

AI Driven Smart Tourism Platform for Personalized Safe & Sustainable Travel Planning



Presenter
R25 -006



R25 - 006

08/04/2025

Meet Our Team



SRIKANTHANS.S

IT21821240



SENEVIRATHNE S.D.C.D

IT21831768



Thuwakaran. R

IT21835728



Kumara L. L. M. N

IT21055058



Table Of Content

01. BACKGROUND
02. RESEARCH PROBLEM
03. OBJECTIVES
04. OVERALL SYSTEM
DIAGRAM
05. INDIVIDUAL COMPONENT
DETAILS



BACKGROUND



- Modern travelers seek highly personalized experiences and seamless itinerary planning tools.
- Artificial Intelligence and real-time data are revolutionizing travel by enabling smarter and tailored experiences.
- Existing platforms lack advanced features like real-time safety updates, collaborative travel platform, flexible itinerary management, and effective budget planning.

RESEARCH PROBLEM



-

Existing platforms fail to combine itinerary optimization, real-time safety alerts, and cost management, making it difficult for travelers to access a comprehensive solution in one place.

02

Travelers have limited options to modify their plans dynamically based on changing preferences, unexpected events, or real-time conditions during their journey.

03

Most platforms do not focus on enabling group-based travel planning or connecting travelers with similar interests to collaborate on trips or events.

04

Current tools lack the ability to provide predictive cost estimates and adjust budgets dynamically, preventing travelers from effectively managing their expenses across different activities

OBJECTIVES

OBJECTIVE 01

Provide a Collaborative Travel Companion Platform for group matching and real-time communication

OBJECTIVE 02

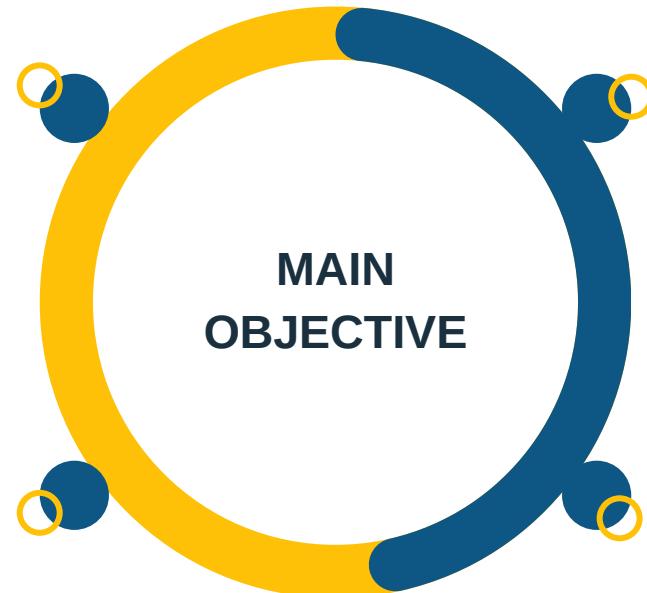
Provide a feature Personalized Recommendation System for AI-driven itineraries.

OBJECTIVE 03

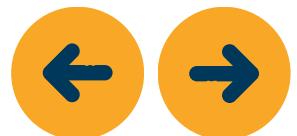
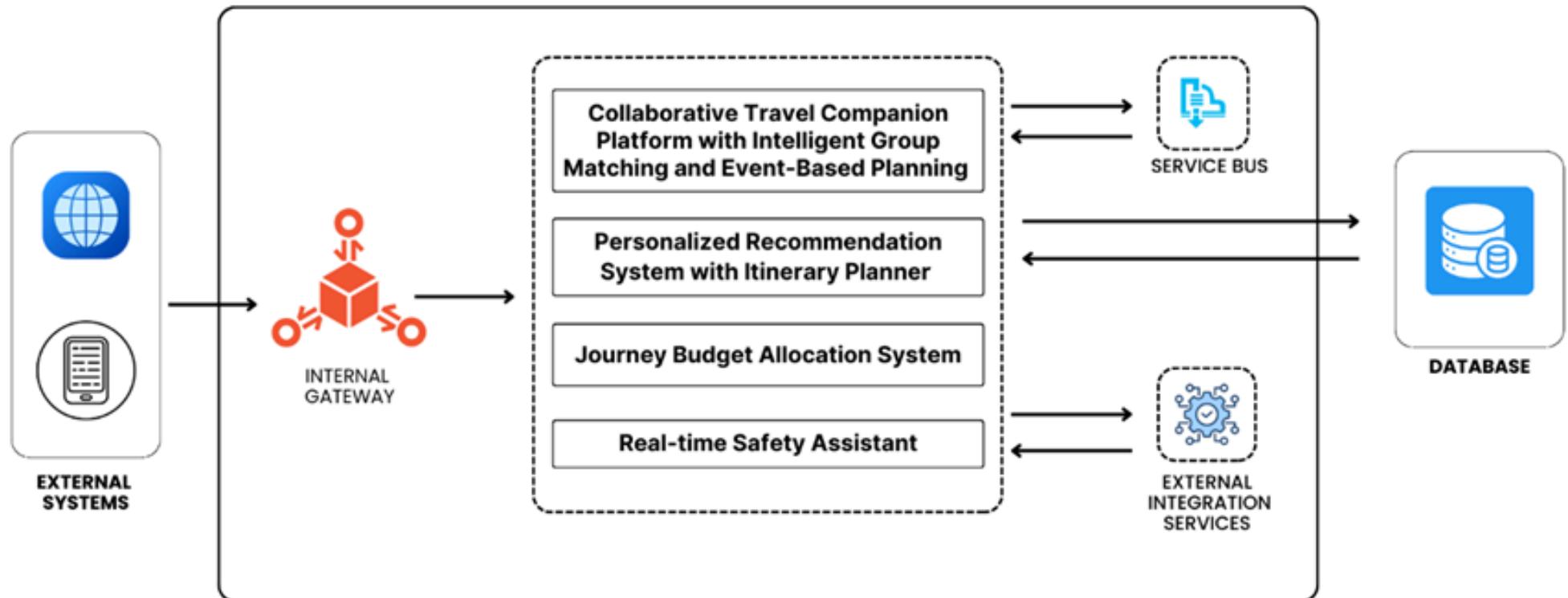
Provide a feature Travel Budget Allocation System for predictive and adjustable cost management.

OBJECTIVE 04

Provide a platform for Real-Time Safety Assistant for instant alerts and data-driven insights.



OVERALL SYSTEM DIAGRAM





GENERAL DETAILS

01. GANTT CHART
02. COMMERCIALIZATION PLAN
03. GITHUB DETAILS
04. PROJECT MANAGEMENT
05. RISK MANAGEMENT



GANTT CHART

Task List	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Research Topic Selection												
Feasibility Study												
Evaluation Feasibility & Background Study												
Topic Evaluation												
Requirement Gathering												
Background Survey												
Literature Review												
Requirement Analysis												
Software Requirement Specification												
Functional & Non-functional Requirements												
Project Charter												
Proposal Presentation												
Project Proposal Report												
Software Design												
Designing Wireframes												
ML Component Development												
Frotend Development												
Progress Presentation 01												
Research Paper												
Unit Testing												
Progress Presentation 02												
Software Integration												
Integration Testing												
Deployment & Maintanance												
Final Presentation & Viva												

COMMERCIALIZATION



The proposed solution will be integrated with the web application and these features are very unique

By comparing the competitors these features are new to the travel platform.

SUBSCRIPTION

ADVERTISING STRATEGY

GITHUB DETAILS

The screenshot shows a GitHub repository page for a project named "Research-Project". The repository is public and has 2 branches and 0 tags. The main file listed is "README.md", which was committed last week by user IT21821240. The repository description is "Final year Research Project". The sidebar on the right provides links to "About", "Releases", and "Packages".

