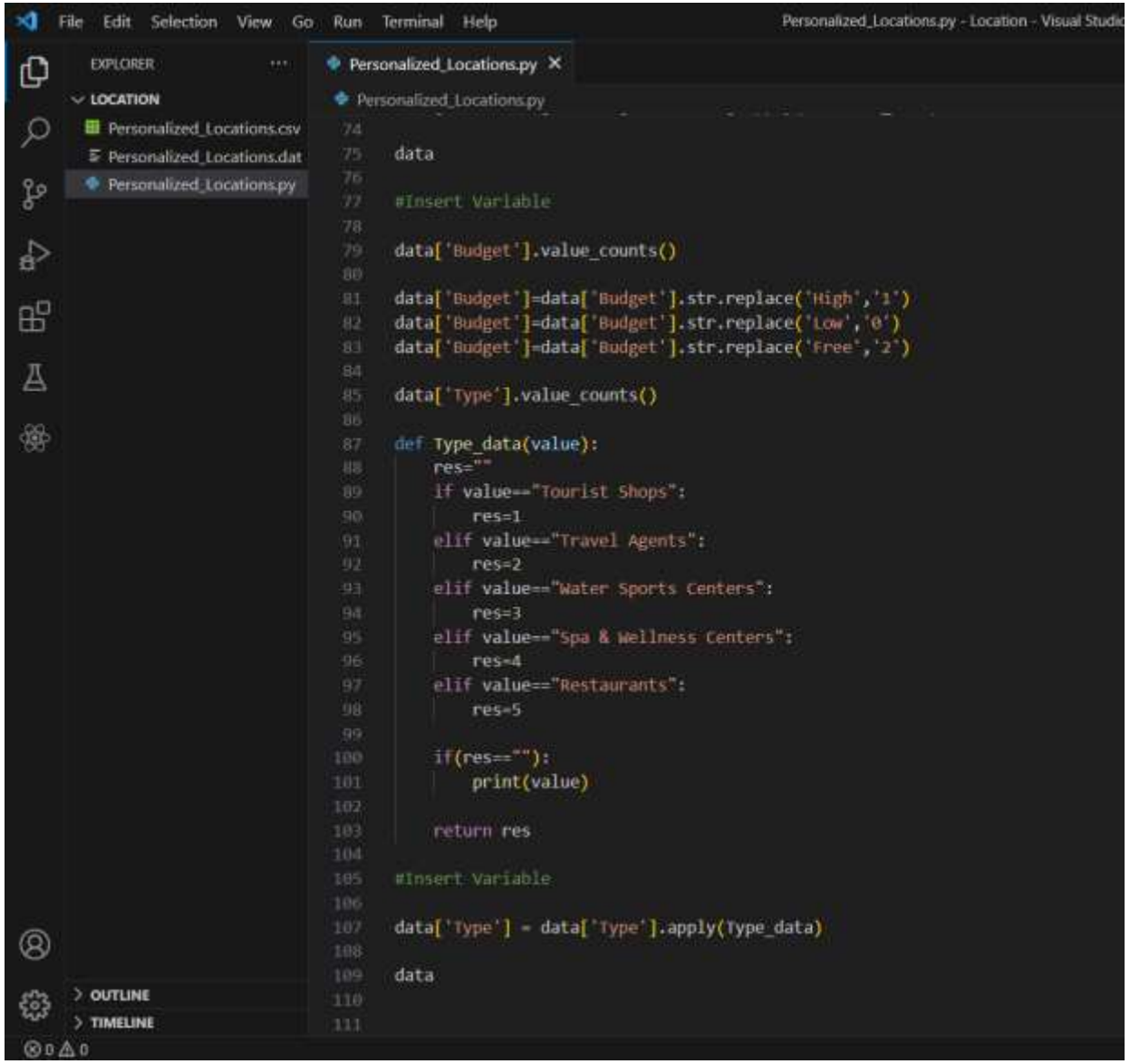
Technologies

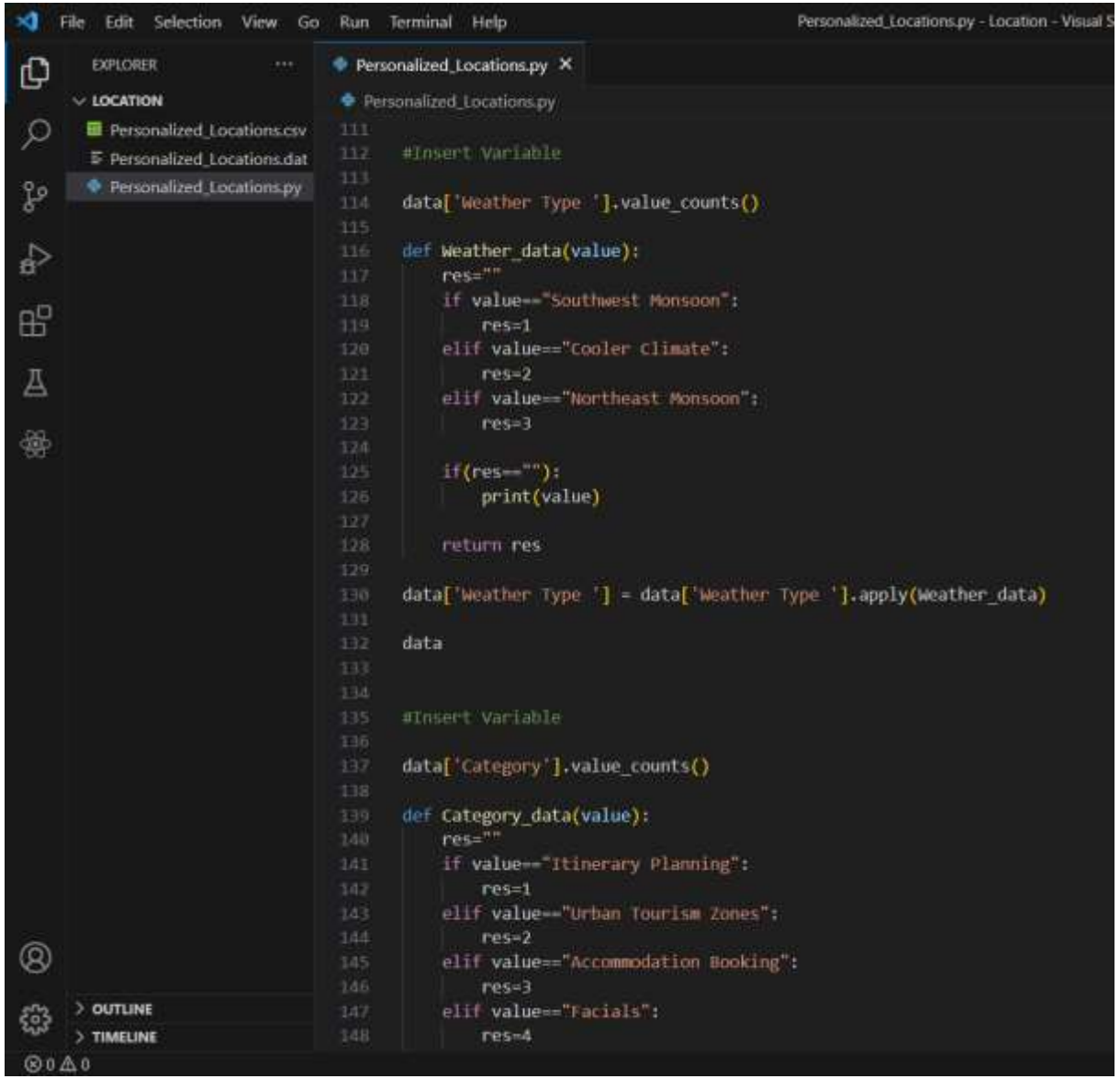
- Python
- Fire Base/MongoDB
- TensorFlow

