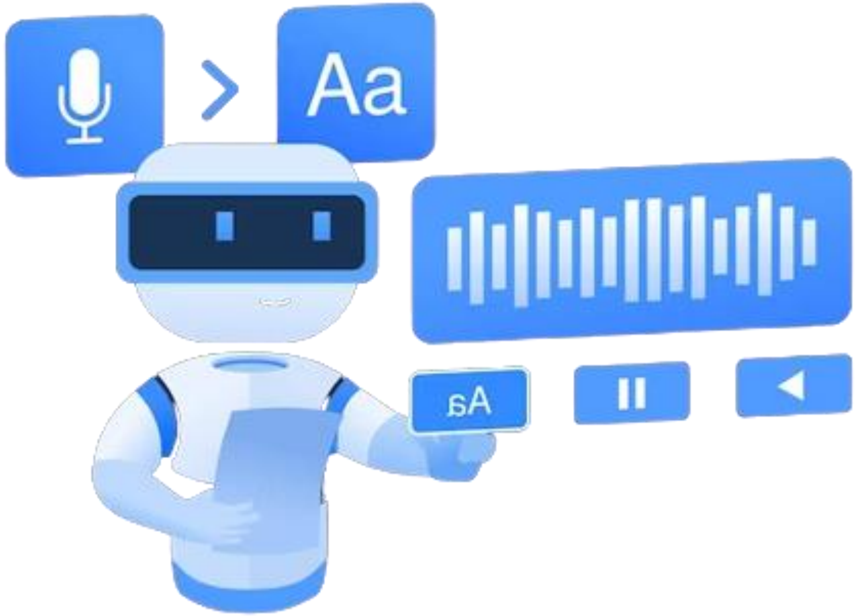
IT20259266 | Ranasinghe P.R.K.U

Software Engineering



AI-Based Chatbot for Native Speakers

Research Problem

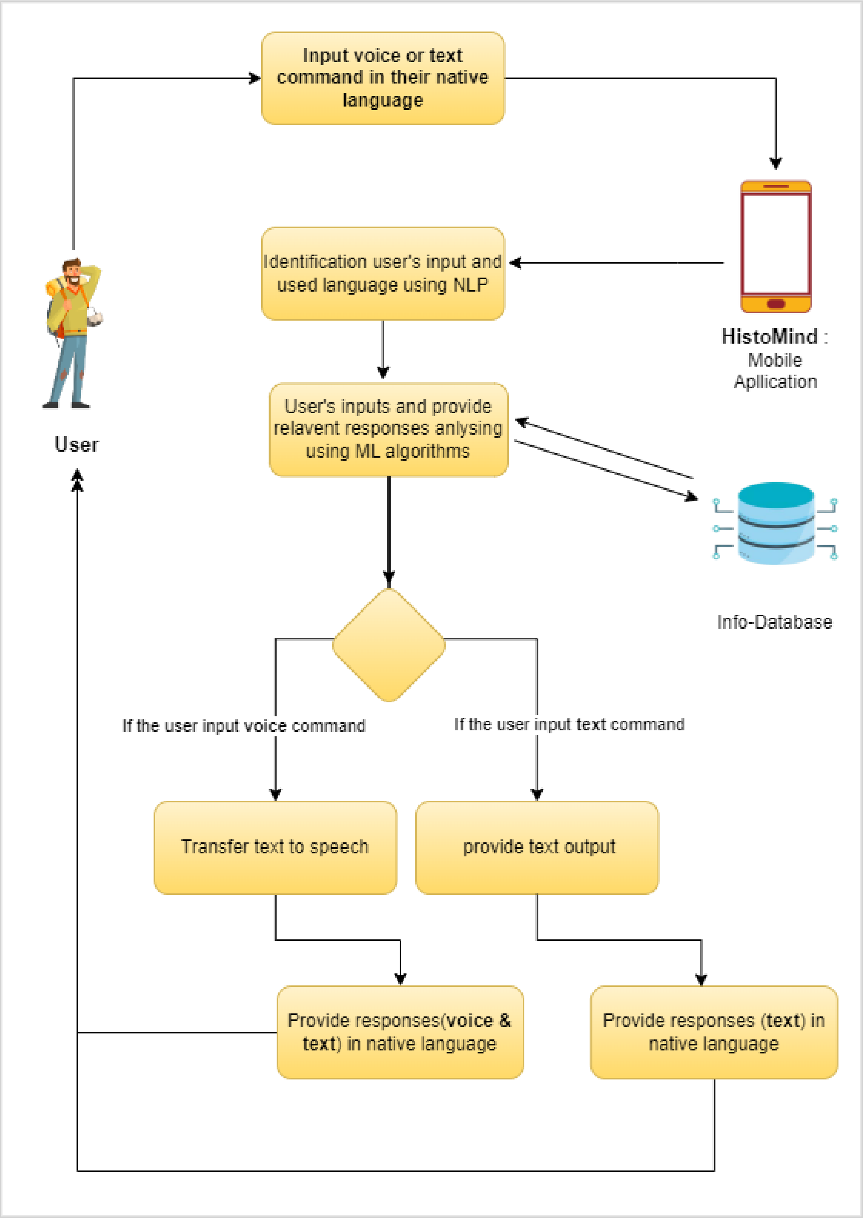
- ✓ The Sri Lankan economy heavily depends on tourism, with **historical sites** being major attractions.

- ✓ **Language limitations** pose difficulties for tourists in interacting with locals and **obtaining accurate information about historical sites.**

- ✓ **Lack of information in languages** like German and Tamil, commonly used by tourists visiting Sri Lanka.

- ✓ Existing information in widely spoken languages like English does not cater to the diverse range of visitors.

User Flow Diagram


Input voice or text command in their native language
HistoMind : Mobile Apllication
Identification user's input and used language using NLP
User's inputs and provide relavent responses anlysing using ML algorithms
Info-Database
User
If the user input voice command
If the user input text command
Transfer text to speech
provide text output
Provide responses(voice & text) in native language
Provide responses (text) in native language

Functional Requirements

- **Language Recognition** - The chatbot should be able to differentiate between different user-spoken languages.
- **Intent Recognition** - The chatbot needs to understand what the user wants and respond accordingly.
- **Knowledge Management** - The chatbot needs to save and retrieve information from a database to give accurate responses
- **Error handling** - The chatbot should handle errors well and provide feedback to users.
- **TESTING**- The system shall undergo regular testing to improve its performance and usability

Current Progress

- Create a **Language Identification Model** using sample dataset(English, German)

- Create Mobile User Interfaces

- Gathered voice datasets for creating a Voice Recognition Model

Methodology

- **Data Collection:** Collect and prepare a dataset of tourist questions and answers related to historical places and landmarks in various languages.

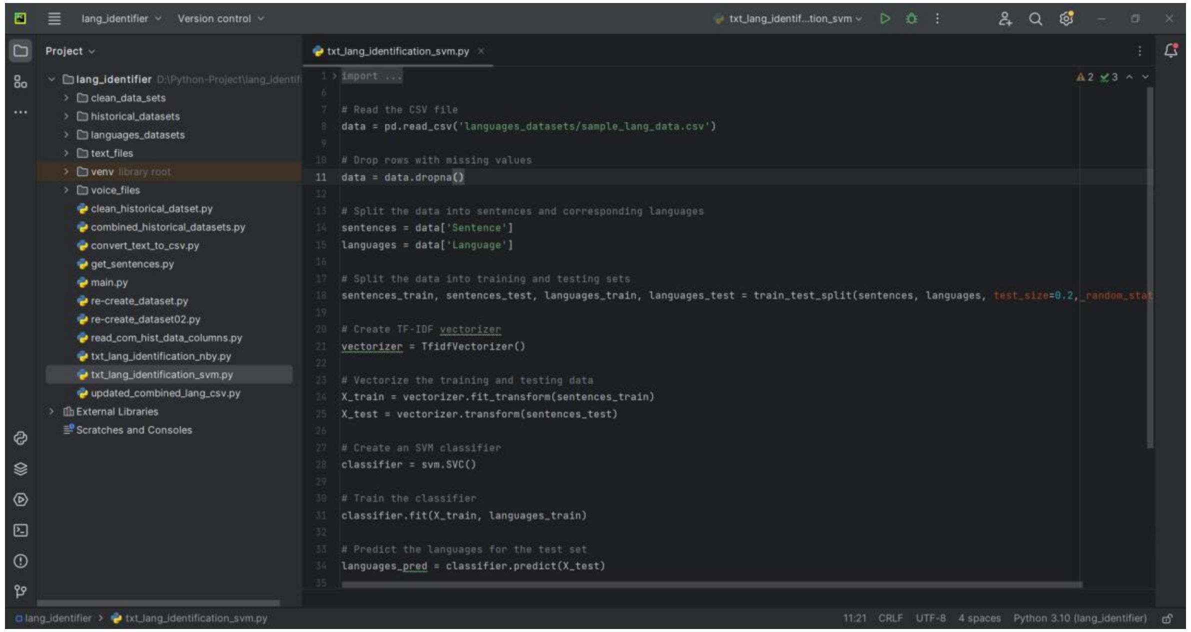
- **Model Development:** Train and optimize an AI-based chatbot model using NLP and ML techniques on the collected dataset.

- **Integration:** Integrate the trained model into a mobile application, allowing tourists to speak or type in their native language, which the chatbot will recognize automatically.

- **Testing:** Conduct thorough testing of the chatbot's functionality and accuracy to ensure it provides accurate and helpful responses to tourist inquiries.