SLIIT-Projects-Y4 / Research-Project

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Edit Pins Watch 0 Fork 0 Star 0

Research-Project Public

main ▾ 2 Branches 0 Tags Go to file Add file Code About

IT21821240 Initial commit 48cf47b · last week 1 Commit

README.md Initial commit last week

README

Research-Project
Final year Research Project

About

Final year Research Project

Readme

Activity

Custom properties

0 stars

0 watching

0 forks

Report repository

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

PROJECT MANAGEMENT

The screenshot shows a Jira software interface for a project titled "Travel Platform". The left sidebar includes links for "Research Project Management", "PLANNING" (Summary, Timeline), "Board" (selected), "Calendar", "List", "Forms", "Goals", "Issues", and "Add view". Below these are sections for "DEVELOPMENT" (Code) and "Project pages" (with a note: "You're in a team-managed project"). The main area displays a Kanban board with four columns: "TO DO 13", "IN PROGRESS 9", "TESTING", and "DONE 6". Each column contains several tasks, each with a title, a "GROUP TASK" or "LEADER'S TASK" indicator, a checkbox for status, and a user icon. The "IN PROGRESS" column has tasks like "Progress Presentation -1" and "Status document 1 - Collaborative Platform". The "DONE" column has tasks like "TAF Document Submission" and "Project Proposal Presentation". A "Quickstart" button is located at the bottom right of the board area.

RISK MITIGATION

Risk	Trigger	Owner	Response	Resource Required
Risk with respect to the Project Team				
Minor delays in deliverables	Small miscommunications or changes in priority	Project Leader	Discuss with team to identify any small delays. Adjust timelines as needed.	Updated Project Schedule Plan/Gantt Chart
Risk with respect to the Panel/Supervisor(s)				
Slight misalignment on expectations	Minor changes in requirements	Project Leader	Confirm and clarify expectations through quick discussions. Update project documents if necessary.	Meeting Notes Updated Project Plan
Panel preferences shift slightly	Panel's evolving preferences or needs	Project Leader	Seek feedback regularly to stay aligned with expectations. Make small adjustments as needed.	Meeting Logs Updated Deliverables
Risk with respect to Project Execution				
Minor technical hiccups	Occasional bugs or troubleshooting	Project Leader	Allocate time for quick debugging and testing. Work with the team to address issues.	Debugging Tools Technical Documentation



IT21821240 - SRIKANTHAN.S

SPECIALIZING - SOFTWARE ENGINEERING

**COLLABORATIVE TRAVEL COMPANION PLATFORM
WITH INTELLIGENT GROUP MATCHING**





INTRODUCTION

01. BACKGROUND
02. RESEARCH QUESTIONS
03. OBJECTIVES



BACKGROUND



- Travel platforms currently lack personalized group matching based on users' specific preferences.
- Limited tools for real-time coordination of activities and cost-sharing among group members.
- Lack of continuous feedback mechanisms to improve recommendations and user satisfaction.

RESEARCH PROBLEM



01

How can intelligent group matching improve the travel experience for users?

02

What features can facilitate effective real-time coordination among travel group members?

03

How can user-generated trip plans help others with similar preferences?

04

How can user groups support travellers in finding suitable travel buddies and planning cost-effective trips?



OBJECTIVES

OBJECTIVE 01

To match users with similar travel interests, styles, destinations, and budget preferences into compatible travel groups.

OBJECTIVE 03

To facilitate seamless communication between group members through a built-in chat system for planning and coordination.

OBJECTIVE 02

To recommend existing travel groups that best suit the user's profile, avoiding the need to create new groups unless necessary.

OBJECTIVE 04

Ensure user-friendly interface and accessible for diverse traveler groups



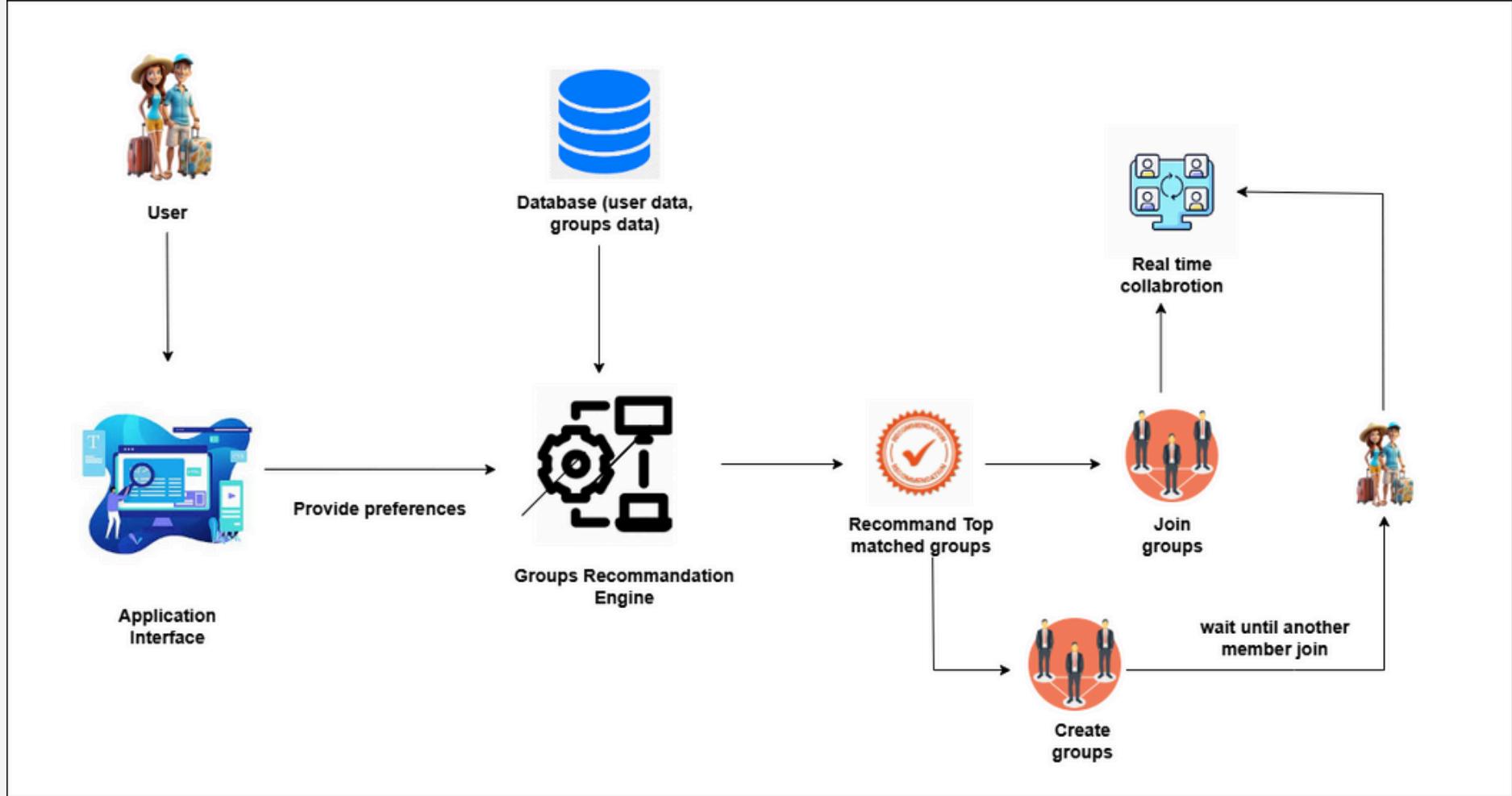


METHODOLOGY

01. SYSTEM DIAGRAM
02. TOOLS AND TECHNOLOGIES
03. REQUIREMENTS
04. GITHUB COMMITS



SYSTEM DIAGRAM



TOOL AND TECHNOLOGIES



Programming Languages
Python



Project Management
Jira



Database
MongoDB



Frameworks & Libraries
lightgbm, pandas, numpy, fastapi, Scikit-learn, sentence-transformers, React



Other Tools
Git , Draw.io
Postman

⋮

REQUIREMENTS

FUNCTIONAL

- Data Collection and Preprocessing
- Model Training and Continuous Learning
- User-Group Matching and Recommendations
- Real-Time Communication
- Trip Planning and Group Collaboration
- Platform Implementation

NON-FUNCTIONAL

- Performance
- Scalability
- Reliability
- Usability
- Security
- Maintainability