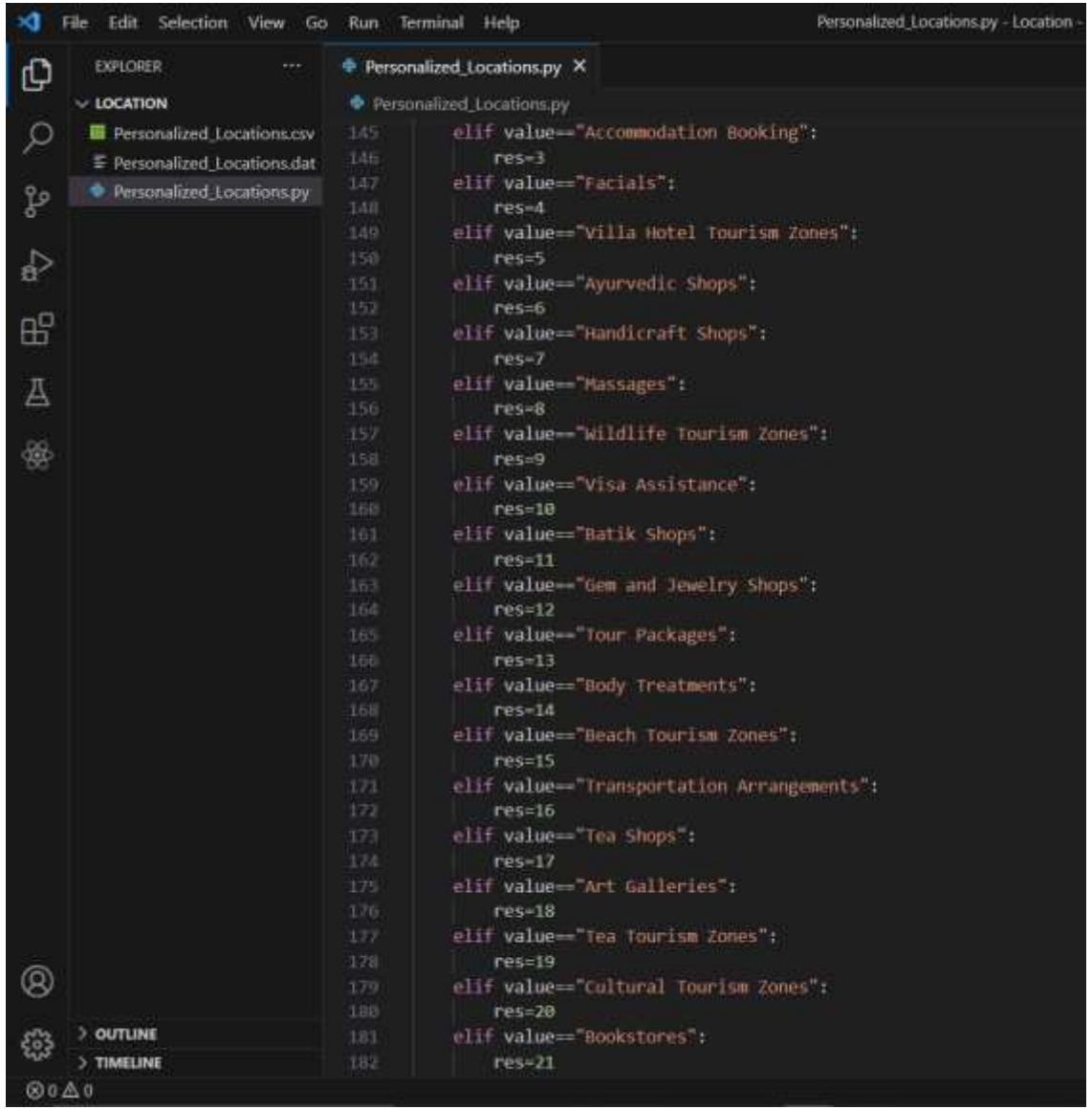
DEMONSTRATION

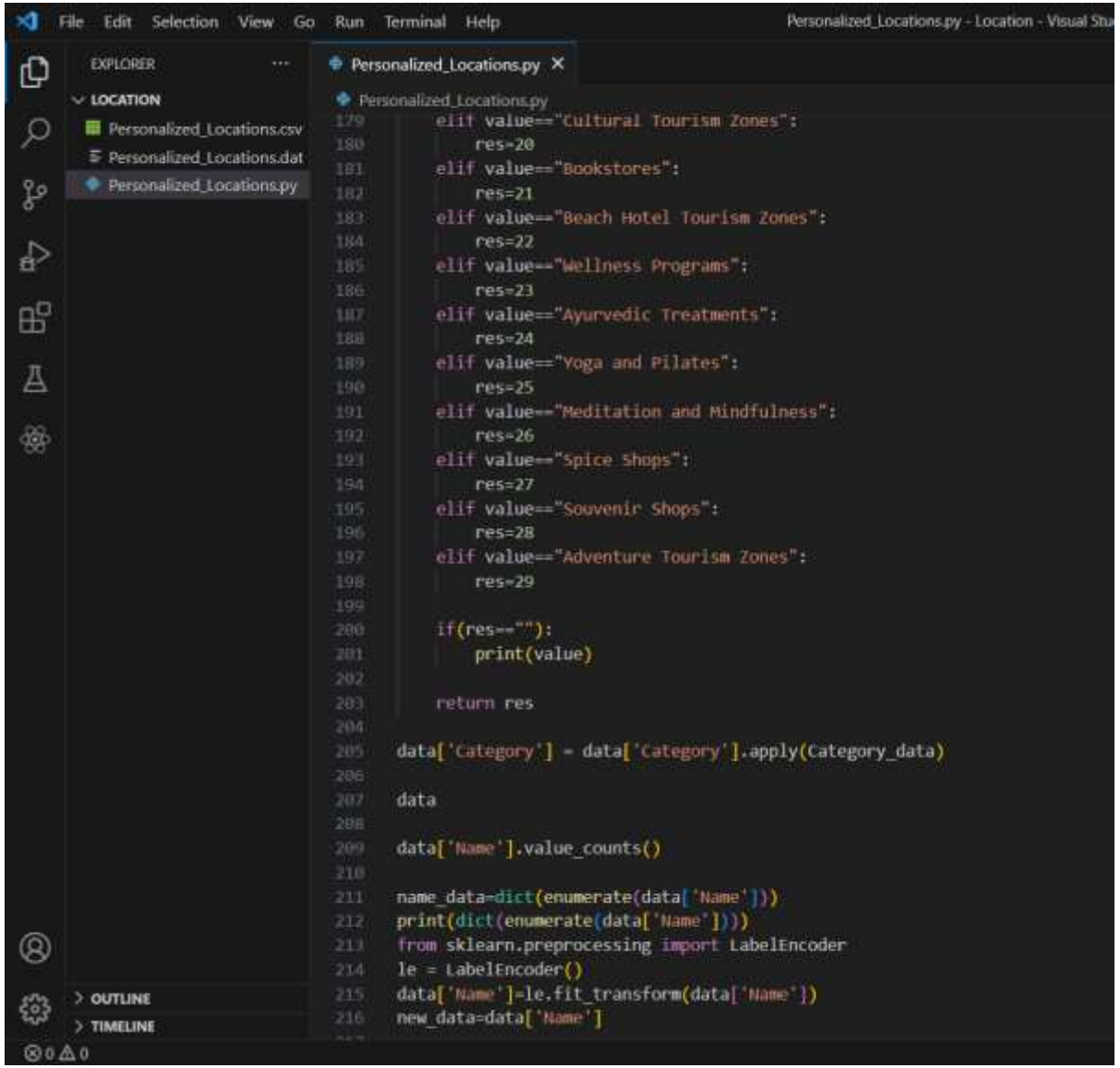

File Edit Selection View Go Run Terminal Help
Personalized_Locations.py - Location - Visual Studio Code
EXPLORER
LOCATION
Personalized_Locations.csv
Personalized_Locations.dat
Personalized_Locations.py
Personalized_Locations.py
import numpy as np #numerical operations and data structures
import pandas as pd #preprocess and manipulate the user-location interaction data
import matplotlib.pyplot as plt #data visualization
import seaborn as sns #high-level interface for creating visually
data = pd.read_csv('Personalized_Locations.csv', encoding='latin-1')
data
sns.heatmap(data.isnull(),yticklabels=False, cmap="viridis")
#Insert Variable
data['District'].value_counts()
def district_data(value):
res=""
if value=="Colombo":
res=1
elif value=="Gampaha":
res=2
elif value=="Mannar":
res=3
elif value=="Kilinochchi":
res=4
elif value=="Mulative":
res=5
elif value=="Jaffna":
res=6
elif value=="Vavuniya":
res=7
elif value=="Batticaloa":
res=8
elif value=="Trincomalee":
res=9
elif value=="Monaragala":
res=10
elif value=="Badulla":
res=11
OUTLINE
TIMELINE