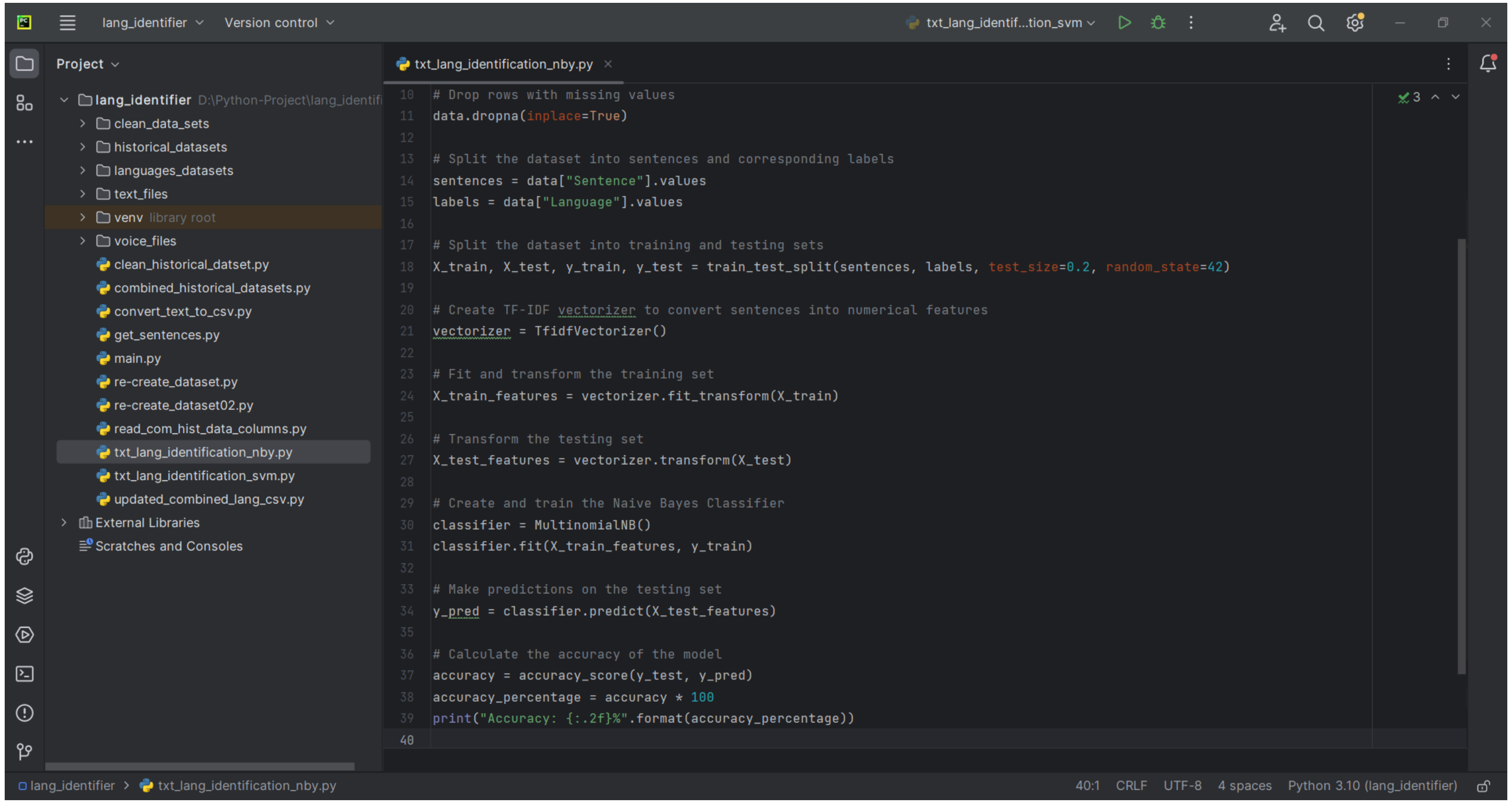
- **Deployment:** Deploy the mobile application with the AI-based chatbot to make it available to tourists and other users, with regular updates and improvements based on user feedback.

Training the dataset - SVM


lang_identifier
Version control
txt_lang_identif...tion_svm
txt_lang_identification_svm.py
Project
lang_identifier D:\Python-Project\lang_identifi
clean_data_sets
historical_datasets
languages_datasets
text_files
venv library root
voice_files
clean_historical_datset.py
combined_historical_datasets.py
convert_text_to_csv.py
get_sentences.py
main.py
re-create_dataset.py
re-create_dataset02.py
read_com_hist_data_columns.py
txt_lang_identification_nby.py
txt_lang_identification_svm.py
updated_combined_lang_csv.py
External Libraries
Scratches and Consoles
11:21
CRLF
UTF-8
4 spaces
Python 3.10 (lang_identifier)

```python
import ...

# Read the CSV file
data = pd.read_csv('languages_datasets/sample_lang_data.csv')

# Drop rows with missing values
data = data.dropna()

# Split the data into sentences and corresponding languages
sentences = data['Sentence']
languages = data['Language']

# Split the data into training and testing sets
sentences_train, sentences_test, languages_train, languages_test = train_test_split(sentences, languages, test_size=0.2, random_stat

# Create TF-IDF vectorizer
vectorizer = TfidfVectorizer()

# Vectorize the training and testing data
X_train = vectorizer.fit_transform(sentences_train)
X_test = vectorizer.transform(sentences_test)

# Create an SVM classifier
classifier = svm.SVC()

# Train the classifier
classifier.fit(X_train, languages_train)

# Predict the languages for the test set
languages_pred = classifier.predict(X_test)
```

Training the dataset - Naive Bayes Classifier


lang_identifier
Version control
txt_lang_identif...tion_svm
Project
txt_lang_identification_nby.py
lang_identifier D:\Python-Project\lang_identifi
clean_data_sets
historical_datasets
languages_datasets
text_files
venv library root
voice_files
clean_historical_datset.py
combined_historical_datasets.py
convert_text_to_csv.py
get_sentences.py
main.py
re-create_dataset.py
re-create_dataset02.py
read_com_hist_data_columns.py
txt_lang_identification_nby.py
txt_lang_identification_svm.py
updated_combined_lang_csv.py
External Libraries
Scratches and Consoles
3
lang_identifier
txt_lang_identification_nby.py
40:1
CRLF
UTF-8
4 spaces
Python 3.10 (lang_identifier)

```python
# Drop rows with missing values
data.dropna(inplace=True)

# Split the dataset into sentences and corresponding labels
sentences = data["Sentence"].values
labels = data["Language"].values

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(sentences, labels, test_size=0.2, random_state=42)

# Create TF-IDF vectorizer to convert sentences into numerical features
vectorizer = TfidfVectorizer()

# Fit and transform the training set
X_train_features = vectorizer.fit_transform(X_train)

# Transform the testing set
X_test_features = vectorizer.transform(X_test)

# Create and train the Naive Bayes Classifier
classifier = MultinomialNB()
classifier.fit(X_train_features, y_train)

# Make predictions on the testing set
y_pred = classifier.predict(X_test_features)

# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
accuracy_percentage = accuracy * 100
print("Accuracy: {:.2f}%".format(accuracy_percentage))

```

Challenges and Risk Mitigation

- Collect a substantial amount of data for the dataset.
- Language Limitations : language processing and translation accuracy
- Speech Recognition : variations in accents, background noise, and speech clarity
- Training the sample model.

Future Work

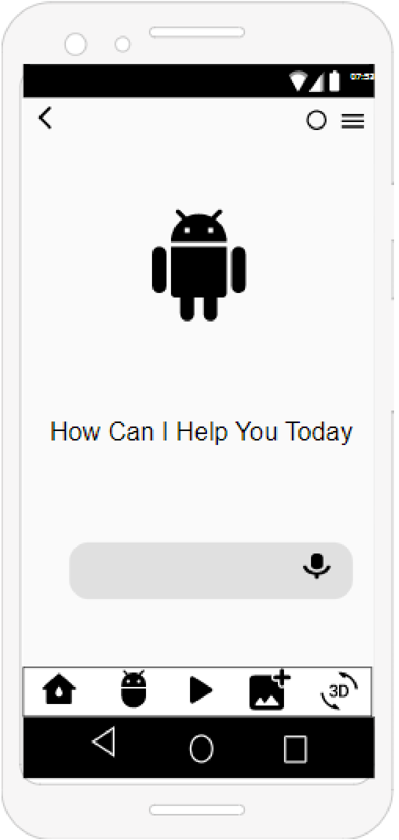
For 90% Progress Presentation

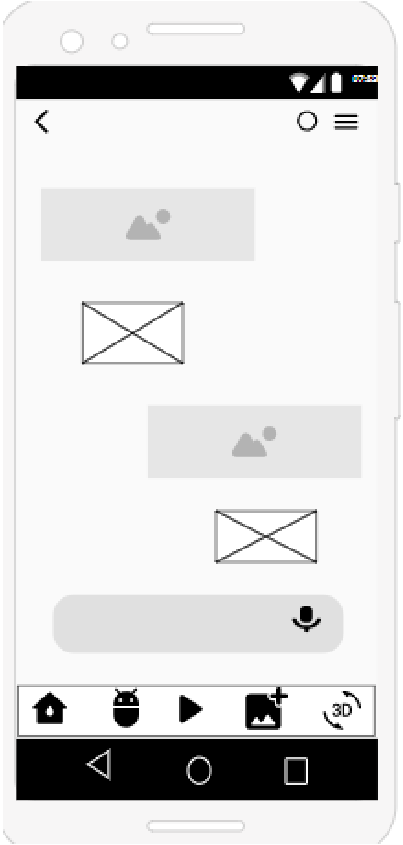
- ✓ Increase the model accuracy level by using a large dataset.
- ✓ Complete the Backend.
- ✓ Complete the Front-End using Flutter.