GITHUB COMMITS

SRIIT-Projects-Y4 / Research-Project

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Commits

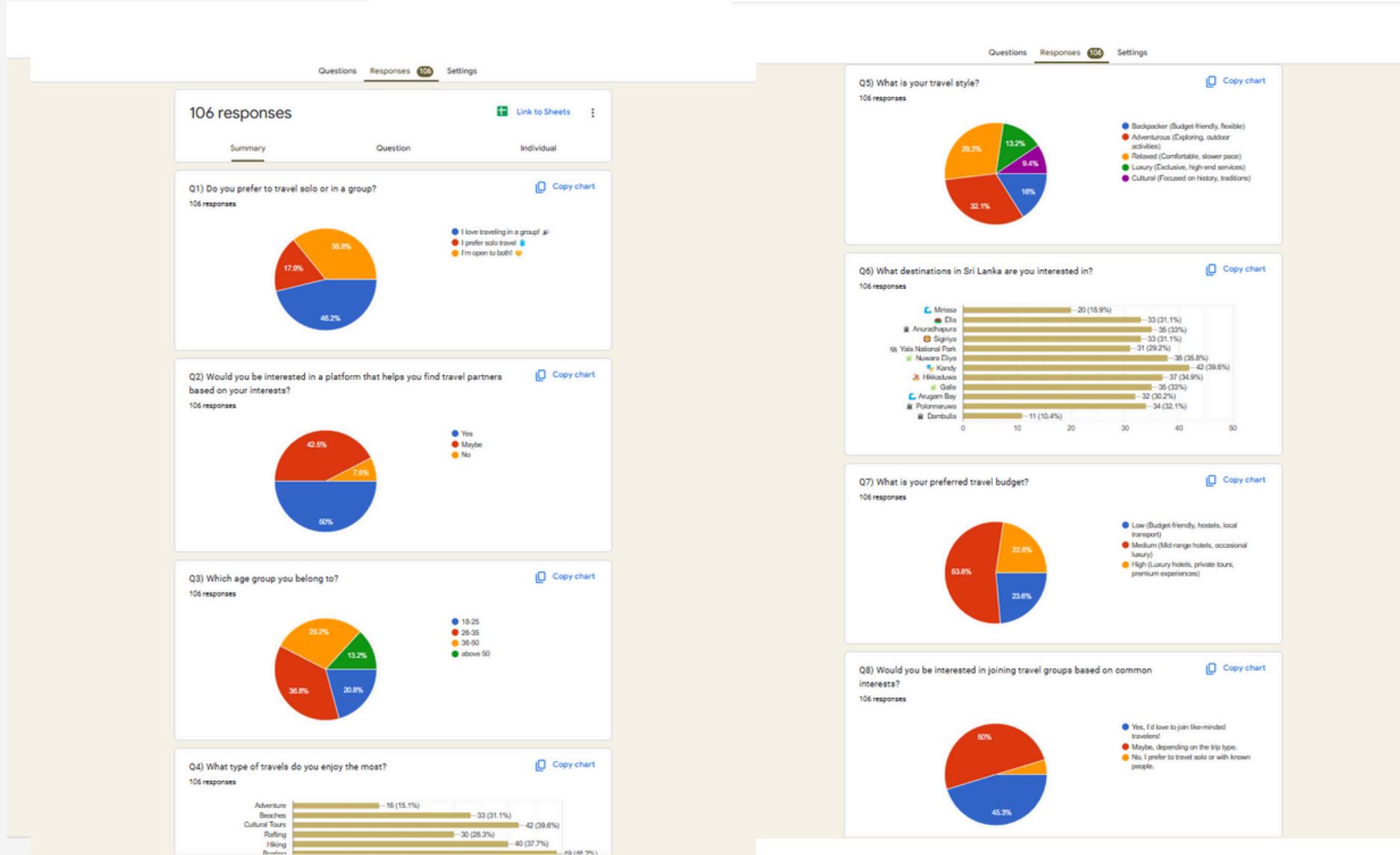
IT21821240/Srika... ▾ IT21821240 All time ▾

- o- Commits on Apr 7, 2025
 - Connected ML model with FastAPI endpoint
IT21821240 committed 2 hours ago 68d07e7 ⌂ ⌂
 - Model Training code added
IT21821240 committed 2 hours ago ba53244 ⌂ ⌂
 - Added data set for model training
IT21821240 committed 2 hours ago b8c7ed2 ⌂ ⌂
- o- Commits on Apr 1, 2025
 - Modal Training
IT21821240 committed last week bdd44f9 ⌂ ⌂
 - Initial commit
IT21821240 authored last week Verified 48cf47b ⌂ ⌂



COMPLETION OF THE PROJECT

DATA COLLECTION - SURVEY



DATA SET CREATION

	A	B	C	D	E	F	G	H	I	J	K
User_ID	Age	User_Interest	Preferred_Destination	Budget	Travel_Style						
2	U0001	20	Cultural Tours	Gregory Lake Community Tsunami Museum	Low	Balanced					
3	U0002	22	Educational Tours, Rafting, Cultural Activities	Arugam Bay	Medium	Luxury					
4	U0003	19	Animal Watching	Pinnawala Elephant Orphanage, Sea Turtle Farm Galle Mahamodara, Ceylon Tea Museum	Low	Backpacker					
5	U0004	25	Yoga, Rafting, Boating	Colombo National Museum, Arugam Bay	Low	Adventurous					
6	U0005	24	Kit Surfing, Swimming	Jaya Sri Maha Bodhi, Polonnaruwa, Rriegala Forest Monastery	Low	Relaxed					
7	U0006	20	Architectural Marvel, Wildlife Spotting	Arugam Bay	Medium	Relaxed					
8	U0007	23	Surfing, Animal Watching, Sunbathing	Lover's Leap Falls	High	Relaxed					
9	U0008	19	Boating, Cultural Tours, Kite Surfing	Lover's Leap Falls, Handunugoda Tea Estate, Jaya Sri Maha Bodhi	Medium	Balanced					
10	U0009	18	Nature Walks, Jet Skiing	Kelaniya Raja Maha Vihara	High	Balanced					
11	U0010	18	Historical Exploration, Wildlife Safaris, Learning History	Udawalawe National Park	High	Balanced					
12	U0011	21	Animal Watching, Fruit Picking	Bennota Beach	High	Luxury					
13	U0012	20	Walking Trails	Temple of the Sacred Tooth Relic, Mihintale, Victoria Park of Nuwara Eliya	Medium	Luxury					
14	U0013	23	Educational Tours, Swimming, Fruit Picking	Dambatenne Tea Factory	Low	Adventurous					
15	U0014	18	Hiking, Yoga, Sunbathing	Horton Plains National Park, National Zoological Gardens of Sri Lanka, Wilpattu National Park	Medium	Adventurous					
16	U0015	18	Kite Surfing	Ravina Ella Falls, Arugam Bay	High	Backpacker					
17	U0016	24	Trekking	National Zoological Gardens of Sri Lanka	Low	Relaxed					
18	U0017	24	Photography, Farm Tours, Rafting	Colombo National Museum, Ceylon Tea Museum, Kalametiya Eco Bird Watching	Medium	Luxury					
19	U0018	23	Trekking, Yoga	Dambulla Cave Temple	Low	Backpacker					
20	U0019	21	Yoga, Hiking	Negombo Beach, Gregory Lake	Low	Adventurous					
21	U0020	18	Photography, Snorkeling	Kalametiya Eco Bird Watching, Hakgala Botanic Gardens	High	Adventurous					
22	U0021	19	Kite Surfing, Worship, Animal Watching	Diyakula Falls	High	Relaxed					
23	U0022	22	Worship, Kite Surfing	Arugam Bay	Low	Balanced					
24	U0023	19	Kite Surfing	Bennota Beach, Tissa Wewa	Low	Relaxed					
25	U0024	18	Boating, Exploring Exhibits	Mount Lavinia Beach, Ramboda Waterfall, Mirissa Beach	Low	Relaxed					
26	U0025	23	Trekking	Passikudah Beach, Temple of the Sacred Tooth Relic, Jethawanaramaya Stupa	High	Adventurous					