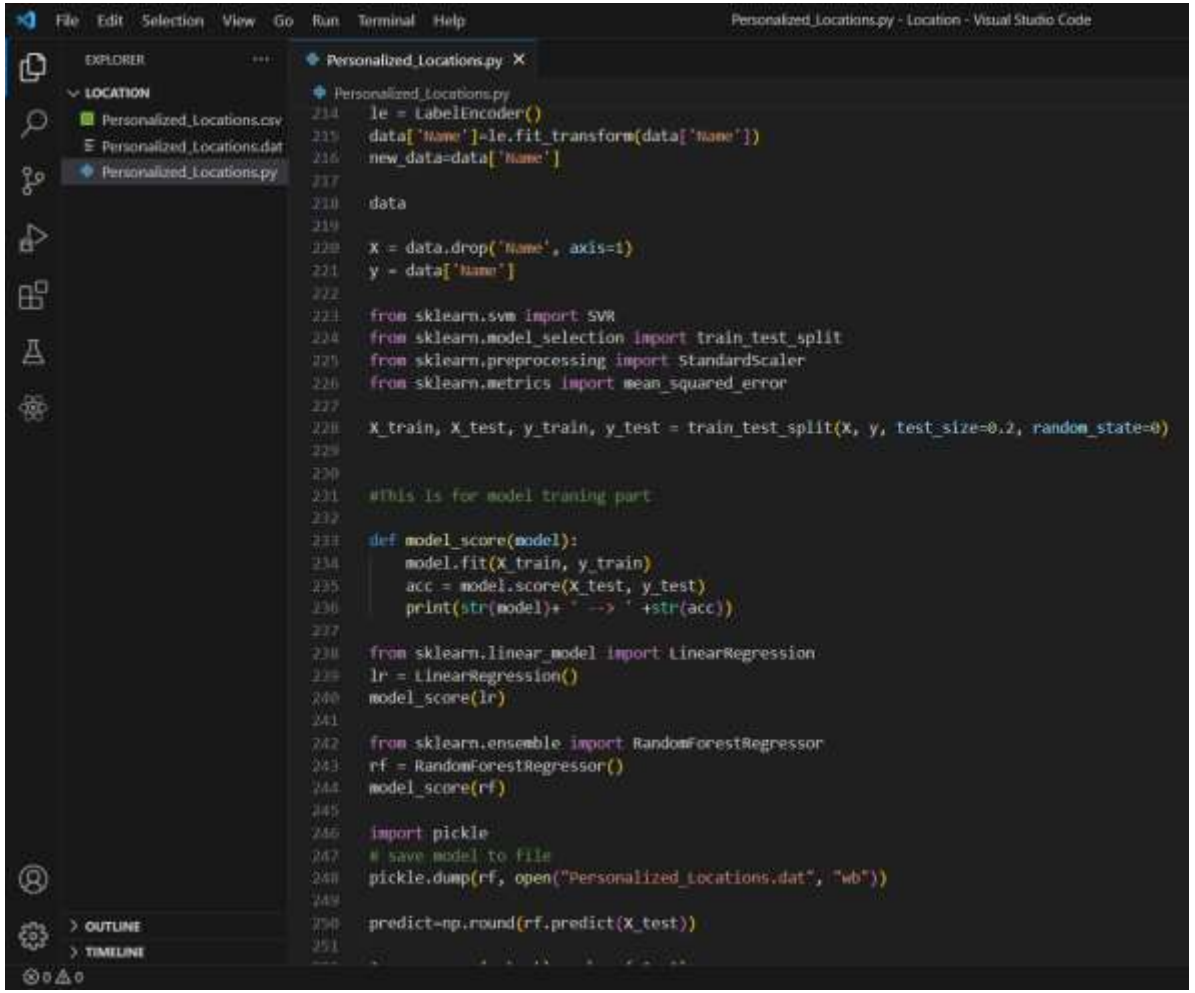

File Edit Selection View Go Run Terminal Help
Personalized_Locations.py - Location - Visual Studio Code
EXPLORER
LOCATION
Personalized_Locations.csv
Personalized_Locations.dat
Personalized_Locations.py
Personalized_Locations.py
res=10
elif value=="Badulla":
res=11
elif value=="Hambanthota":
res=12
elif value=="Matara":
res=13
elif value=="Galle":
res=14
elif value=="Matale":
res=15
elif value=="Nuwara Eliya":
res=16
elif value=="Kandy":
res=17
elif value=="Polonnaruwa":
res=18
elif value=="Anuradhapura":
res=19
elif value=="Kegalle":
res=20
elif value=="Ratnapura":
res=21
elif value=="Puttalam":
res=22
elif value=="Kurunagala":
res=23
elif value=="Kalutara":
res=24
elif value=="Ampara":
res=25
if(res==""):
print(value)
return res
data['District'] = data['District'].apply(district_data)
OUTLINE
TIMELINE