For Final Presentation

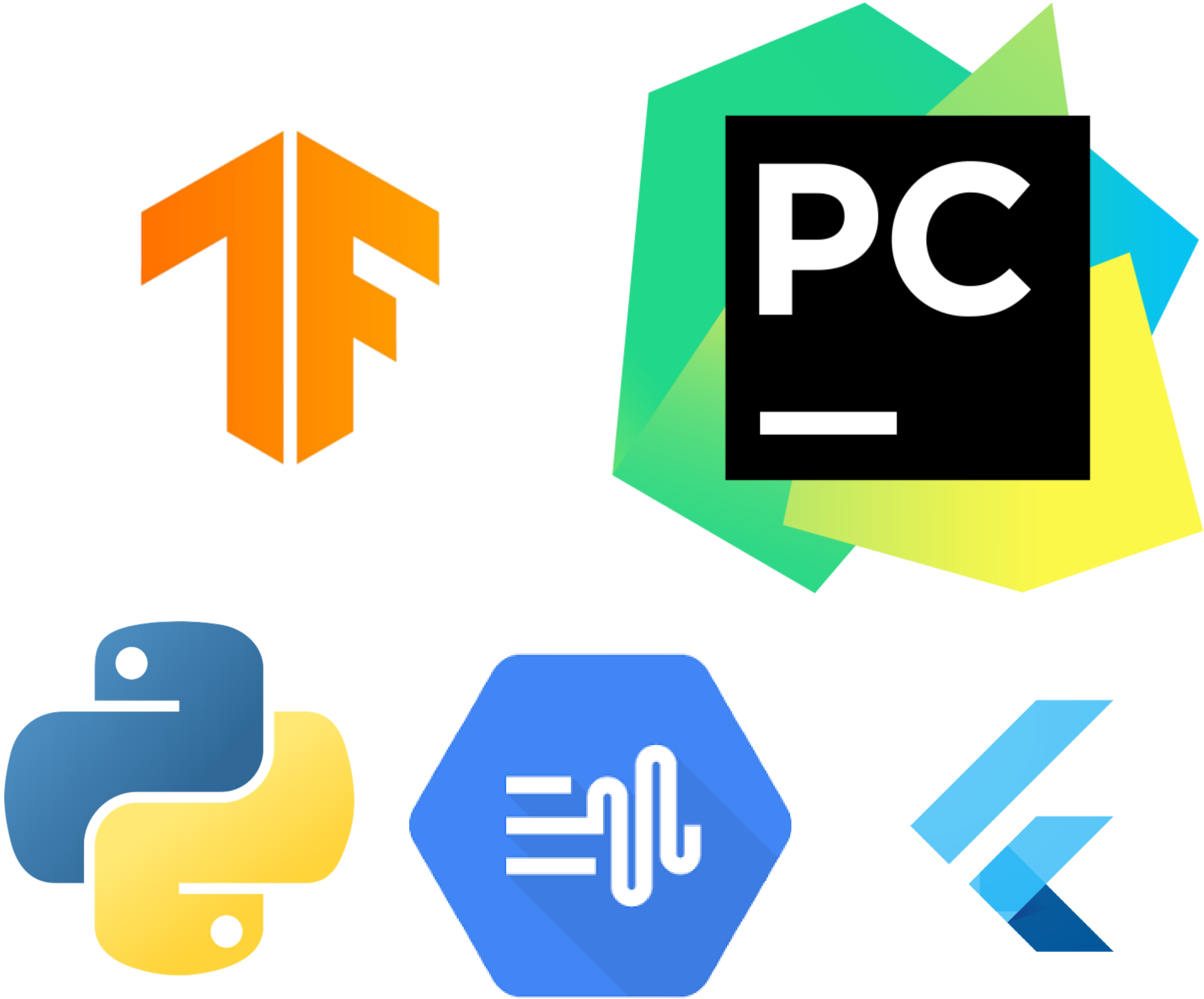
- ✓ Integrate the component with other team members.
- ✓ Complete running app.

Interface Designs

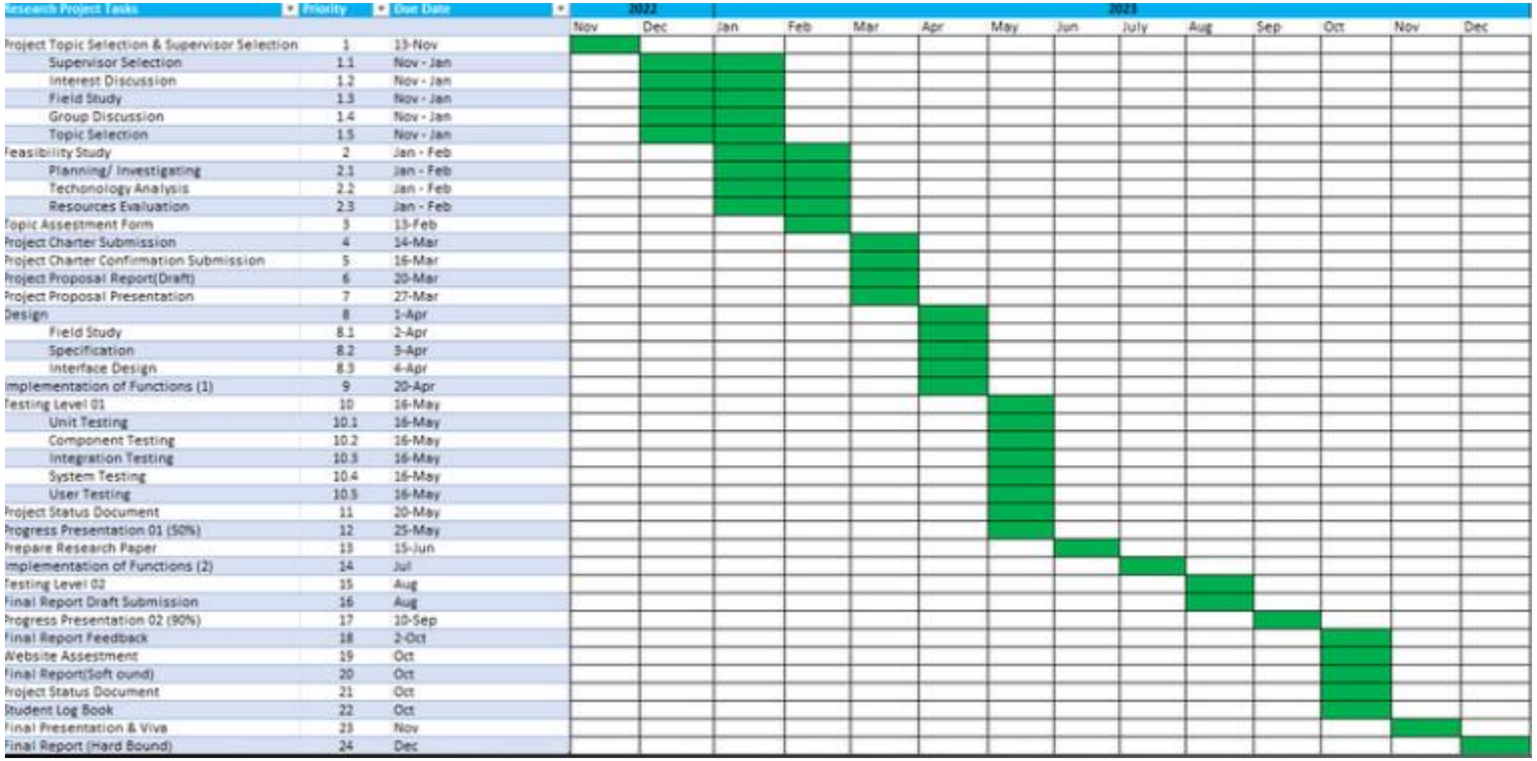




Tools & Technologies



Gantt Chart



esearch Project Tasks	Priority	Due Date	2022		2023											
			Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	July	Aug	Sep	Oct	Nov	Dec
roject Topic Selection & Supervisor Selection	1	13-Nov	🟩													
Supervisor Selection	1.1	Nov - Jan		🟩	🟩											
Interest Discussion	1.2	Nov - Jan		🟩	🟩											
Field Study	1.3	Nov - Jan		🟩	🟩											
Group Discussion	1.4	Nov - Jan		🟩	🟩											
Topic Selection	1.5	Nov - Jan		🟩	🟩											
easibility Study	2	Jan - Feb			🟩	🟩										
Planning/ Investigating	2.1	Jan - Feb			🟩	🟩										
Techonology Analysis	2.2	Jan - Feb			🟩	🟩										
Resources Evaluation	2.3	Jan - Feb			🟩	🟩										
opic Assesment Form	3	13-Feb				🟩										
roject Charter Submission	4	14-Mar					🟩									
roject Charter Confirmation Submission	5	16-Mar					🟩									
roject Proposal Report(Draft)	6	20-Mar					🟩									
roject Proposal Presentation	7	27-Mar					🟩									
esign	8	1-Apr						🟩								
Field Study	8.1	2-Apr						🟩								
Specification	8.2	3-Apr						🟩								
Interface Design	8.3	4-Apr						🟩								
mplementation of Functions (1)	9	20-Apr						🟩								
esting Level 01	10	16-May							🟩							
Unit Testing	10.1	16-May							🟩							
Component Testing	10.2	16-May							🟩							
Integration Testing	10.3	16-May							🟩							
System Testing	10.4	16-May							🟩							
User Testing	10.5	16-May							🟩							
roject Status Document	11	20-May							🟩							
rogress Presentation 01 (50%)	12	25-May							🟩							
repare Research Paper	13	15-Jun								🟩						
mplementation of Functions (2)	14	Jul									🟩					
esting Level 02	15	Aug										🟩				
inal Report Draft Submission	16	Aug										🟩				
rogress Presentation 02 (90%)	17	10-Sep											🟩			
inal Report Feedback	18	2-Oct												🟩		
Vebsite Assestment	19	Oct												🟩		
inal Report(Soft ound)	20	Oct												🟩		
roject Status Document	21	Oct												🟩		
tudent Log Book	22	Oct												🟩		
inal Presentation & Viva	23	Nov													🟩	
inal Report (Hard Bound)	24	Dec														🟩

Model Demonstration

IT20008178 | PRIYARATNE K.K.M.M

Information Technology

Functional Requirements

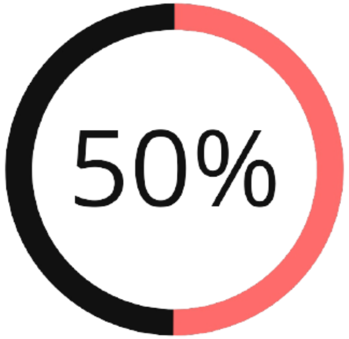
- The system should be able to collect a set of images of historical places from different sources, such as online databases, user submissions, or social media.
- The location should be able to be found by the system. matching the database's information.
- The system should be able to store the images and associated metadata, such as the location and date of the image, to enable future retrieval and analysis.
- Based on user comments and new data, the system should be able to employ methods of machine learning to increase its accuracy and effectiveness over time.

Research Problem

Our research aims to develop a reliable and accurate image recognition system for identifying and categorizing historical landmarks. We will address challenges such as variations in lighting, historical changes, and cultural differences to create a dependable system that works consistently in real-world contexts.