	A	B	C	D	E	F	G
Group_ID	Group_Name	Status	Budget	Travel_Style	Destinations_Planned		Group_Interest
1	Radiant Nomads	Active	High	Relaxed	Wilpattu National Park, Sinharaja Forest Reserve, Diyakula Falls		Kite Surfing, Agricultural Workshops
2	3	Inactive	Medium	Relaxed	Ranawana Falls, Samadhi Statue, Brief Garden, Bevis Bawa, Rriegala Forest Monastery		Hiking, Fruit Picking
4	Adventure Nomads	Active	Medium	Adventurous	Brief Garden, Bevis Bawa, Ceylon Tea Museum, Sigiriya The Ancient Rock Fortress, Pigeon Island National Park		Photography, Cultural Exploration
5	Thrillseekers	Active	High	Adventurous	Pedro Tea Factory, Pigeon Island National Park		Swimming, Agricultural Workshops
6	Wanderlust Discoverers	Inactive	Low	Relaxed	Diyakula Falls, Arugam Bay, Udwawale National Park		Animal Watching, Swimming
7	Relaxed Sojourners	Inactive	Medium	Luxury	Diyakula Falls, Arugam Bay, Udwawale National Park		Cultural Tours, Hiking
8	Uively Wanderers	Inactive	Low	Relaxed	Colombo National Museum, Pigeon Island National Park, World Buddhist Museum		Sightseeing, Hiking
9	Elite Pioneers	Active	Medium	Adventurous	Kandy Lake, Polonnaruwa		Tea Plucking, Rafting
10	Energetic Pathfinders	Active	High	Balanced	National Zoological Gardens of Sri Lanka, Marble Beach, Bundala National Park		Animal Watching, Fishing
11	Epic Drifters	Inactive	Low	Backpacker	Bluefield Tea Gardens, Royal Botanical Gardens		Educational Tours, Historical Exploration
12	Cheerful Wanderers	Inactive	High	Adventurous	Wilpattu National Park, Victoria Park of Nuwara Eliya, Mihintale		Fishing, Kite Surfing
13	Epic Pioneers	Active	High	Adventurous	Jaya Sri Maha Bodhi, New Ranwell Spice Garden, Udwawale National Park		Sightseeing, Wildlife Safaris
14	Brave Travellers	Active	Medium	Backpacker	Ariyapala Mask Museum, Udwawale National Park, Udwawale National Park, Pigeon Island National Park		Swimming, Boating
15	Relaxing Pioneers	Active	Medium	Backpacker	Marble Beach, Tissa Wewa, Baker's Falls		Nature Photography, Boating
16	Relaxing Sojourners	Active	Medium	Adventurous	Mount Lavinia Beach		Hiking, Meditation
17	Majestic Adventurers	Inactive	High	Balanced	World Buddhist Museum, Samadhi Statue, Ramboda Waterfall, Community Tsunami Museum, Ruwanwelisaya		Trekking, Animal Watching
18	Cultural Globetrotters	Active	Low	Backpacker	World Buddhist Museum, Samadhi Statue, Ramboda Waterfall, Community Tsunami Museum, Ruwanwelisaya		Photography, Cultural Exploration
19	Majestic Seekers	Inactive	Medium	Backpacker	Wilpattu National Park, Sea Turtle Farm Galle Mahamodara		Animal Watching, Hiking
20	Explorer Adventurers	Inactive	Low	Backpacker	Abhayagiri Dagoba, Baker's Falls, Udwawale National Park		Animal Watching, Architectural Marvel
21	Vibrant Escapists	Active	High	Adventurous	Kandy Lake, St. Clair's Falls, Nilaveli Beach		Nature Walks, Surfing
22	Energetic Escapists	Active	High	Relaxed	Jaffna Fort, Polonnaruwa, Dagoba of Thuparama, Samadhi Statue, Glenloch Tea Factory		Tea Plucking, Animal Watching
23	Thrilling Dreamers	Active	Low	Balanced	Colombo National Museum, Konawewaram Temple, Ambewela Farms		Learning History, Wildlife Spotting
24	Majestic Dreamers	Active	Low	Relaxed	Tevin Baths (Kuttan Poluna), Gangaramaya (Vihara) Buddhist Temple		Worship, Cultural Tours
25	Epic Travellers	Active	Medium	Backpacker	Sigiriya Museum, Galile Fort, Glenloch Tea Factory		Educational Tours, Kit Surfing
26	Wanderlust Pioneers	Inactive	High	Luxury	Lover's Leap Falls, World Buddhist Museum, Rriegala Forest Monastery, Ravana Ella Falls, Arugam Bay		Architectural Marvel, Hiking
	Vibrant Explorers	Active	Low	Adventurous	Diyakula Falls, Ariyapala Mask Museum, Brief Garden - Bevis Bawa		

DATA PRE PROCESSING

```
# Load Dataset
user_df = pd.read_csv("/content/drive/My Drive/Colab Notebooks/User_Preference.csv")
group_df = pd.read_csv("/content/drive/My Drive/Colab Notebooks/Group_Dataset.csv")
interaction_df = pd.read_csv("/content/drive/My Drive/Colab Notebooks/User-Group_Interactions_Dataset.csv")
```

Python

```
# Step 1: Preprocess multi-label fields
def split_multi(x):
    if pd.isna(x):
        return []
    return [i.strip() for i in x.split(',') if isinstance(x, str) else []]

user_df['User_Interest'] = user_df['User_Interest'].apply(split_multi)
user_df['Preferred_Destination'] = user_df['Preferred_Destination'].apply(split_multi)
group_df['Group_Interest'] = group_df['Group_Interest'].apply(split_multi)
group_df['Destinations_Planned'] = group_df['Destinations_Planned'].apply(split_multi)
```

Python

```
#Step 2: Merge DataFrames into a single DataFrame
df = interaction_df.merge(user_df, on='User_ID').merge(group_df, on='Group_ID')
```

Python

```
#Step 3: Ensure no missing values
df['Budget_x'] = df['Budget_x'].fillna('Unknown')
df['Travel_Style_x'] = df['Travel_Style_x'].fillna('Unknown')
df['Age'] = df['Age'].fillna(df['Age'].median())
df['Current_Members'] = df['Current_Members'].fillna(df['Current_Members'].median())
```

Python

MODEL SELECTION

Selected Model LightGBM with LambdaRank

Explored Random Forest Classifier model to predict whether a user would join a group

It had two major limitations:

- It did not rank groups by relevance.
- It could only say “join or not”, not which group is most suitable

Reasons:

- Optimized for Ranking
- Supports Complex Features
- High Performance
- Better Accuracy
- Hybrid Capability



MODEL TRAINING

The screenshot shows a Jupyter Notebook interface with two code cells and a command-line output area.

Code Cell 1:

```
#Step 6: Prepares data for the ranking model
X = np.hstack([
    age_scaled,
    budget_encoded,
    style_encoded,
    df[['Interest_Similarity', 'Destination_Similarity']].values,
    members_scaled
])
y = (df['Joined'] == 'Yes').astype(int).values
user_ids = df['User_ID'].values

#Step 6: Prepares data for evaluation and training
train_users, test_users = train_test_split(np.unique(user_ids), test_size=0.2, random_state=42)
train_mask = df['User_ID'].isin(train_users)

X_train, X_test = X[train_mask], X[~train_mask]
y_train, y_test = y[train_mask], y[~train_mask]
train_group = df[train_mask].groupby('User_ID').size().tolist()
test_group = df[~train_mask].groupby('User_ID').size().tolist()
test_user_ids = user_ids[~train_mask]

#Step 7: Builds a model that ranks groups
train_data = lgb.Dataset(X_train, label=y_train, group=train_group)
valid_data = lgb.Dataset(X_test, label=y_test, group=test_group)

params = {
    'objective': 'lambdaRank',
    'metric': 'ndcg',
    'learning_rate': 0.1,
    'max_depth': 7,
    'verbosity': -1,
    'random_state': 42
}

callbacks = [
    lgb.early_stopping(stopping_rounds=10),
    lgb.log_evaluation(period=10)
]

ranker = lgb.train(
    params=params,
    train_set=train_data,
    num_boost_round=150,
    valid_sets=[valid_data],
    valid_names=['validation'],
    callbacks=callbacks
)
```