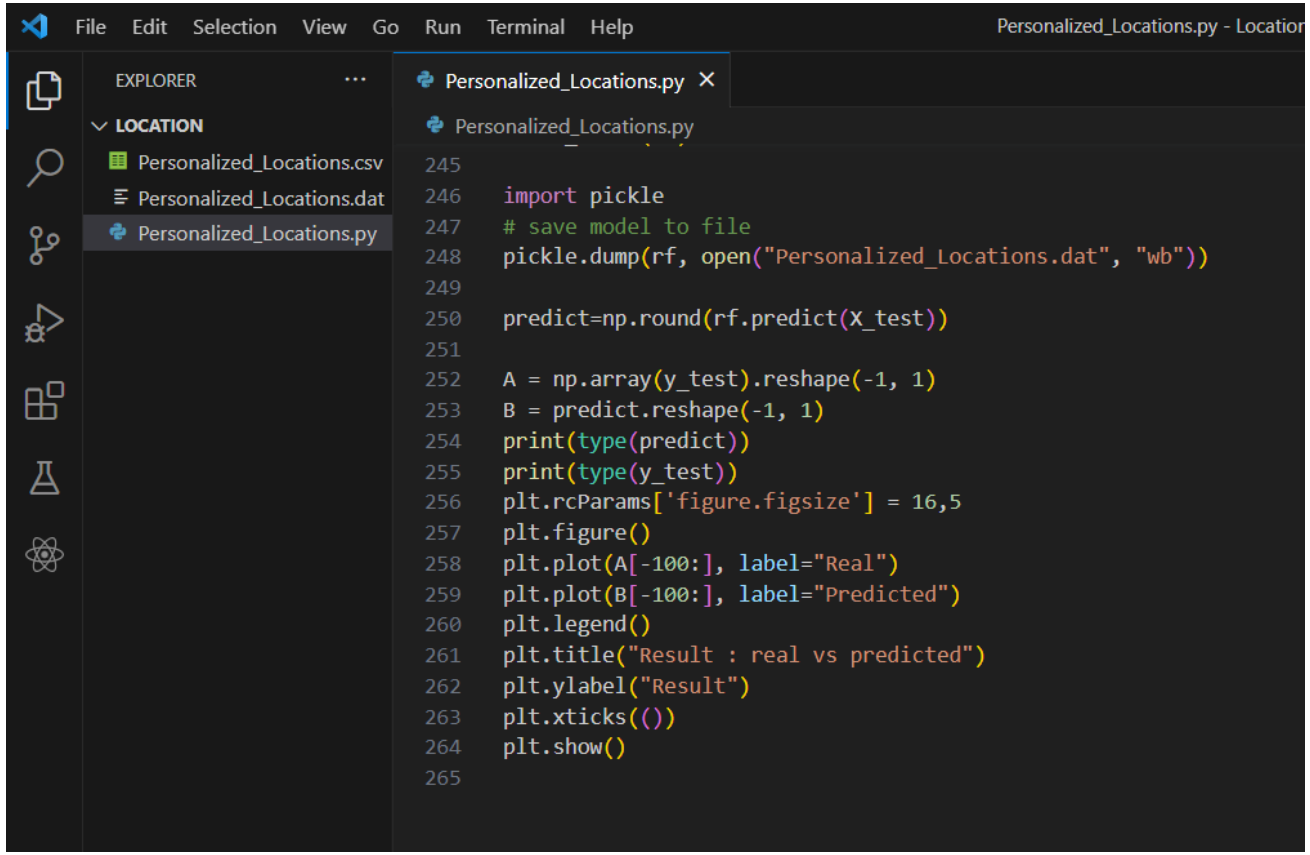

File Edit Selection View Go Run Terminal Help
Personalized_Locations.py - Location - Visual Studio
EXPLORER
LOCATION
Personalized_Locations.csv
Personalized_Locations.dat
Personalized_Locations.py
Personalized_Locations.py
OUTLINE
TIMELINE
data
#Insert Variable
data['Budget'].value_counts()
data['Budget']=data['Budget'].str.replace('High','1')
data['Budget']=data['Budget'].str.replace('Low','0')
data['Budget']=data['Budget'].str.replace('Free','2')
data['Type'].value_counts()
def Type_data(value):
res=""
if value=="Tourist Shops":
res=1
elif value=="Travel Agents":
res=2
elif value=="Water Sports Centers":
res=3
elif value=="Spa & Wellness Centers":
res=4
elif value=="Restaurants":
res=5
if(res==""):
print(value)
return res
#Insert Variable
data['Type'] = data['Type'].apply(Type_data)
data

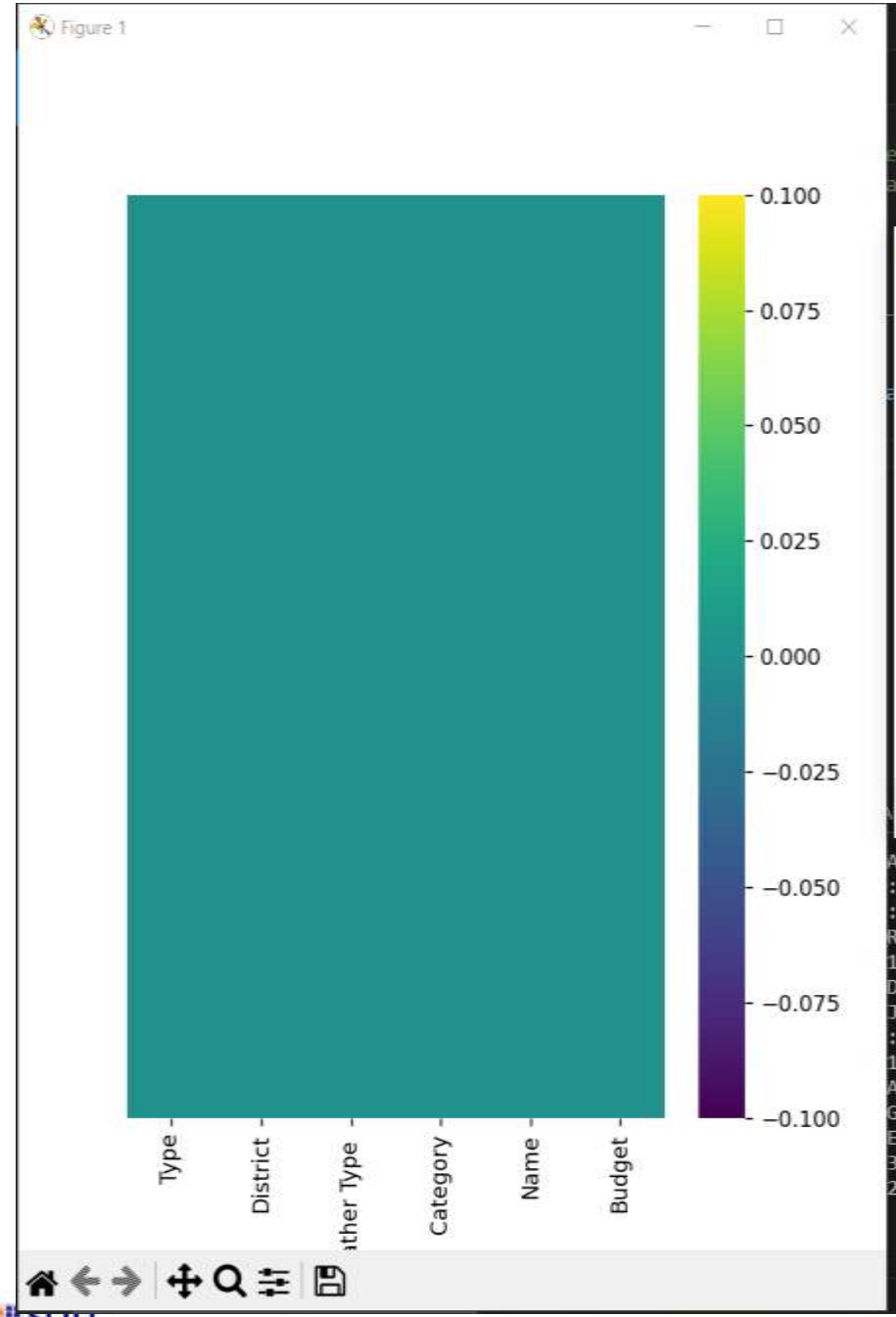

File Edit Selection View Go Run Terminal Help
Personalized_Locations.py - Location - Visual S
EXPLORER
LOCATION
Personalized_Locations.csv
Personalized_Locations.dat
Personalized_Locations.py
Personalized_Locations.py
OUTLINE
TIMELINE
#Insert Variable
data['Weather Type '].value_counts()
def Weather_data(value):
res=""
if value=="Southwest Monsoon":
res=1
elif value=="Cooler Climate":
res=2
elif value=="Northeast Monsoon":
res=3
if(res==""):
print(value)
return res
data['Weather Type '] = data['Weather Type '].apply(Weather_data)
data
#Insert Variable
data['Category'].value_counts()
def Category_data(value):
res=""
if value=="Itinerary Planning":
res=1
elif value=="Urban Tourism Zones":
res=2
elif value=="Accommodation Booking":
res=3
elif value=="Facials":
res=4


File Edit Selection View Go Run Terminal Help
Personalized_Locations.py - Location -
EXPLORER
LOCATION
Personalized_Locations.csv
Personalized_Locations.dat
Personalized_Locations.py
Personalized_Locations.py
OUTLINE
TIMELINE
elif value=="Accommodation Booking":
res=3
elif value=="Facials":
res=4
elif value=="Villa Hotel Tourism Zones":
res=5
elif value=="Ayurvedic Shops":
res=6
elif value=="Handicraft Shops":
res=7
elif value=="Massages":
res=8
elif value=="Wildlife Tourism Zones":
res=9
elif value=="Visa Assistance":
res=10
elif value=="Batik Shops":
res=11
elif value=="Gem and Jewelry Shops":
res=12
elif value=="Tour Packages":
res=13
elif value=="Body Treatments":
res=14
elif value=="Beach Tourism Zones":
res=15
elif value=="Transportation Arrangements":
res=16
elif value=="Tea Shops":
res=17
elif value=="Art Galleries":
res=18
elif value=="Tea Tourism Zones":
res=19
elif value=="Cultural Tourism Zones":
res=20
elif value=="Bookstores":
res=21