Current Progress

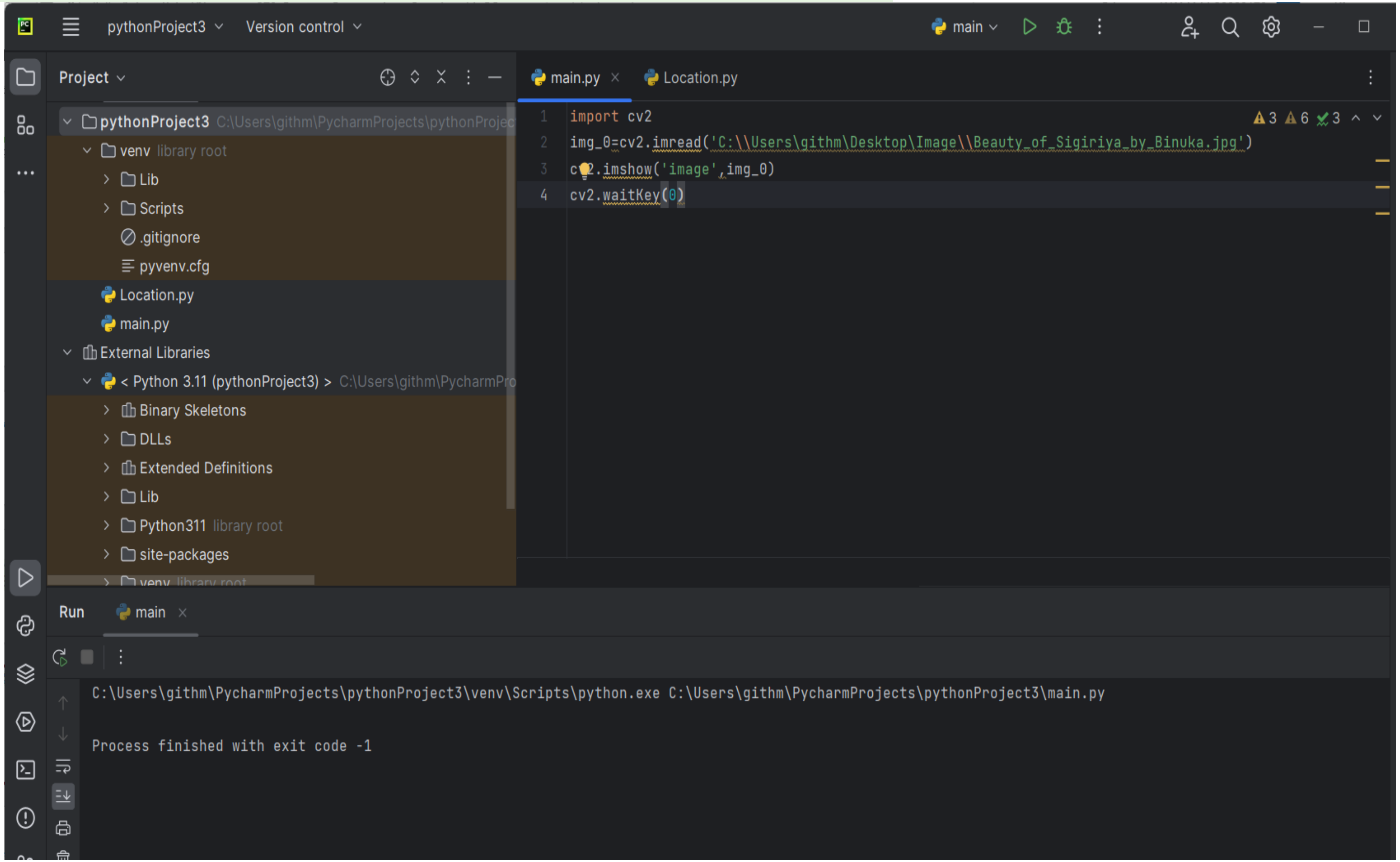
- Gather dataset for Sigiriya Rock fortress.
- Create Mobile user interfaces.
- Create demo Application using OpenCV.



Methodology

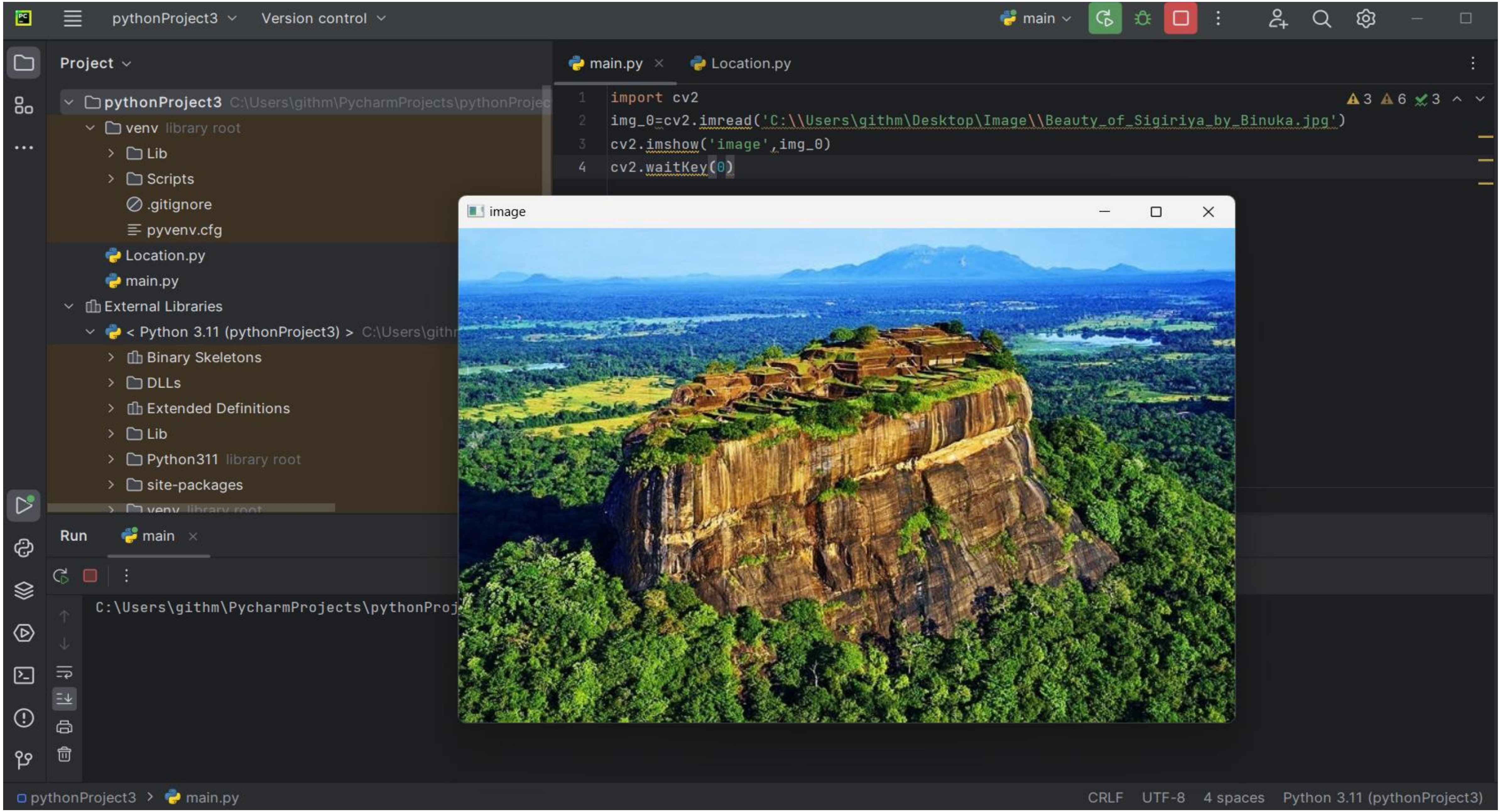
The methodology involves a comprehensive approach to developing an accurate and reliable image recognition system for identifying historical sites and landmarks. This involves various steps including data collection, preprocessing, feature extraction, algorithm development, training and testing, integration, collaboration, and continuous improvement. By following this methodology, researchers can develop a robust system that can contribute to the documentation and preservation of cultural heritage.

4.Out put of the create model



```python
import cv2
img_0=cv2.imread('C:\\Users\githm\Desktop\Image\\Beauty_of_Sigiriya_by_Binuka.jpg')
cv2.imshow('image',img_0)
cv2.waitKey(0)
```

```
C:\Users\githm\PycharmProjects\pythonProject3\venv\Scripts\python.exe C:\Users\githm\PycharmProjects\pythonProject3\main.py

Process finished with exit code -1
```



```python
import cv2
img_0=cv2.imread('C:\\Users\githm\Desktop\Image\\Beauty_of_Sigiriya_by_Binuka.jpg')
cv2.imshow('image',img_0)
cv2.waitKey(0)
```

```
C:\Users\githm\PycharmProjects\pythonProj
```

Challenges and Risk Mitigation

- Limited and Unjust- Dataset.
- Similar-looking Places.
- Generalization in Unknown Locations

Future Work

For 90% Progress Presentation

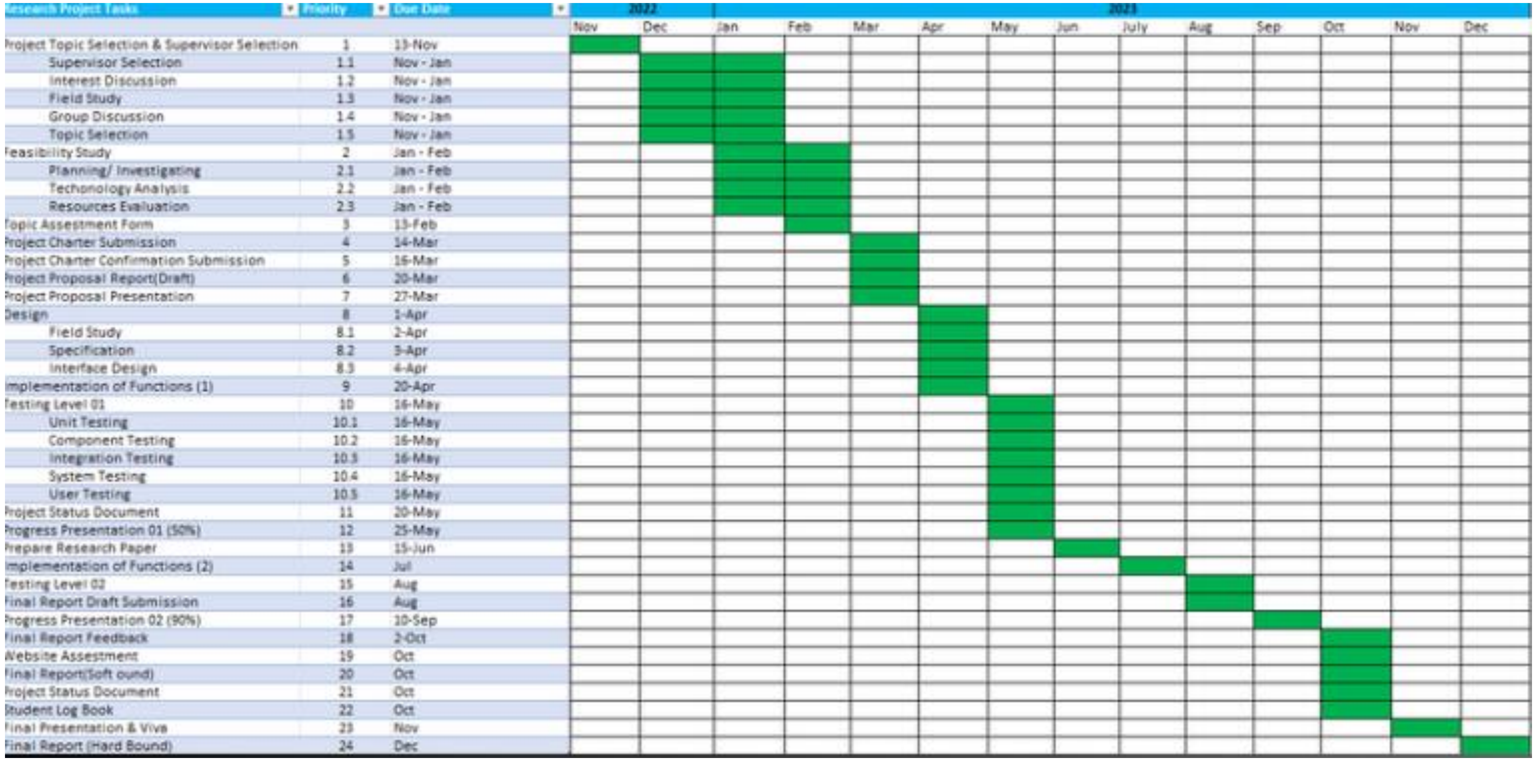
- ✓ Increase the model accuracy level by using a large dataset.
- ✓ Complete the Backend.
- ✓ Complete the Front-end using Flutter .