Code Cell 2:

```
#Step 8: Assesses the effectiveness of the ranking model
from sklearn.metrics import ndcg_score

def group_by_user(y_true, y_pred, user_ids):
    grouped_true, grouped_pred = [], []
    for uid in np.unique(user_ids):
        mask = user_ids == uid
        grouped_true.append(y_true[mask])
        grouped_pred.append(y_pred[mask])
    return grouped_true, grouped_pred

y_pred = ranker.predict(X_test)
true_groups, pred_groups = group_by_user(y_test, y_pred, test_user_ids)

ndcg_at_3 = np.mean([
    ndcg_score(true, [pred], k=3)
    for true, pred in zip(true_groups, pred_groups)
    if len(true) >= 3
])
print("✅ NDCG@3 Score:", ndcg_at_3)
```

Output Area:

```
... ✅ NDCG@3 Score: 0.6430903102557878

#save the model
model_path = "/content/drive/MyDrive/trained_lgbm_ranker_semantic.txt"
ranker.save_model(model_path)
print("✅ Model saved to:", model_path)

... ✅ Model saved to: /content/drive/MyDrive/trained_lgbm_ranker_semantic.txt
```

Python

Training until validation scores don't improve for 10 rounds
[10] validation's ndcg@1: 0.66 validation's ndcg@2: 0.735712 validation's ndcg@3: 0.853632 validation's ndcg@4: 0.853632 validation's ndcg@5: 0.853632
Early stopping, best iteration is:
[1] validation's ndcg@1: 0.73 validation's ndcg@2: 0.775178 validation's ndcg@3: 0.883427 validation's ndcg@4: 0.883427 validation's ndcg@5: 0.883427

WORKING COMPONENT

```
> import joblib
import numpy as np
import pandas as pd
from sentence_transformers import SentenceTransformer
import lightgbm as lgb

# Load saved components
model = lgb.Booster(model_file="/content/drive/MyDrive/trained_lgbm_ranker_semantic.txt")
budget_encoder = joblib.load("/content/drive/MyDrive/budget_encoder.pkl")
style_encoder = joblib.load("/content/drive/MyDrive/style_encoder.pkl")
scaler = joblib.load("/content/drive/MyDrive/feature_scaler.pkl")
sbert = SentenceTransformer('all-MiniLM-L6-v2')

# Load group dataset
group_df = pd.read_csv("/content/drive/My Drive/Colab Notebooks/Group_Dataset.csv")
group_df['Group_Interest'] = group_df['Group_Interest'].fillna('').apply(lambda x: [i.strip() for i in x.split(',')])
group_df['Destinations_Planned'] = group_df['Destinations_Planned'].fillna('').apply(lambda x: [i.strip() for i in x.split(',')])

def recommend_groups_for_user(age, budget, interests, preferred_destinations=None, travel_style="Unknown"):

    # Clean up destinations list
    preferred_destinations = preferred_destinations if preferred_destinations else []

    # Compute user embeddings
    user_interest_text = ", ".join(interests)
    user_interest_emb = sbert.encode([user_interest_text])[0]

    if preferred_destinations:
        user_dest_text = ", ".join(preferred_destinations)
        user_dest_emb = sbert.encode([user_dest_text])[0]
    else:
        user_dest_emb = np.zeros(384) # Fallback for destination

    # Group embeddings
    group_interest_embs = sbert.encode(group_df['Group_Interest'].apply(lambda x: ', '.join(x)).tolist())
    group_dest_embs = sbert.encode(group_df['Destinations_Planned'].apply(lambda x: ', '.join(x)).tolist())

    # Compute distances between user embeddings and group embeddings
    user_dest_emb_norm = user_dest_emb / np.linalg.norm(user_dest_emb)
    user_interest_emb_norm = user_interest_emb / np.linalg.norm(user_interest_emb)

    group_interest_embs_norm = group_interest_embs / np.linalg.norm(group_interest_embs, axis=1, keepdims=True)
    group_dest_embs_norm = group_dest_embs / np.linalg.norm(group_dest_embs, axis=1, keepdims=True)

    # Compute cosine similarity between user embeddings and group embeddings
    sim_scores = np.dot(user_dest_emb_norm, group_interest_embs_norm.T) + np.dot(user_interest_emb_norm, group_dest_embs_norm.T)

    # Sort groups by similarity
    sorted_indices = np.argsort(sim_scores, axis=0)[::-1]
    sorted_sim_scores = sim_scores[sorted_indices]

    # Filter groups based on budget and travel style
    filtered_indices = []
    for index in sorted_indices:
        if group_df.loc[index, 'Budget'] <= budget and group_df.loc[index, 'Travel_Style'] == travel_style:
            filtered_indices.append(index)

    # Get top N groups
    top_n_indices = filtered_indices[:n]

    # Get group names
    top_n_group_names = group_df.loc[top_n_indices, 'Group_Name'].values
```

```
▶ recommend_groups_for_user(  
    age=23,  
    budget="High",  
    interests=["Hiking", "Nature"],  
    travel_style="Backpacking"  
)  
[47] Python  
... /usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but OneHotEncoder was fitted  
  warnings.warn(  
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but OneHotEncoder was fitted  
  warnings.warn(  
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but MinMaxScaler was fitted  
  warnings.warn(  
...  

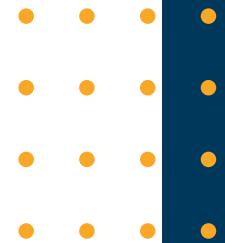

|     | Group_ID | Group_Interest          | Destinations_Planned                               | Current_Members | Score    |
|-----|----------|-------------------------|----------------------------------------------------|-----------------|----------|
| 19  | G0020    | [Nature Walks, Surfing] | [Jaffna Fort, Polonnaruwa, Dagoba of Thuparama...] | 8               | 0.151932 |
| 118 | G0119    | [Sightseeing, Wildlife] | [Mount Lavinia Beach, National Zoological Gard...] | 9               | 0.151932 |
| 74  | G0075    | [Meditation, Hiking]    | [Negombo Beach, Abhayagiri Dagaba, Mount Lavin...] | 6               | 0.151932 |


```



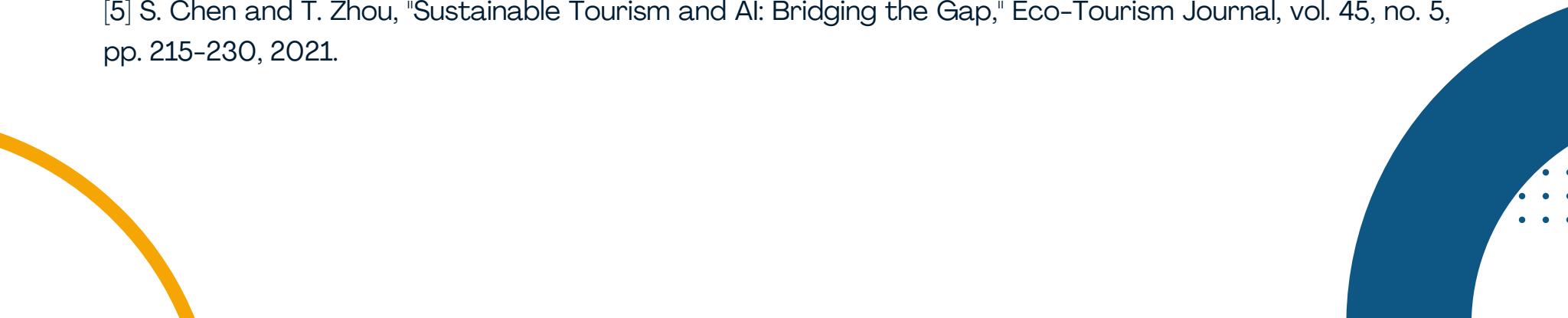
TASK TO BE DONE

- UI/UX Design and front-end development
- Frontend and Backend Integration
- Support for Multimedia in Chat
- Message History & Pagination
- Real-time Sync Across Devices/Tabs
- Security & Moderation