File Edit Selection View Go Run Terminal Help
Personalized_Locations.py - Location - Visual Stu
EXPLORER
LOCATION
Personalized_Locations.csv
Personalized_Locations.dat
Personalized_Locations.py
Personalized_Locations.py
OUTLINE
TIMELINE
elif value=="Cultural Tourism Zones":
res=20
elif value=="Bookstores":
res=21
elif value=="Beach Hotel Tourism Zones":
res=22
elif value=="Wellness Programs":
res=23
elif value=="Ayurvedic Treatments":
res=24
elif value=="Yoga and Pilates":
res=25
elif value=="Meditation and Mindfulness":
res=26
elif value=="Spice Shops":
res=27
elif value=="Souvenir Shops":
res=28
elif value=="Adventure Tourism Zones":
res=29
if(res==""):
print(value)
return res
data['Category'] = data['Category'].apply(Category_data)
data
data['Name'].value_counts()
name_data=dict(enumerate(data['Name']))
print(dict(enumerate(data['Name'])))
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data['Name']=le.fit_transform(data['Name'])
new_data=data['Name']

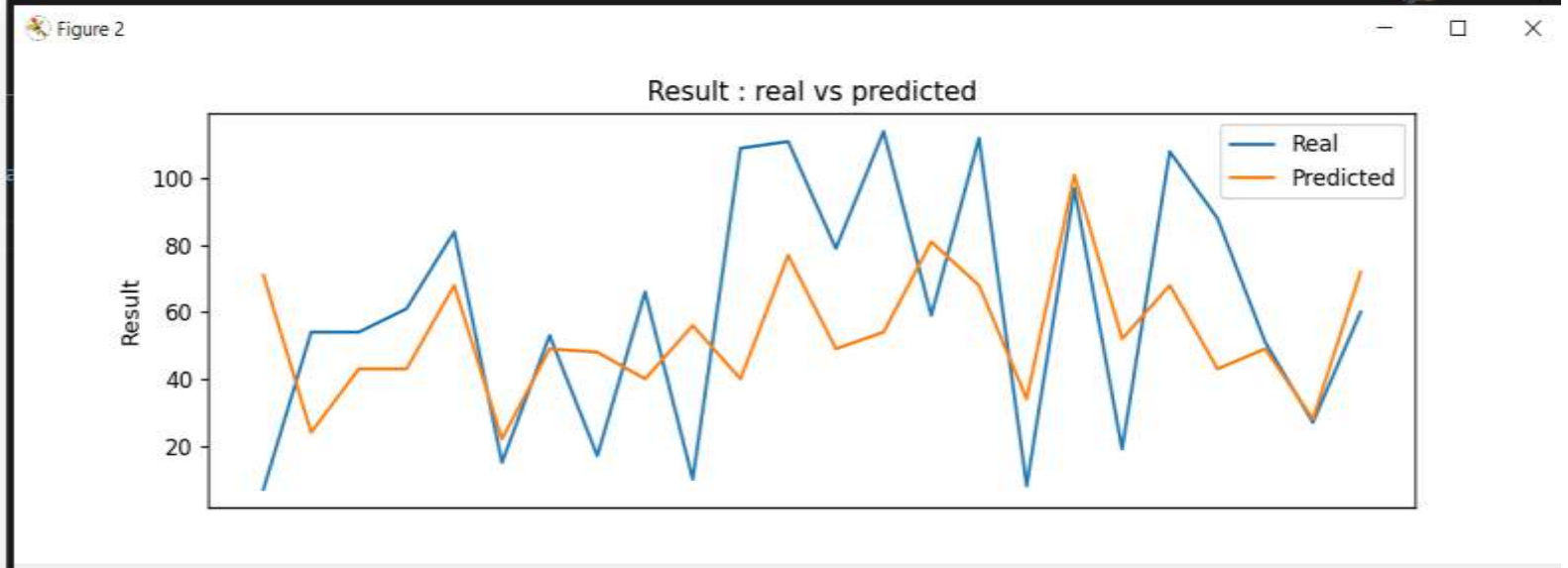

File Edit Selection View Go Run Terminal Help
Personalized_Locations.py - Location - Visual Studio Code
EXPLORER
LOCATION
Personalized_Locations.csv
Personalized_Locations.dat
Personalized_Locations.py
Personalized_Locations.py
le = LabelEncoder()
data['Name']=le.fit_transform(data['Name'])
new_data=data['Name']
data
X = data.drop('Name', axis=1)
y = data['Name']
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
#This is for model traning part
def model_score(model):
model.fit(X_train, y_train)
acc = model.score(X_test, y_test)
print(str(model)+ ' --> ' +str(acc))
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
model_score(lr)
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
model_score(rf)
import pickle
# save model to file
pickle.dump(rf, open("Personalized_Locations.dat", "wb"))
predict=np.round(rf.predict(X_test))
OUTLINE
TIMELINE


File Edit Selection View Go Run Terminal Help
Personalized_Locations.py - Location
EXPLORER
LOCATION
Personalized_Locations.csv
Personalized_Locations.dat
Personalized_Locations.py
Personalized_Locations.py
import pickle
# save model to file
pickle.dump(rf, open("Personalized_Locations.dat", "wb"))
predict=np.round(rf.predict(X_test))
A = np.array(y_test).reshape(-1, 1)
B = predict.reshape(-1, 1)
print(type(predict))
print(type(y_test))
plt.rcParams['figure.figsize'] = 16,5
plt.figure()
plt.plot(A[-100:], label="Real")
plt.plot(B[-100:], label="Predicted")
plt.legend()
plt.title("Result : real vs predicted")
plt.ylabel("Result")
plt.xticks(())
plt.show()



Personalized_Locations.py - Location - Visual Studio Code

erations and data structures
and manipulate the user-location interaction data




```
AGICAL ISLE HOLIDAYS (PVT) LTD', 67: 'TAPROBANE TRAVELS ( PVT ) LTD', 68: "CEYLON'S BEST TOURS (P
: 'GLOBAL CONNECTION TRAVELS & TOURS (PVT) LTD', 71: 'REGIONAL DEVELOPMENT HOLDINGS (PVT) LTD', 7
: 'GALLE TRAVLS & TOURS (PVT) LTD', 74: 'WILD HOLIDAYS TRAVELS (PVT) LTD', 75: 'UNITED LEISURE (P
RA TRAVELS & TOURS (PVT) LTD', 78: 'MAL KEY TRAVELS & TOURS (PVT) LTD', 79: 'KAMWELTA TRAVEL & TO
1: 'N M LANKA TRAVELS COMPANY', 82: 'UPALI TRAVELS (PVT) LTD', 83: 'M.D.GUNASENA TRAVEL INTERNATI
D', 85: 'GREEN HOLIDAY CENTRE (PVT) LTD', 86: 'CLASSIC TRAVEL (PVT) LTD', 87: 'ORIENT TOURS', 88:
JEWELLERS', 91: "TONY'S FAIR", 92: 'NISHANTHA WOOD CARVINGS', 93: 'HOUSE OF FASHINES GARMENTS', 9
: 'NORITAKE LANKA PORECELAIN', 97: 'PRASANNA GEM CENTER', 98: 'ASIAN GEMS & JEWELLERY', 99: 'SERE
101: 'LAKMEDURA', 102: 'HOUSE OF FASHION GARMENTS', 103: 'TIESH BY LAKMINI', 104: 'SILK & STYLE',
ANAJA GEM MUSEUM', 107: 'FIOR DRISSAGE JEWELLERS', 108: 'PREMADASAS GEMS AND JEWELLERY', 109: 'PR
GEMS', 111: 'BATIK FAIR BY KOTTEGODA', 112: 'BATIK FAIR BY KOTTEGODA', 113: 'BATIK LAND', 114: 'D
ERS', 116: 'A.M.A.CAREEM JEWELLERS', 117: 'A.M.A CAREEM JEWELLERS', 118: 'ISINI GEM JEWELLERS'}
3
207216
```