For Final Presentation

- ✓ Integrate the component with other team members.
- ✓ Complete running app.

Model Demonstration

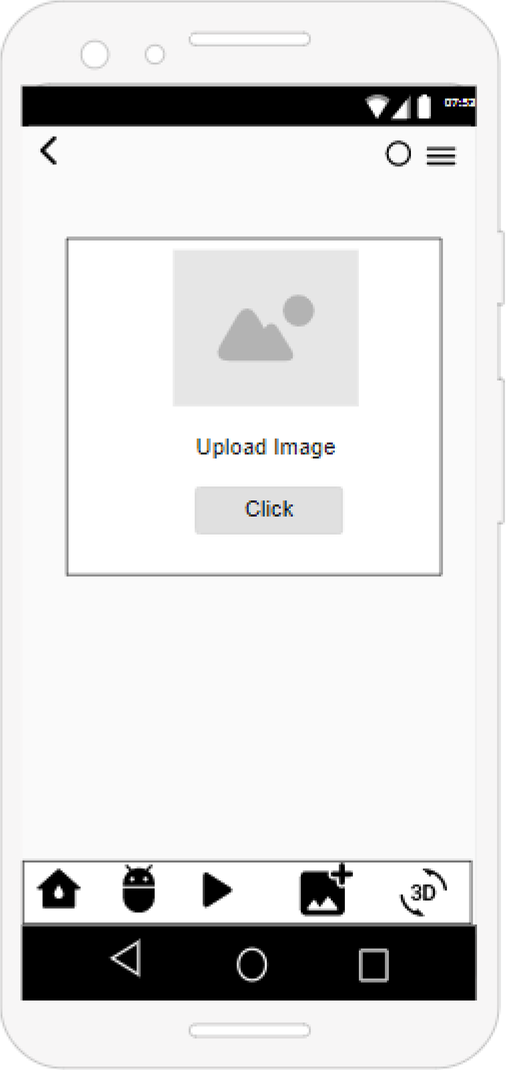
Gantt Chart

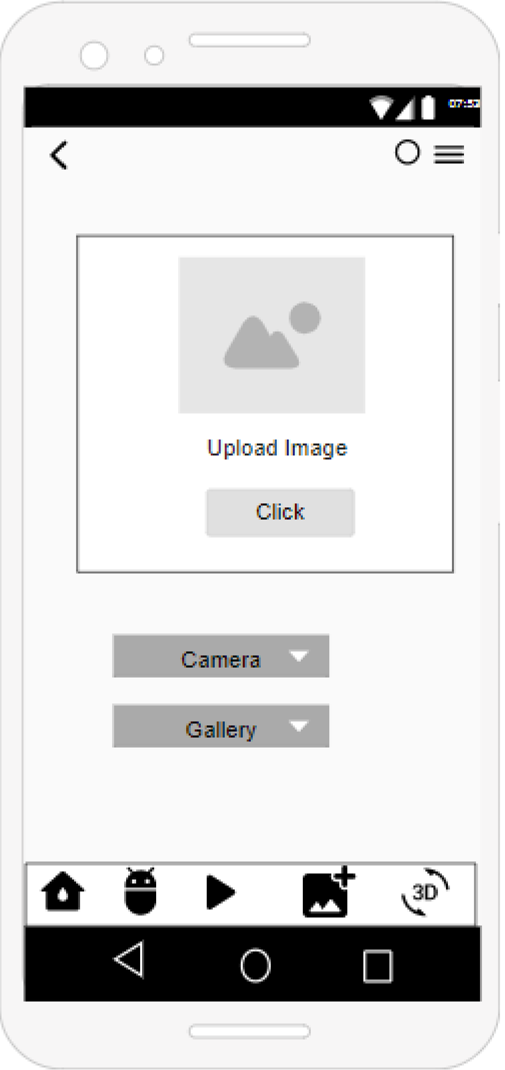


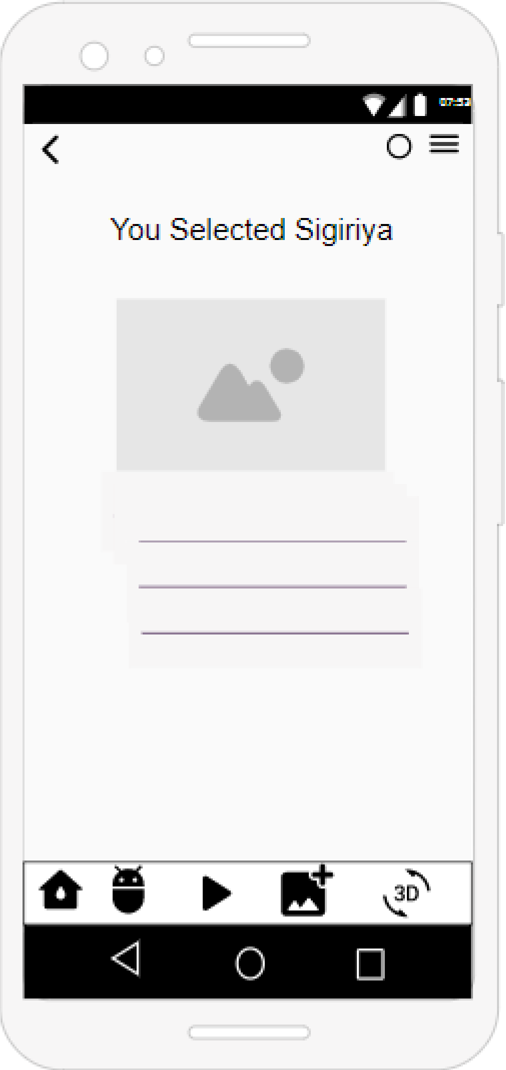
esearch Project Tasks	Priority	Due Date	2022		2023											
			Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	July	Aug	Sep	Oct	Nov	Dec
roject Topic Selection & Supervisor Selection	1	13-Nov	🟩													
Supervisor Selection	1.1	Nov - Jan		🟩	🟩											
Interest Discussion	1.2	Nov - Jan		🟩	🟩											
Field Study	1.3	Nov - Jan		🟩	🟩											
Group Discussion	1.4	Nov - Jan		🟩	🟩											
Topic Selection	1.5	Nov - Jan		🟩	🟩											
easibility Study	2	Jan - Feb			🟩	🟩										
Planning/ Investigating	2.1	Jan - Feb			🟩	🟩										
Techonology Analysis	2.2	Jan - Feb			🟩	🟩										
Resources Evaluation	2.3	Jan - Feb			🟩	🟩										
opic Assesment Form	3	13-Feb				🟩										
roject Charter Submission	4	14-Mar					🟩									
roject Charter Confirmation Submission	5	16-Mar					🟩									
roject Proposal Report(Draft)	6	20-Mar					🟩									
roject Proposal Presentation	7	27-Mar					🟩									
esign	8	1-Apr						🟩								
Field Study	8.1	2-Apr						🟩								
Specification	8.2	3-Apr						🟩								
Interface Design	8.3	4-Apr						🟩								
mplementation of Functions (1)	9	20-Apr						🟩								
esting Level 01	10	16-May							🟩							
Unit Testing	10.1	16-May							🟩							
Component Testing	10.2	16-May							🟩							
Integration Testing	10.3	16-May							🟩							
System Testing	10.4	16-May							🟩							
User Testing	10.5	16-May							🟩							
roject Status Document	11	20-May							🟩							
rogress Presentation 01 (50%)	12	25-May							🟩							
repare Research Paper	13	15-Jun								🟩						
mplementation of Functions (2)	14	Jul									🟩					
esting Level 02	15	Aug										🟩				
inal Report Draft Submission	16	Aug										🟩				
rogress Presentation 02 (90%)	17	10-Sep											🟩			
inal Report Feedback	18	2-Oct												🟩		
Vebsite Assestment	19	Oct												🟩		
inal Report(Soft ound)	20	Oct												🟩		
roject Status Document	21	Oct												🟩		
tudent Log Book	22	Oct												🟩		
inal Presentation & Viva	23	Nov													🟩	
inal Report (Hard Bound)	24	Dec														🟩