REFERENCES

- [1] K. Liu et al., "AI in Tourism: Enhancing Personalized Travel Experiences," *Journal of Travel Research*, vol. 60, no. 3, pp. 385-397, 2021.
 - [2] A. Smith and J. Lee, "Machine Learning Applications in Travel Itinerary Planning," *Travel Tech Review*, vol. 58, no. 2, pp. 112-129, 2020.
 - [3] H. Zhang and Y. Wang, "Sentiment Analysis for Travel Recommendations: A Review," *International Journal of Tourism Research*, vol. 23, no. 4, pp. 468-482, 2020.
 - [4] M. Kumar and R. Ranjan, "Real-time Communication in Group Travel: Challenges and Opportunities," *Tourism Management Perspectives*, vol. 35, no. 1, pp. 121-134, 2020.
 - [5] S. Chen and T. Zhou, "Sustainable Tourism and AI: Bridging the Gap," *Eco-Tourism Journal*, vol. 45, no. 5, pp. 215-230, 2021.
- 



IT21831768 - SENEVIRATHNE.S.D.C.D

SPECIALIZING - SOFTWARE ENGINEERING





AGENDA

01. BACKGROUND
02. RESEARCH QUESTIONS
03. OBJECTIVES
04. SYSTEM DIAGRAM
05. WORK COMPLETION
06. TASK TO BE DONE





BACKGROUND



RECOMMENDATION SYSTEM

- Users struggle to find locations matching specific preferences
- Traditional systems lack personalized recommendations.
- Need for combining multiple recommendation approaches

ITINERARY PLANNER SYSTEM

- Efficient route planning is critical for Sri Lanka tourism.
- Province-based travel optimization reduces wasted time.

LOCATION SIMILARITY SYSTEM

- Tourism locations share multidimensional similarities.
- Different similarity metrics serve different user needs.
- Comprehensive similarity analysis enhances discovery

RESEARCH Problems?



01

How can multiple recommendation approaches be effectively combined for optimal tourism suggestions?

02

What balancing approach between user preferences and travel logistics creates the most satisfying itineraries?

03

Which combination of similarity dimensions provides the most meaningful location relationships?

04

How does our approach solve limitations found in existing tourism solutions?

OBJECTIVES

Enhance User Interaction and Engagement

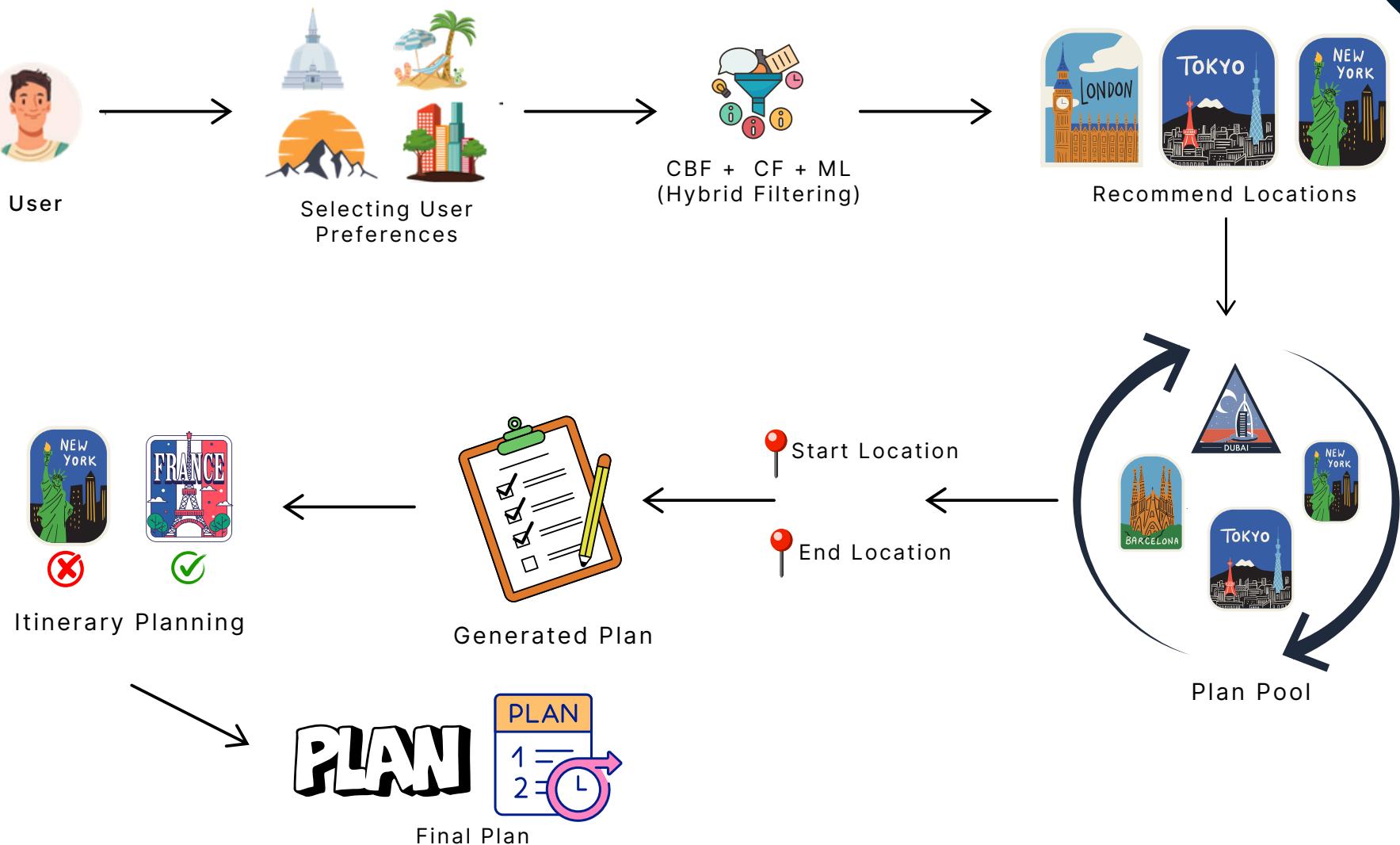
Generate Personalized Recommendations

Suggest Nearby Attractions and Activities

Provide Real-Time Dynamic Itinerary Adjustments

Ensure System Performance and Scalability

SYSTEM DIAGRAM



TOOL AND TECHNOLOGIES



Programming Languages

Python



Project Management

Jira



Database

MongoDB



Frameworks & Libraries

Pandas, Spacy, React,
Scikit-learn



Other Tools

Git , Draw.io Postman,
Docker, Kubernetes

REQUIREMENTS

FUNCTIONAL

- Registration and user profile management.
- Preference collection, including destinations, interests, and budgets.
- Real-time recommendation generation based on user inputs.
- Drag-and-drop functionality for itinerary planning.
- Ability to finalize and save travel plans.

NON-FUNCTIONAL

- Real-time performance and low response times.
- Scalability to handle large datasets and high user traffic.
- Secure storage and transmission of sensitive user data.

GITHUB COMMITS

The screenshot shows the GitHub commits page for the repository 'last-chaanz/tourism-datachecks'. The page displays two sets of commits: one from March 25, 2025, and another from March 18, 2025. Each commit is shown with its title, author, date, commit hash, and copy/paste options.