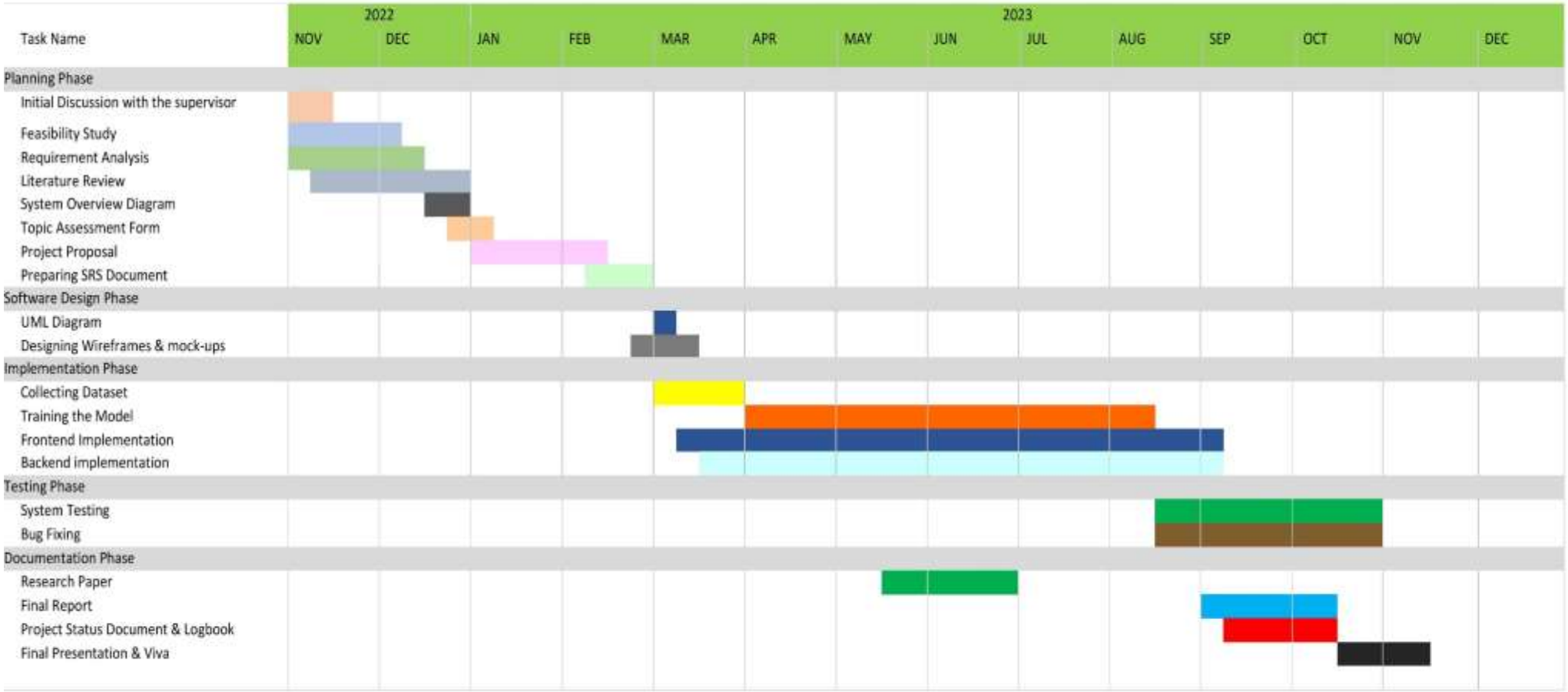
Python
Python

Ln 11, Col 17 Spaces: 4 UTF-8 CRLF Python 3.11.1 64-bit

TASK TO BE DONE

- The front end and the back end of the app need to be developed.
- Integration of Systems
- System Integration
- The model needs to be improved more and implemented.

GANTT CHART


Task Name
2022
2023
NOV
DEC
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
Planning Phase
Initial Discussion with the supervisor
Feasibility Study
Requirement Analysis
Literature Review
System Overview Diagram
Topic Assessment Form
Project Proposal
Preparing SRS Document
Software Design Phase
UML Diagram
Designing Wireframes & mock-ups
Implementation Phase
Collecting Dataset
Training the Model
Frontend Implementation
Backend implementation
Testing Phase
System Testing
Bug Fixing
Documentation Phase
Research Paper
Final Report
Project Status Document & Logbook
Final Presentation & Viva

REFERENCES

[1] "history," [Online]. Available: https://www.sltda.gov.lk/en/history.

[2] A. Abeysinghe, "The Diplomat," 08 November 2022. [Online]. Available: https://thediplomat.com/2022/11/sri-lankas-tourism-industry-revives-despite-public-protests/#:~:text=Sri%20Lanka's%20tourism%20sector%20plays,needed%20revenue%20for%20the%20government.

[3] "wikipedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Tourism_in_Sri_Lanka#:~:text=For%20centuries%2C%20Sri%20Lanka%20has,its%20size%20in%20the%20world%22.

[4] A. Kontogianni and E. Alepis, "Smart tourism: State of the art and literature review for the last six years," Array, 2020.

[5] M. Figueredo, J. Ribeiro, . N. Cacho, A. Thome, A. Cacho, F. Lopes and V. Araujo, "From Photos to Travel Itinerary: A Tourism Recommender System for," 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications, pp. 85-92, 2018.

[6] U. Gretzel, M. Sigala, Z. Xiang and C. Koo, "Smart tourism: foundations and developments," Crossmark, pp. 180-188, 2015.

[7] A. Savvopoulos, G. Kalogeras, A. Mariotti , C. Bernini , C. Alexakos and A. Kalogeras, "Data Analytics supporting decision making in the Tourism Sector," IEEE Xplore, 2019.

[8] H. Alghamdi, . S. Zhu and A. E. Saddik, "E-Tourism: Mobile Dynamic Trip Planner," 2016 IEEE International Symposium on Multimedia, pp. 185-188, 2016.

[9] C. Choi, M. Cho, J. Choi, M. Hwang, J. Park and P. Kim, "Travel Ontology for Intelligent Recommendation System," 2009 Third Asia International Conference on Modelling & Simulation, pp. 637-642, 2009.

[10] Prasadini Ruwani Gunathilake , "Machine Learning-Based Smart Tourist Support System (Smart Guide)," Springer, 2022.

IT20147846 | THENNAKOON S.U.

B.Sc.(Hons) in Information Technology Specializing in Information Technology

INDIVIDUAL COMPONENT

Developing a system that analyze the emotional state of the user based on their facial expressions and provide personalized travel recommendations tailored to their emotional state.

RESEARCH QUESTION

- Travel recommendations are often based on a generic set of preferences rather than individual emotional states.
- Traditional surveys or feedback mechanisms may not capture the real-time emotional state of the user.
- Users may have different emotional states during different stages of their travel experience.