Technologies to be used

Interface Designs







Thank you

IT20209520 | Navoda R.C

Software Engineering

Research Problem

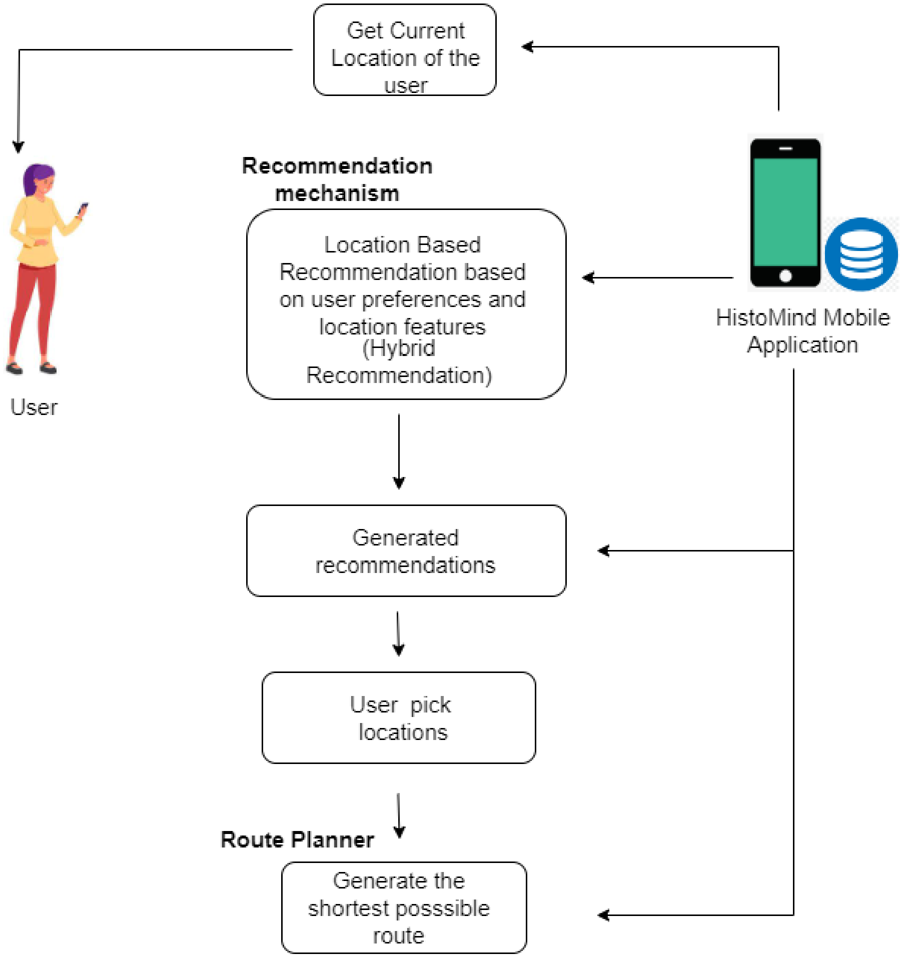
Limited software solutions:

Despite the country's rich historical sites, there is a lack of effective software solutions to promote and publicize these attractions to potential tourists. The existing methods are insufficient in leveraging technology to attract visitors.

Lack of cost-effective trip planning:

There is currently no comprehensive strategy in place that allows tourists to visit multiple historical sites at a lower cost within a single trip season. This hinders the potential for tourists to explore and experience the diverse cultural heritage of Sri Lanka.

User Flow Diagram


Get Current Location of the user
User
Recommendation mechanism
Location Based Recommendation based on user preferences and location features (Hybrid Recommendation)
HistoMind Mobile Application
Generated recommendations
User pick locations
Route Planner
Generate the shortest posssible route

Functional Requirements

Location-based recommendation mechanism:

The system should suggest historical places based on the user's current location.

It should also recommend nearby tourism attractions along with historical sites.

The recommendation algorithm should prioritize historical value over popularity.

Route Planner:

The system should allow users to select multiple historical locations on a map.

It should find the shortest path between all the destinations using an optimal shortest path algorithm.

Effective routing is crucial for efficient trip planning.

User Interface:

User Interface

- The mobile app should have a user-friendly interface.
- Proposed historical sites and nearby attractions should be displayed on the interface.
- The interface should provide mapping of selected historical locations and show the recommended path, with options for customization.

Optimization

- The system should optimize the shortest path algorithm to reduce computing costs.
- It should consider real-time traffic data to optimize the user's path for efficient navigation.
- Data Management:

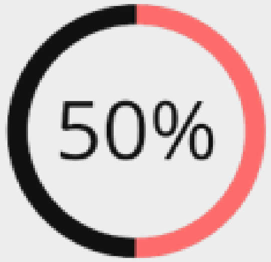
Methodology

- Implement a location-based recommendation mechanism considering the user's current location.
- Prioritize historical value over popularity in the recommendation algorithm.
- Develop a route planner to find the shortest path between selected historical locations.
- Optimize the shortest path algorithm to reduce computing costs.

- Create a user-friendly interface displaying proposed historical sites and nearby attractions.
- Enable customization options for the recommended path in the user interface.
- Ensure data accuracy by updating information on historical sites and tourist attractions.
- Implement effective data management techniques to reduce computational costs.
- Maintain privacy and security of user data through secure storage and transmission.
- Incorporate user authentication and access control in the system.

Current Progress

- Merge the datasets
- Feature engineering
  - ✓ Distance Calculation
  - ✓ Historical Place Categories
  - ✓ Ratings and Reviews
  - ✓ Historical Place Popularity
  - ✓ Time of Visit
- Handle categorical variables
- Normalize or scale the data
- Split the data
- Data Labeling
- Content-based filtering



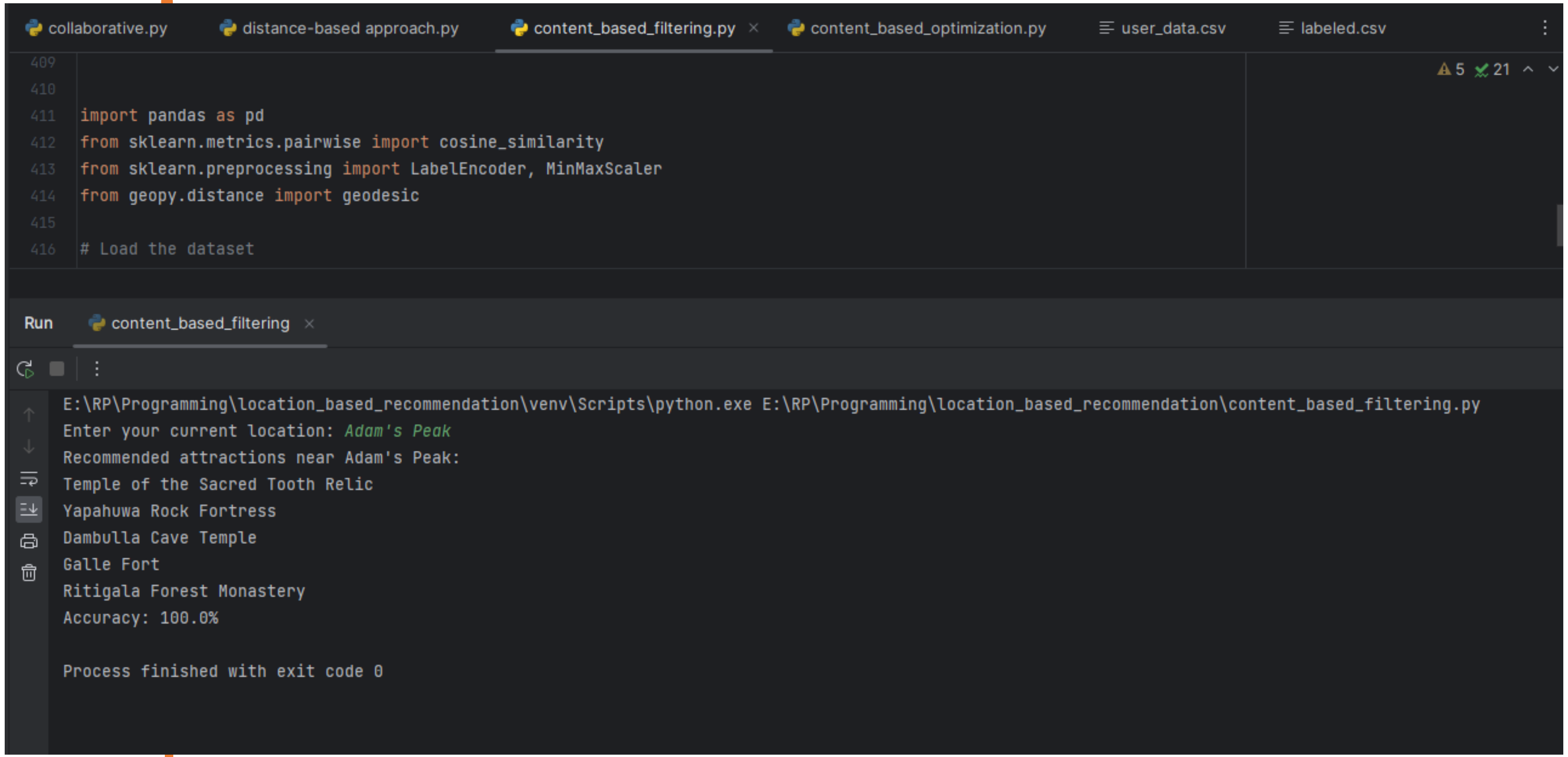
Optimization the content-based filtering

Instead of calculating distances using a loop, the apply function is used along with *geodesic* to calculate distances between the user's location and all other attractions. This eliminates the need for the distances list.

Sorted the attractions directly from the dataset based on distances. This avoids creating a separate list of attractions with distances and sorting it.

Instead of using if conditions and checking duplicates in a *loop*, the attractions are filtered to exclude the user's current location and directly selected the top *n_recommendations* attractions. This is done using pandas' DataFrame operations, which are more efficient.