Commits on Mar 25, 2025:

- final files** by last-chaanz committed 2 weeks ago (commit hash: 4085cfcc)
- location file** by last-chaanz committed 2 weeks ago (commit hash: fc322bd)
- user file** by last-chaanz committed 2 weeks ago (commit hash: 3b79e82)

Commits on Mar 18, 2025:

- generated file set 02** by last-chaanz committed 3 weeks ago (commit hash: 9109f66)
- generated file set 01** by last-chaanz committed 3 weeks ago (commit hash: 50d7f34)
- file updatios and modal creatiion starting with metrices** by last-chaanz committed 3 weeks ago (commit hash: 9e4db10)



COMPLETION OF THE PROJECT

DATA COLLECTION

User_ID	User_Country	User_Age_Group	Gender	Travel_Companion	Location_Type	Preferred_Activities
User 1	Australia	36-50	Male	Family	['Nature & Wildlife Areas', 'Gardens', 'Beaches']	['Wildlife Safaris', 'Fishing', 'Factory Tours', 'Hiking']
User 2	Australia	26-35	Male	Friends	['Nature & Wildlife Areas', 'National Parks', 'Beaches']	['Cultural Tours', 'Boat Tours', 'Sightseeing', 'Snorkeling']
User 3	Australia	18-25	Male	Friends	['National Parks', 'Beaches']	['Boat Tours', 'Jet Skiing', 'Hiking', 'Rafting']
User 4	Portugal	18-25	Male	Solo	['Museums', 'Farms', 'Beaches']	['Yoga', 'Photography', 'Jet Skiing', 'Snorkeling']
User 5	Italy	18-25	Male	Family	['Religious Sites', 'Historic Sites', 'Beaches']	['Rafting', 'Wildlife Spotting', 'Hiking', 'Jet Skiing']
User 6	United Arab Emirates	18-25	Female	Solo	['Historic Sites', 'Beaches']	['Beach Hopping', 'Nature Walks', 'Photography', 'Hiking']
User 7	Indonesia	36-50	Male	Friends	['Historic Sites', 'Beaches']	['Historical Exploration', 'Hiking', 'Walking Trails', 'Bird Watching']
User 8	Canada	18-25	Female	Solo	['Historic Sites', 'Beaches', 'Bodies of Water']	['Nature Walks', 'Photography', 'Beach Hopping', 'Snorkeling']
User 9	New Zealand	36-50	Male	Family	['Beaches', 'Historic Sites', 'Farms']	['Agricultural Workshops', 'Historical Exploration', 'Wildlife Safaris', 'Hiking']
User 10	United Kingdom	51+	Female	Couple	['Museums', 'Historic Sites', 'Beaches']	['Cultural Activities', 'Meditation', 'Relaxation']
User 11	United Kingdom	18-25	Female	Friends	['Waterfalls', 'Beaches', 'Bodies of Water']	['Boat Tours', 'Beach Hopping', 'Snorkeling', 'Surfing']
User 12	Japan	36-50	Male	Family	['Gardens', 'Beaches']	['Fishing', 'Historical Exploration', 'Wildlife Safaris', 'Hiking']
User 13	Australia	26-35	Female	Family	['Historic Sites', 'Beaches']	['Photography', 'Yoga', 'Nature Walks']
User 14	United Kingdom	26-35	Male	Family	['Religious Sites', 'Beaches']	['Hiking', 'Sightseeing', 'Cultural Tours']
User 15	Germany	18-25	Female	Couple	['Beaches', 'Zoological Gardens']	['Relaxation', 'Yoga', 'Beach Hopping', 'Sunbathing']
User 16	Ukraine	18-25	Female	Solo	['Nature & Wildlife Areas', 'Beaches']	['Photography', 'Hiking', 'Nature Walks', 'Yoga']
User 17	United Kingdom	26-35	Female	Couple	['National Parks', 'Beaches']	['Sunbathing', 'Meditation', 'Yoga']
User 18	India	26-35	Male	Family	['Beaches', 'Nature & Wildlife Areas', 'Religious Sites']	['Wildlife Spotting', 'Factory Tours', 'Cultural Tours']
User 19	United Kingdom	51+	Male	Friends	['Waterfalls', 'Historic Sites', 'Beaches', 'Museums', 'Gardens']	['Meditation', 'Wildlife Safaris', 'Worship']
User 20	Viet Nam	51+	Female	Friends	['Religious Sites', 'National Parks', 'Beaches']	['Cultural Exploration', 'Gardens', 'Worship', 'Nature Walks']

Location_ID	Location_Name	Located_City	Located_Province	Location_Type	Latitude	Longitude	Avg_Rating	Review_Count	Activities
LOC_1	Arugam Bay	Arugam Bay	Eastern Province	Beaches	6.8379771	81.8251687	4.06	187	['Swimming', 'Surfing', 'Sunbathing', 'Yoga', 'Boat Tours']
LOC_2	Bentota Beach	Bentota	Southern Province	Beaches	6.4234707	79.9953199	4.59	233	['Swimming', 'Surfing', 'Sunbathing', 'Snorkeling']
LOC_3	Hikkaduwa Beach	Hikkaduwa	Southern Province	Beaches	6.1377266	80.0990596	4.02	213	['Swimming', 'Surfing', 'Sunbathing', 'Snorkeling']
LOC_4	Jungle Beach	Unawatuna	Southern Province	Beaches	6.0186943	80.2394104	3.54	183	['Swimming', 'Sunbathing', 'Snorkeling', 'Hiking']
LOC_5	Kalutara Beach	Kalutara	Western Province	Beaches	6.6098466	79.9474179	3.67	218	['Swimming', 'Surfing', 'Sunbathing', 'Snorkeling']
LOC_6	Marble Beach	Trincomalee	Eastern Province	Beaches	7.7853051	81.4278984	3.97	145	['Swimming', 'Surfing', 'Sunbathing', 'Snorkeling']
LOC_7	Mirissa Beach	Mirissa	Southern Province	Beaches	5.9447028	80.4591608	4.27	217	['Swimming', 'Surfing', 'Sunbathing', 'Snorkeling']
LOC_8	Mount Lavinia Beach	Colombo	Western Province	Beaches	6.842362	79.8624086	4.13	180	['Swimming', 'Sunbathing', 'Kite Surfing']
LOC_9	Negombo Beach	Negombo	Western Province	Beaches	7.2055208	79.8512562	3.12	209	['Swimming', 'Surfing', 'Sunbathing', 'Snorkeling']
LOC_10	Nilaveli Beach	Nilaveli	Eastern Province	Beaches	8.7003072	81.19205	4.33	165	['Swimming', 'Surfing', 'Sunbathing', 'Snorkeling']
LOC_11	Passikudah Beach	Kalkudah	Eastern Province	Beaches	7.9299939	81.5611852	4.56	160	['Swimming', 'Surfing', 'Sunbathing', 'Snorkeling']
LOC_12	Gregory Lake	Nuwara Eliya	Central Province	Bodies of Water	6.95863634	80.77845098	3.96	256	['Boating', 'Walking Trails']
LOC_13	Kandy Lake	Kandy	Central Province	Bodies of Water	7.2912017	80.6420565	3.94	301	['Boating', 'Walking Trails']
LOC_14	Tissa Wewa	Tissamaharama	Southern Province	Bodies of Water	6.2895818	81.2821647	4.43	91	['Historical Exploration', 'Architectural Marvel']
LOC_15	Twin Baths (Kuttam Pokuna)	Anuradhapura	North Central Province	Bodies of Water	8.1995638	80.6326916	3.89	191	['Historical Exploration', 'Architectural Marvel']
LOC_16	Ambewela Farms	Nuwara Eliya	Central Province	Farms	6.8917934	80.8038646	3.9	136	['Factory Tours', 'Agricultural Workshops']
LOC_17	Bluefield Tea Gardens	Nuwara Eliya	Central Province	Farms	7.2564996	80.7214417	4.22	314	['Factory Tours', 'Agricultural Workshops']
LOC_18	Dambatenne Tea Factory	Haputale	Uva Province	Farms	6.7833795	81.0034028	3.36	172	['Factory Tours', 'Agricultural Workshops']
LOC_19	Damro Labookellie Tea Centre	Nuwara Eliya	Central Province	Farms	7.2564996	80.7214417	4.04	90	['Factory Tours', 'Agricultural Workshops']
LOC_20	Glenloch Tea Factory	Katukitula	Central Province	Farms	7.0771474	80.675667	3.82	302	['Factory Tours', 'Agricultural Workshops']