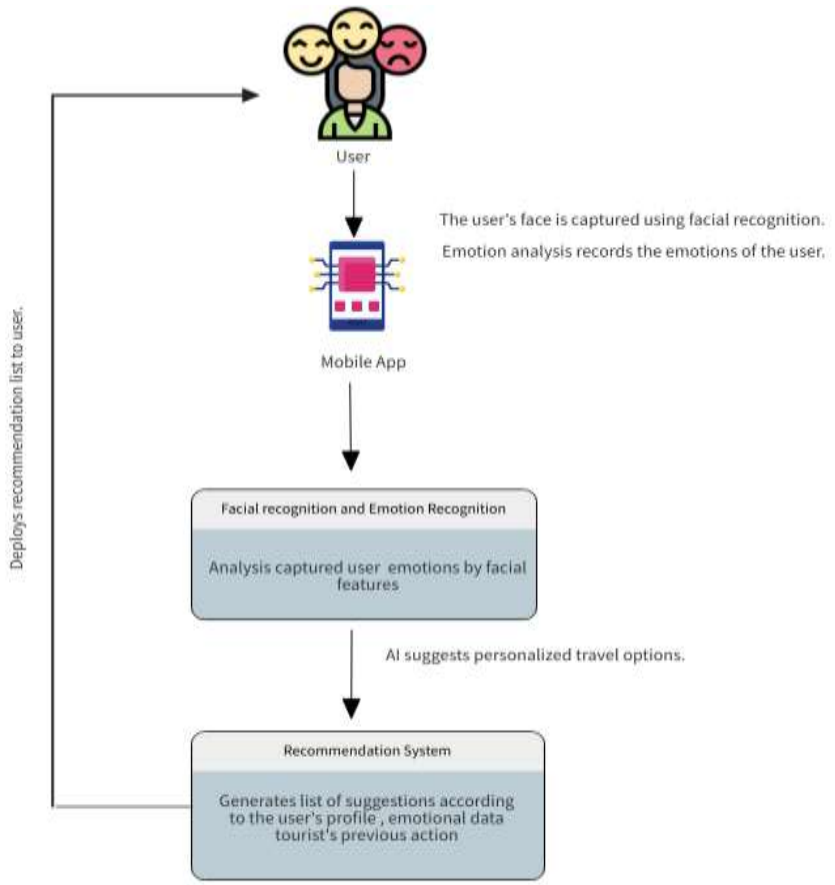
SPECIFICS AND SUB OBJECTIVES

- Accurately identify the emotions of tourists through facial recognition technology and provide personalized recommendations and support services based on the emotional state of the user.
- Offer a range of travel options that cater to different emotional states, such as adventure activities for those feeling adventurous or relaxing experiences for those feeling stressed.

METHODOLOGY

System Diagram

- Gather images on different types of emotions to data set



METHODOLOGY

Technologies

- Python
- Keras
- TensorFlow

DEMONSTRATION

face_classification.ipynb

File Edit View Insert Runtime Tools Help Save failed

Comment Share

+ Code + Text

RAM Disk

```
[1] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt

    import tensorflow as tf
    from tensorflow.keras.preprocessing.image import ImageDataGenerator
    from tensorflow.keras.applications.inception_v3 import InceptionV3
    from tensorflow.keras.layers import Dense, Dropout, BatchNormalization
    from tensorflow.keras.models import Model
    from tensorflow.keras.optimizers import Adam
    from tensorflow.keras.preprocessing import image
    from tensorflow.keras.preprocessing.image import load_img, img_to_array
```

```
[2] from google.colab import drive
    drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[3] %cd '/content/drive/MyDrive/Smart Travel Recommendation and Tourism Support Mobile App/Emotion based activity suggestion'
```

```
/content/drive/MyDrive/Smart Travel Recommendation and Tourism Support Mobile App/Emotion based activity suggestion
```

```
[4] data_dir = '/content/drive/MyDrive/Smart Travel Recommendation and Tourism Support Mobile App/Emotion based activity suggestion'

    data = tf.keras.preprocessing.image_dataset_from_directory(data_dir)
```

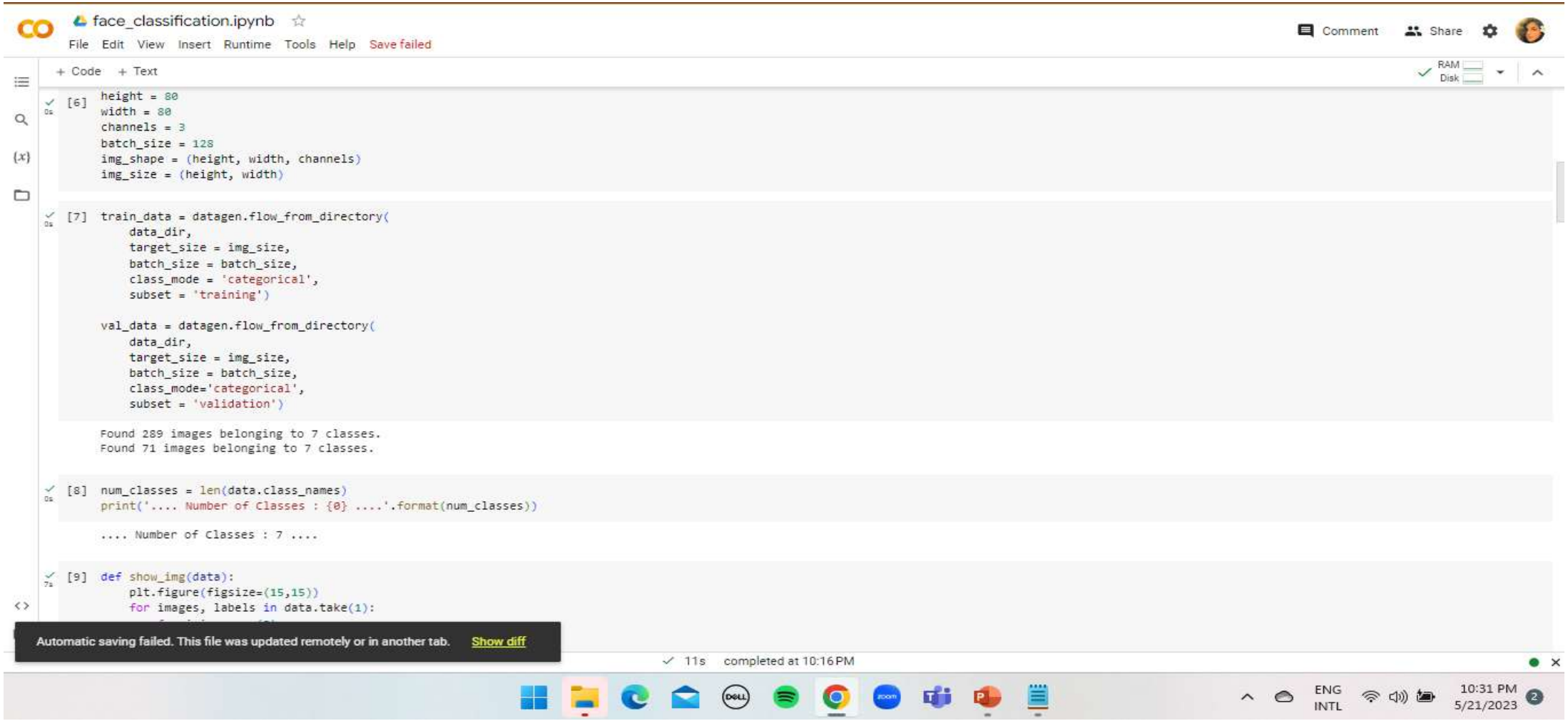
```
Found 360 files belonging to 7 classes.
```

```
[5] datagen = ImageDataGenerator(
            rescale = 1./255,
            rotation_range=40,
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

11s completed at 10:16 PM

ENG INTL 10:31 PM 5/21/2023


face_classification.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment
Share
+ Code + Text
RAM
Disk
[6] height = 80
width = 80
channels = 3
batch_size = 128
img_shape = (height, width, channels)
img_size = (height, width)
[7] train_data = datagen.flow_from_directory(
data_dir,
target_size = img_size,
batch_size = batch_size,
class_mode = 'categorical',
subset = 'training')
val_data = datagen.flow_from_directory(
data_dir,
target_size = img_size,
batch_size = batch_size,
class_mode='categorical',
subset = 'validation')
Found 289 images belonging to 7 classes.
Found 71 images belonging to 7 classes.
[8] num_classes = len(data.class_names)
print('.... Number of Classes : {0} ....'.format(num_classes))
.... Number of Classes : 7 ....
[9] def show_img(data):
plt.figure(figsize=(15,15))
for images, labels in data.take(1):
Automatic saving failed. This file was updated remotely or in another tab.
Show diff
11s completed at 10:16 PM
ENG
INTL
10:31 PM
5/21/2023

face_classification.ipynb

File Edit View Insert Runtime Tools Help Last saved at 12:02 AM

Comment Share Connect

+ Code + Text