Output of the Content-based Filtering

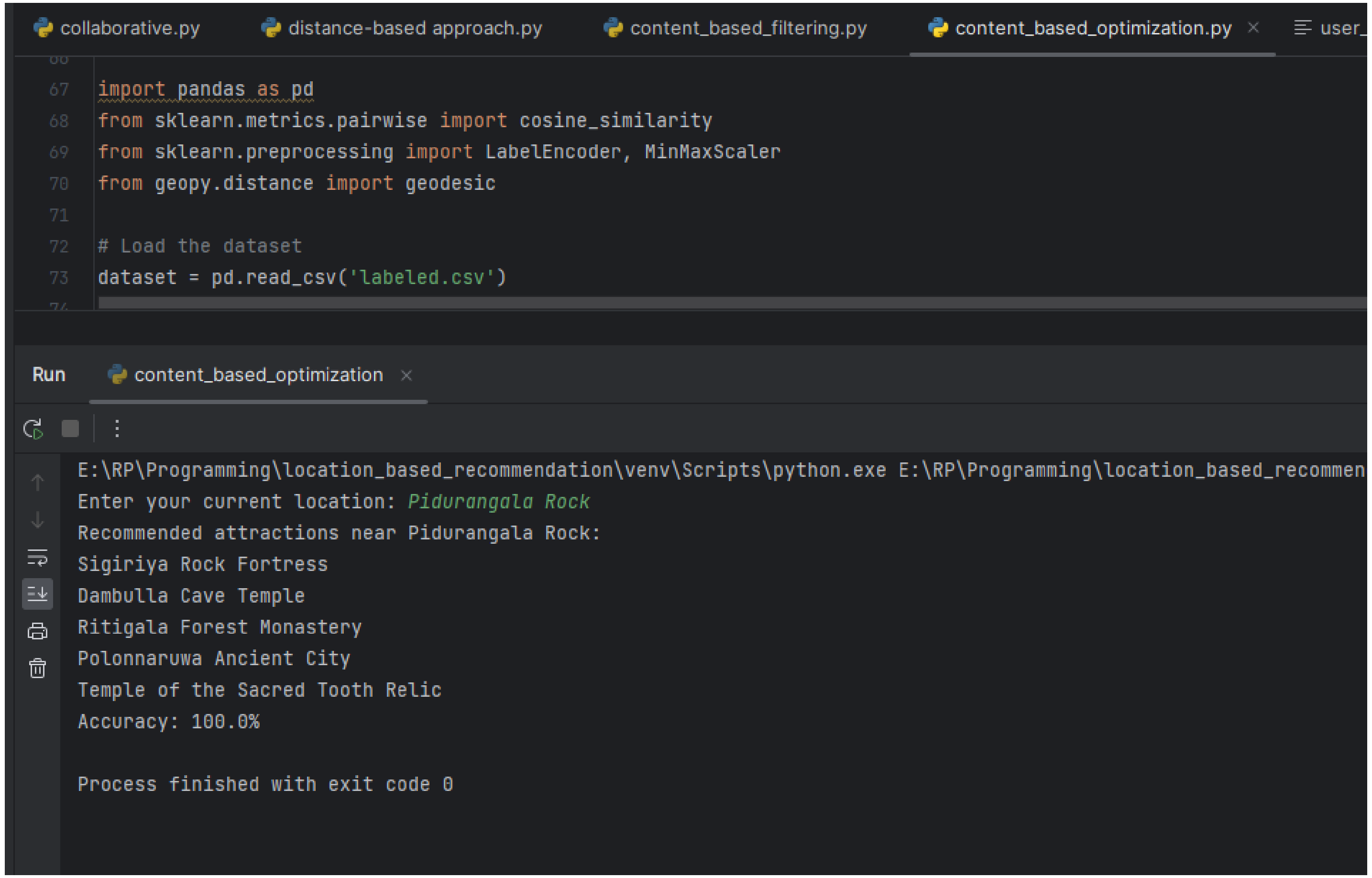


collaborative.py
distance-based approach.py
content_based_filtering.py
content_based_optimization.py
user_data.csv
labeled.csv
Run
content_based_filtering

```python
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from geopy.distance import geodesic

# Load the dataset
```

```
E:\RP\Programming\location_based_recommendation\venv\Scripts\python.exe E:\RP\Programming\location_based_recommendation\content_based_filtering.py
Enter your current location: Adam's Peak
Recommended attractions near Adam's Peak:
Temple of the Sacred Tooth Relic
Yapahuwa Rock Fortress
Dambulla Cave Temple
Galle Fort
Ritigala Forest Monastery
Accuracy: 100.0%

Process finished with exit code 0
```

Output of the Content-based Filtering Optimization


collaborative.py
distance-based approach.py
content_based_filtering.py
content_based_optimization.py
user_
Run
content_based_optimization

```python
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from geopy.distance import geodesic

# Load the dataset
dataset = pd.read_csv('labeled.csv')
```

```
E:\RP\Programming\location_based_recommendation\venv\Scripts\python.exe E:\RP\Programming\location_based_recommen
Enter your current location: Pidurangala Rock
Recommended attractions near Pidurangala Rock:
Sigiriya Rock Fortress
Dambulla Cave Temple
Ritigala Forest Monastery
Polonnaruwa Ancient City
Temple of the Sacred Tooth Relic
Accuracy: 100.0%

Process finished with exit code 0
```

Challenges and Risk Mitigation

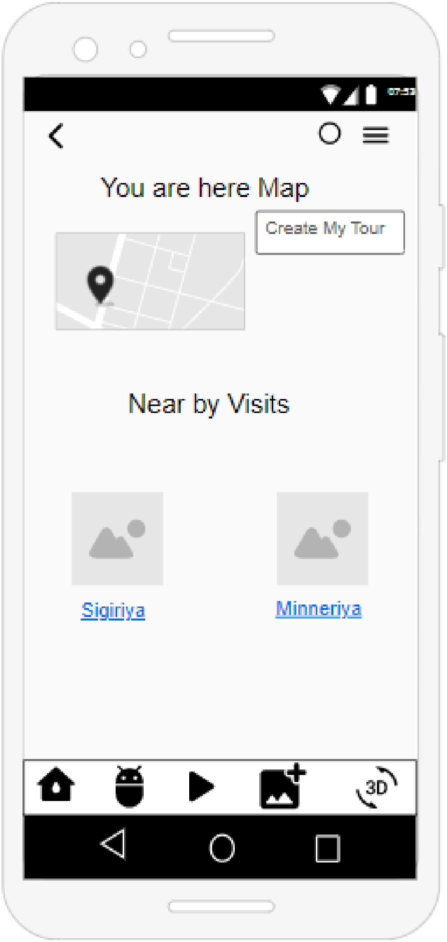
Challenges

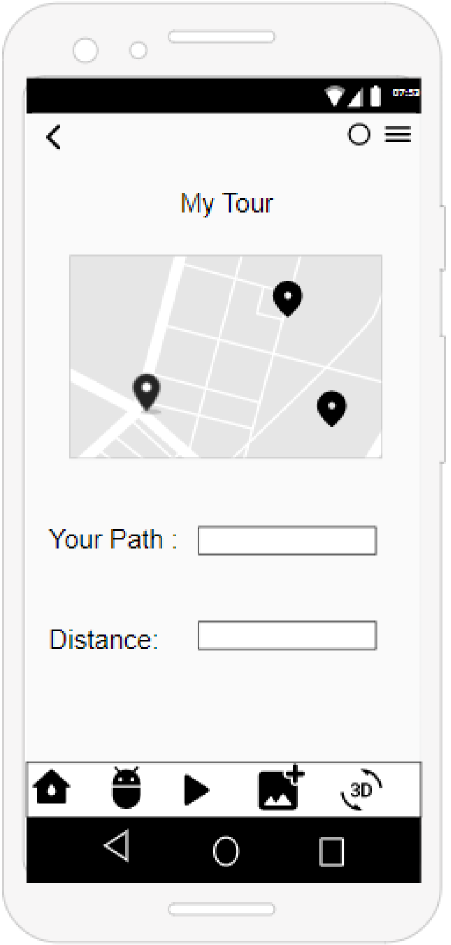
- Data Availability
- Algorithm Complexity
- Scalability
- User Experience

Risk Mitigation

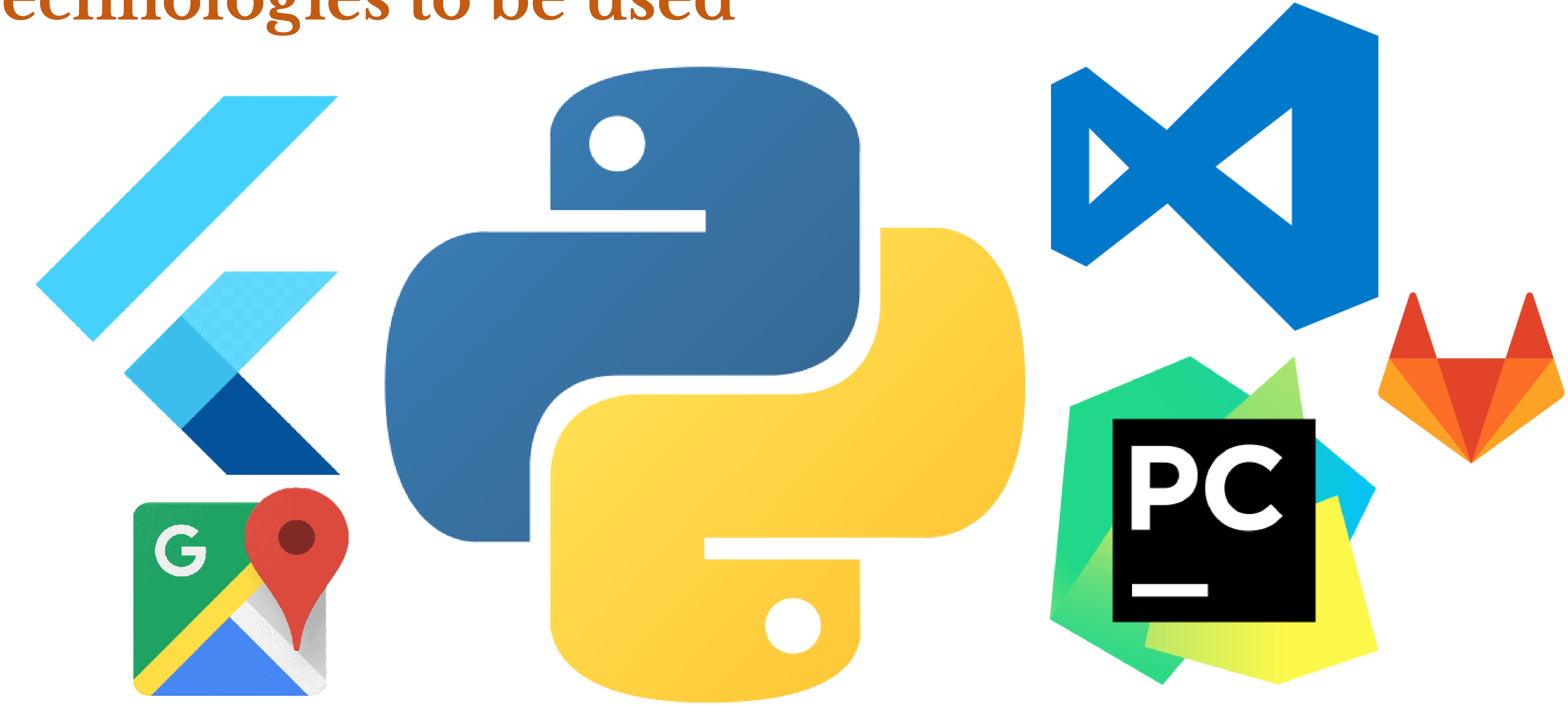
- Data Quality Control
- Algorithm Optimization
- Scalability Planning
- User Testing and Feedback
- Security and Privacy Measures
- Continuous Monitoring and Maintenance

Interface Designs





Technologies to be used



Future Work

- **For 90% Progress Presentation**

✓ Increase the model accuracy level by using a large dataset.