```
[ ] num_classes = len(data.class_names)
    print('.... Number of Classes : {0} ....'.format(num_classes))
```

```
.... Number of Classes : 7 ....
```

```
[ ] def show_img(data):
        plt.figure(figsize=(15,15))
        for images, labels in data.take(1):
            for i in range(9):
                ax = plt.subplot(3, 3, i + 1)
                ax.imshow(images[i].numpy().astype("uint8"))
                ax.axis("off")

    show_img(data)
```

ENG INTL 12:31 AM 5/22/2023

face_classification.ipynb

File Edit View Insert Runtime Tools Help Last saved at 12:02 AM

Comment Share Connect

+ Code + Text

```
[ ] pre_trained = InceptionV3(weights='imagenet', include_top=False, input_shape=img_shape, pooling='avg')

    for layer in pre_trained.layers:
        layer.trainable = False
```

```
[ ] x = pre_trained.output
    x = BatchNormalization(axis=-1, momentum=0.99, epsilon=0.001)(x)
    x = Dropout(0.2)(x)
    x = Dense(1024, activation='relu')(x)
    x = Dropout(0.2)(x)
    predictions = Dense(num_classes, activation='softmax')(x)

    model = Model(inputs = pre_trained.input, outputs = predictions)
    model.compile(optimizer = Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```
[ ] model.summary()
```

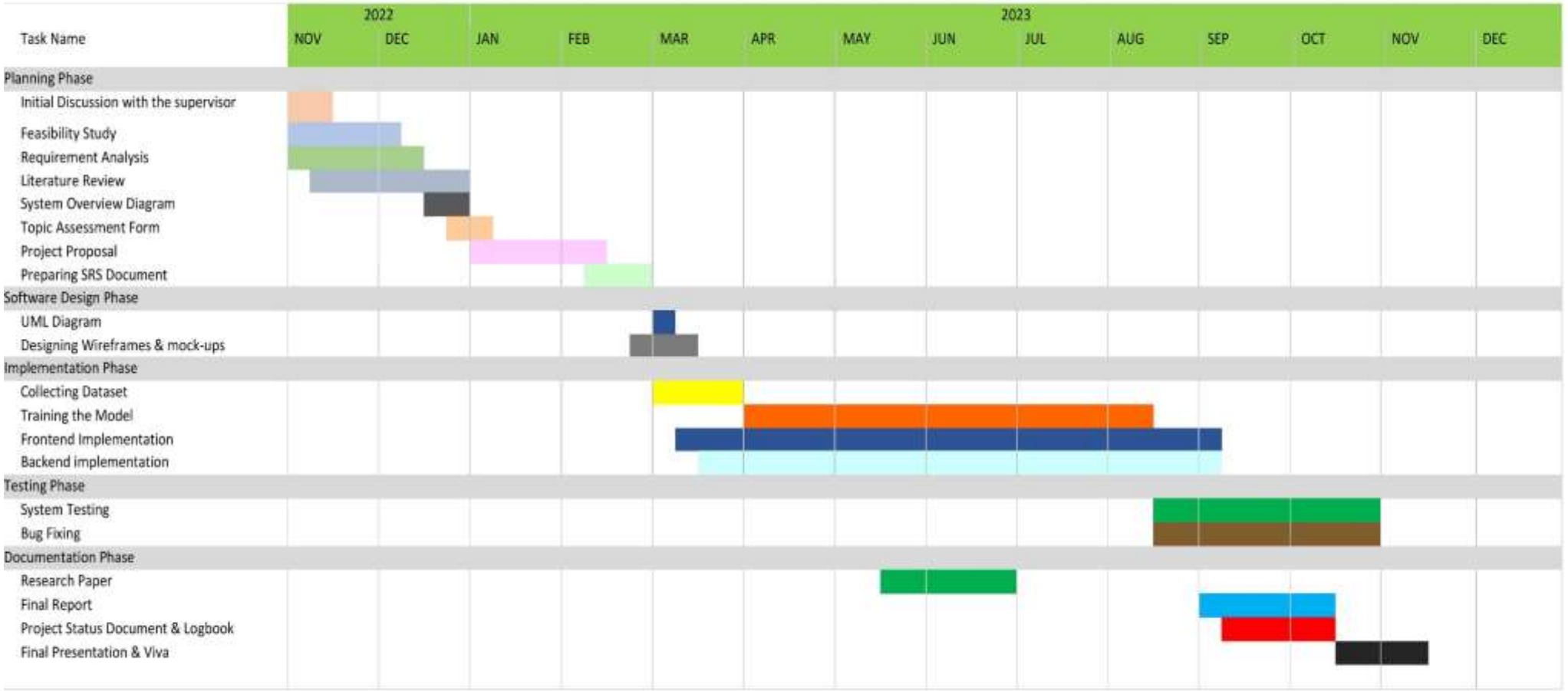
```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 80, 80, 3)]  0           []

 conv2d (Conv2D)                (None, 39, 39, 32)   864         ['input_1[0][0]']

 batch_normalization (BatchNorm  (None, 39, 39, 32)  96          ['conv2d[0][0]']
 alization)

 activation (Activation)        (None, 39, 39, 32)   0           ['batch_normalization[0][0]']

 conv2d_1 (Conv2D)              (None, 37, 37, 32)   9216        ['activation[0][0]']

 batch_normalization_1 (BatchNo  (None, 37, 37, 32)  96          ['conv2d_1[0][0]']
 rmalization)

 activation_1 (Activation)      (None, 37, 37, 32)   0           ['batch_normalization_1[0][0]']
```

ENG INTL 12:32 AM 5/22/2023

TASK TO BE DONE

- Fine-tune the model to improve its accuracy.
- Front-end mobile application implementation
- Implementation of the application's back end
- Integration of Systems
- System testing from start to finish

GANTT CHART


Task Name
2022
NOV
DEC
2023
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
Planning Phase
Initial Discussion with the supervisor
Feasibility Study
Requirement Analysis
Literature Review
System Overview Diagram
Topic Assessment Form
Project Proposal
Preparing SRS Document
Software Design Phase
UML Diagram
Designing Wireframes & mock-ups
Implementation Phase
Collecting Dataset
Training the Model
Frontend Implementation
Backend implementation
Testing Phase
System Testing
Bug Fixing
Documentation Phase
Research Paper
Final Report
Project Status Document & Logbook
Final Presentation & Viva

REFERENCES

- [1] S. M, "Industry 4.0 and lean management:," a proposed integration model and research propositions, vol. 6, no. 1, pp. 416-432, January 2018.
- [2] S. Ziyadin, O. Litvishko, M. Dubrova, G. Smagulova and M. Suyunchaliyeva, "Diversification tourism in the conditions of the digitalization," vol. 10, no. 02, pp. 1055-1070, February 2019.
- [3] X. Zheng, Y. Luo, Z. Xu, Q. Yu and L. Lu, "Tourism Destination Recommender System for the Cold Start Problem," KSII Transactions on Internet and Information Systems, vol. 10, no. 8, pp. 3192-3212, August 2016.
- [4] E. T. S and V. C, "Understanding Personalization of Recommender System: A Domain," vol. 13, no. 15, pp. 12422-12428, 2018.
- [5] WTO, "A Practical Guide to Tourism Destination Management," World Tourism Organization, 2007.
- [6] B. Cvetkovic, R. Szeklicki, V. Janko, . P. Lutomski and M. Lustrek , "Real-time activity monitoring with a wristband and a smartphone.," Information Fusion, vol. 43, pp. 77-93, 2018.
- [7] L. Santamaria-Granados, J. Mendoza-Moreno, A. ChantreAstaiza and M. Munoz-Organero, "Tourist Experiences Recommender System Based on Emotion Recognition with Wearable Data," Sensors, vol. 21, p. 7854, 2021.
- [8] L. Santamaria-Granados, J. Mendoza-Moreno and G. Ramirez-Gonzalez, "Tourist Recommender Systems Based on Emotion Recognition," A Scientometric Review, vol. 13, no. 2, 2021.