✓ Ternary recommendation based on the facts

  ▪ popular tourist attractions

  ▪ lesser-known attractions

  ▪ Neutral Recommendations

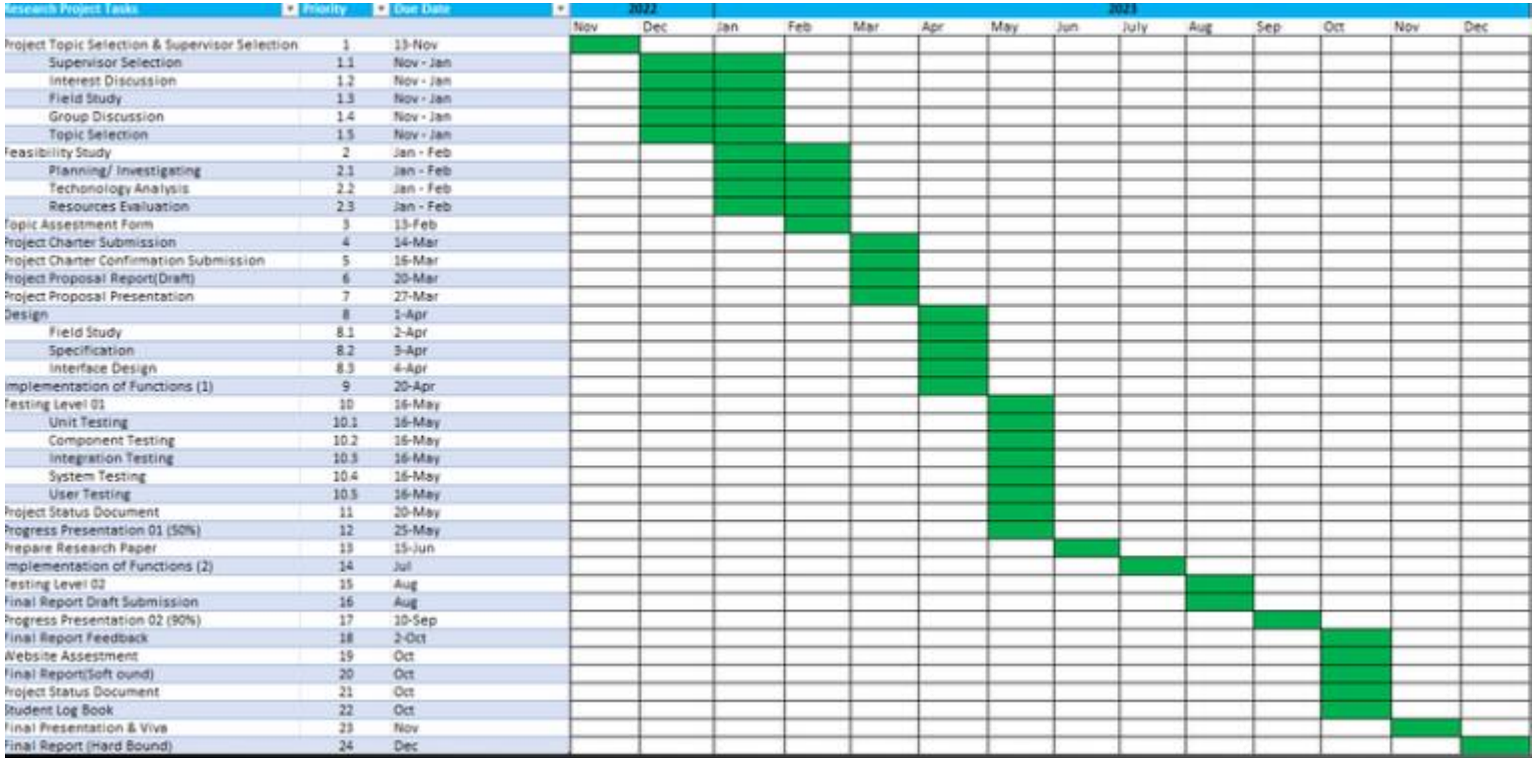
✓ Route planner and the shortest path optimization

- **For Final Presentation**

  ✓ Complete the frontend

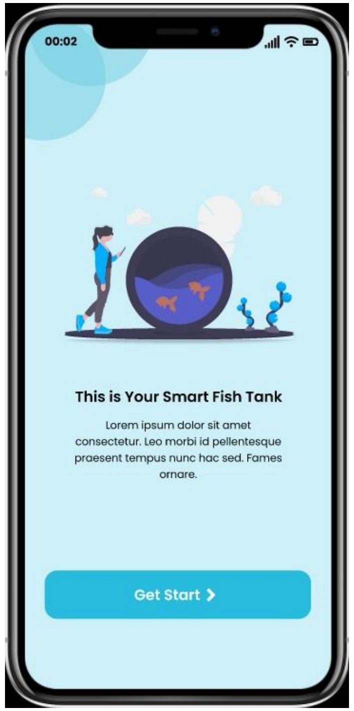
  ✓ Deployment

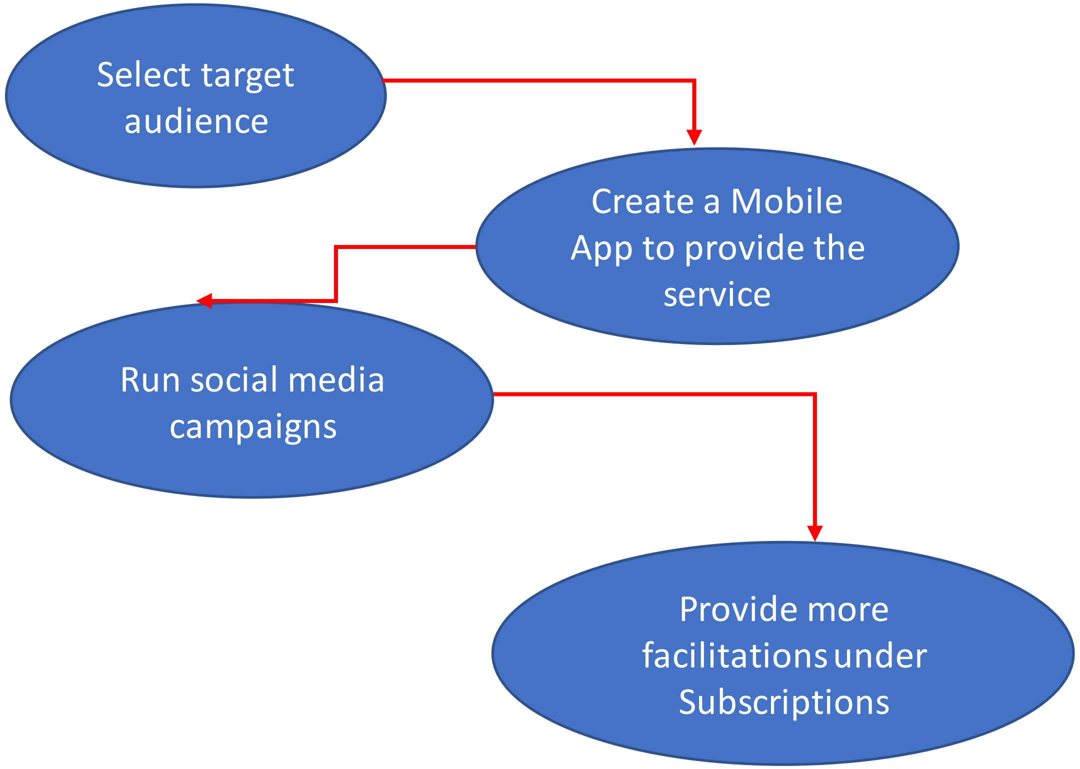
Gantt Chart



esearch Project Tasks	Priority	Due Date	2022		2023											
			Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	July	Aug	Sep	Oct	Nov	Dec
roject Topic Selection & Supervisor Selection	1	13-Nov	🟩													
Supervisor Selection	1.1	Nov - Jan		🟩	🟩											
Interest Discussion	1.2	Nov - Jan		🟩	🟩											
Field Study	1.3	Nov - Jan		🟩	🟩											
Group Discussion	1.4	Nov - Jan		🟩	🟩											
Topic Selection	1.5	Nov - Jan		🟩	🟩											
easibility Study	2	Jan - Feb			🟩	🟩										
Planning/ Investigating	2.1	Jan - Feb			🟩	🟩										
Techonology Analysis	2.2	Jan - Feb			🟩	🟩										
Resources Evaluation	2.3	Jan - Feb			🟩	🟩										
opic Assestment Form	3	13-Feb				🟩										
roject Charter Submission	4	14-Mar					🟩									
roject Charter Confirmation Submission	5	16-Mar					🟩									
roject Proposal Report(Draft)	6	20-Mar					🟩									
roject Proposal Presentation	7	27-Mar					🟩									
esign	8	1-Apr						🟩								
Field Study	8.1	2-Apr						🟩								
Specification	8.2	3-Apr						🟩								
Interface Design	8.3	4-Apr						🟩								
mplementation of Functions (1)	9	20-Apr						🟩								
esting Level 01	10	16-May							🟩							
Unit Testing	10.1	16-May							🟩							
Component Testing	10.2	16-May							🟩							
Integration Testing	10.3	16-May							🟩							
System Testing	10.4	16-May							🟩							
User Testing	10.5	16-May							🟩							
roject Status Document	11	20-May							🟩							
rogress Presentation 01 (50%)	12	25-May							🟩							
repare Research Paper	13	15-Jun								🟩						
mplementation of Functions (2)	14	Jul									🟩					
esting Level 02	15	Aug										🟩				
inal Report Draft Submission	16	Aug										🟩				
rogress Presentation 02 (90%)	17	10-Sep											🟩			
inal Report Feedback	18	2-Oct												🟩		
Vebsite Assestment	19	Oct												🟩		
inal Report(Soft ound)	20	Oct												🟩		
roject Status Document	21	Oct												🟩		
tudent Log Book	22	Oct												🟩		
inal Presentation & Viva	23	Nov													🟩	
inal Report (Hard Bound)	24	Dec														🟩

Commercialization Plan


00:02
This is Your Smart Fish Tank
Lorem ipsum dolor sit amet consectetur. Leo morbi id pellentesque praesent tempus nunc hac sed. Fames ornare.
Get Start >


Select target audience
Create a Mobile App to provide the service
Run social media campaigns
Provide more facilitations under Subscriptions

Model Demonstration