RECOMMENDATION SYSTEM

a). Three-pronged approach

- Content-Based: Matches user activities with location offerings (40% weight)
- Collaborative: Leverages similar travelers' ratings (35% weight)
- Machine Learning: Predicts ratings using XGBoost model (25% weight)

b). Adaptive weighting adjusts based on available data

XGBOOST

```
def _train_enhanced_model(self):
    """Train an enhanced XGBoost model with more features and hyperparameter tuning"""
    # Encode locations for ML
    location_encoder = LabelEncoder()
    location_encoder.fit(self.reviews_df['Location_Name'])
    self.reviews_df['Location_Name_encoded'] = location_encoder.transform(self.reviews_df['Location_Name'])
    self.locations_df['location_Name_encoded'] = location_encoder.transform(self.locations_df['Location_Name'])

    # Prepare feature set - include all encoded features and engineered features
    features = ['User_Age_Group_encoded', 'Gender_encoded', 'Travel_Companion_encoded', 'Location_Name_encoded']

    # Add location type if available
    if 'Location_Type_encoded' in self.reviews_df.columns:
        features.append('Location_Type_encoded')

    # Add user country if available
    if 'User_Country_encoded' in self.reviews_df.columns:
        features.append('User_Country_encoded')

    # Add user rating behavior features if available
    for col in ['Avg_Rating', 'Rating_Count', 'Rating_Variability']:
        if col in self.reviews_df.columns:
            features.append(col)

    # Add activity features
    features.append('Activity_Match_Ratio')
    features.append('Activity_Count')

    # Add activity-specific features
    for col in self.reviews_df.columns:
        if col.startswith('Has_'):
            features.append(col)
```

```
XGBoost model performance: RMSE=0.4891, R2=0.7655
Interpretation:
- RMSE: Average prediction error is 0.49 stars
- R2: Model explains 0.77 of rating variance (excellent)
```

RANDOM FOREST

```
def _train_rf_model(self):
    """Train a Random Forest model with hyperparameter tuning"""
    start_time = time.time() # Start measuring training time

    # Encode locations for ML
    location_encoder = LabelEncoder()
    location_encoder.fit(self.reviews_df['Location_Name'])
    self.reviews_df['Location_Name_encoded'] = location_encoder.transform(self.reviews_df['Location_Name'])
    self.locations_df['location_Name_encoded'] = location_encoder.transform(self.locations_df['Location_Name'])

    # Prepare feature set - include all encoded features and engineered features
    features = ['User_Age_Group_encoded', 'Gender_encoded', 'Travel_Companion_encoded', 'Location_Name_encoded']

    # Add location type if available
    if 'Location_Type_encoded' in self.reviews_df.columns:
        features.append('Location_Type_encoded')

    # Add user country if available
    if 'User_Country_encoded' in self.reviews_df.columns:
        features.append('User_Country_encoded')

    # Add user rating behavior features if available
    for col in ['Avg_Rating', 'Rating_Count', 'Rating_Variability']:
        if col in self.reviews_df.columns:
            features.append(col)

    # Add activity features
    features.append('Activity_Match_Ratio')
    features.append('Activity_Count')

    # Add activity-specific features
    for col in self.reviews_df.columns:
        if col.startswith('Has_'):
            features.append(col)
```

```
Random Forest model performance: RMSE=0.5235, R2=0.6835
Interpretation:
- RMSE: Average prediction error is 0.52 stars
- R2: Model explains 0.68 of rating variance (good)
```

OUTPUT

```
=====
TOP RECOMMENDED LOCATIONS
=====

🌟 TOP PICK: Wilpattu National Park (📍 Nature & Wildlife Areas)
★ Overall Score: 0.86/1.0
📍 Activity Match: 3/4 activities
⭐ Ratings: CBF: 0.94, CF: 4.33, ML: 3.57
📍 Activities: 🦌 Wildlife Spotting, 🐦 Bird Watching, 🏞️ Nature Walks
📄 Based on 18 reviews
```



LOCATION SIMILARITY SYSTEM

a). Hybrid similarity matrix combining four dimensions:

- Numerical: Rating patterns and popularity (40% weight)
- Categorical: Location types and provinces (20% weight)
- Geographic: Physical proximity (20% weight)
- Activity-based: Shared activities (20% weight)

b). Multiple recommendation modes for different user scenarios

```

# 1. Get details for start and end cities
start_info = get_city_details_combined(start_city, locations_dataset)
end_info = get_city_details_combined(end_city, locations_dataset)

start_province = start_info["province"]
end_province = end_info["province"]

print(f"\nStart: {start_info['name']} in {start_province}")
print(f"End: {end_info['name']} in {end_province}")

# Track locations to exclude from city attractions (to avoid duplication)
excluded_locations = [start_info["name"], end_info["name"]]

```

```

# 2. Find the province corridor (path of provinces between start and end)
province_corridor = find_province_path(start_province, end_province)
print(f"\nProvince corridor: {' → '.join(province_corridor)}")

```

```

# 3. Get plan pool locations by province corridor
corridor_plan = get_plan_pool_by_corridor(province_corridor, locations_dataset,
plan_pool_locations)

```

```

# 4. Create the itinerary, province by province
itinerary = []

# Start with start city
itinerary.append({
    "name": start_info["name"],
    "type": "City" if start_info.get("type") is None else start_info["type"],
    "province": start_province,
    "is_city": True,
    "lat": start_info["lat"],
    "lng": start_info["lng"]
})

# Current position is start city
current_lat, current_lng = start_info["lat"], start_info["lng"]

```

OUTPUT

```

TRIP SUMMARY
=====
📍 FROM: Galle Fort (Southern Province)
📍 TO: Anuradhapura (North Central Province)
🔴 TOTAL DISTANCE: 309.9 km
⭐ ATTRACTIONS: 5
ROUTE: 🇸ා Southern Province → 🇲ු Western Province → 🌳 North Western Province → 🏰 North Central Province

```



PLANNER SYSTEM

- Province corridor planning creates optimal province-to-province routes
- Distance optimization using Haversine calculations for efficient sequencing
- Attraction clustering by province minimizes backtracking
- Intelligent day trip handling for circular routes
- Gap filling with nearby attractions to ensure comprehensive itineraries

```

# Convert numeric columns and fill missing values with 0
locations["Avg_Rating"] = pd.to_numeric(locations["Avg_Rating"], errors='coerce').fillna(0)
locations["Review_Count"] = pd.to_numeric(locations["Review_Count"], errors='coerce').fillna(0)
# Fill missing activities with an empty string
locations['Activities'] = locations['Activities'].fillna('')

# Normalize numerical features using MinMaxScaler
scaler = MinMaxScaler()
locations[["Avg_Rating", "Review_Count"]] = scaler.fit_transform(locations[["Avg_Rating",
    "Review_Count"]])

```

```

plan = get_itinerary_plan(
    location_name="Galle Fort",
    included_provinces=["Central Province", "Southern Province", "Western Province"],
    top_n=5
)

```

OUTPUT

```

📍 LOCATION: Galle Fort
=====
📍 LOCATION DETAILS
-----
📍 Type:📍 Historic Sites
📍 Province:📍 Southern Province
📍 Rating:📍 ★★★★ (4.5/5)
📍 Activities:📍 ['Sightseeing',📍 'Photography',📍 'Learning History',📍 'Cultural Tours']

```

```

def recommend_by_activities(user_activities, top_n=5):
    # Clean user activities
    user_set = set([act.strip().lower() for act in user_activities])

    scores = {}
    for idx, row in locations.iterrows():
        loc_set = clean_activities(row["Activities"])
        if not user_set and not loc_set:
            sim = 1.0
        else:
            union = user_set.union(loc_set)
            sim = len(user_set.intersection(loc_set)) / len(union) if union else 0.0
        scores[row["Location_Name"]] = sim

    scores_df = pd.DataFrame(scores.items(), columns=["Location",
        "Activities_Jaccard_Similarity"])
    scores_df = scores_df.sort_values("Activities_Jaccard_Similarity", ascending=False)
    return scores_df.head(top_n)

```

📍 TOP RECOMMENDED PLACES (HYBRID SYSTEM)

- 📍 Sigiriya The Ancient Rock Fortress
 - 📍 Match:📍 0.66
 - 📍 Rating:📍 ★★★★ (4.7/5)
 - 📍 Province:📍 Central Province

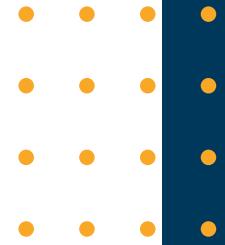
📍 SIMILAR HISTORIC SITES LOCATIONS

- 📍 Sigiriya The Ancient Rock Fortress
 - 📍 Rating:📍 ★★★★ (4.7/5)
 - 📍 Province:📍 Central Province



TASK TO BE DONE

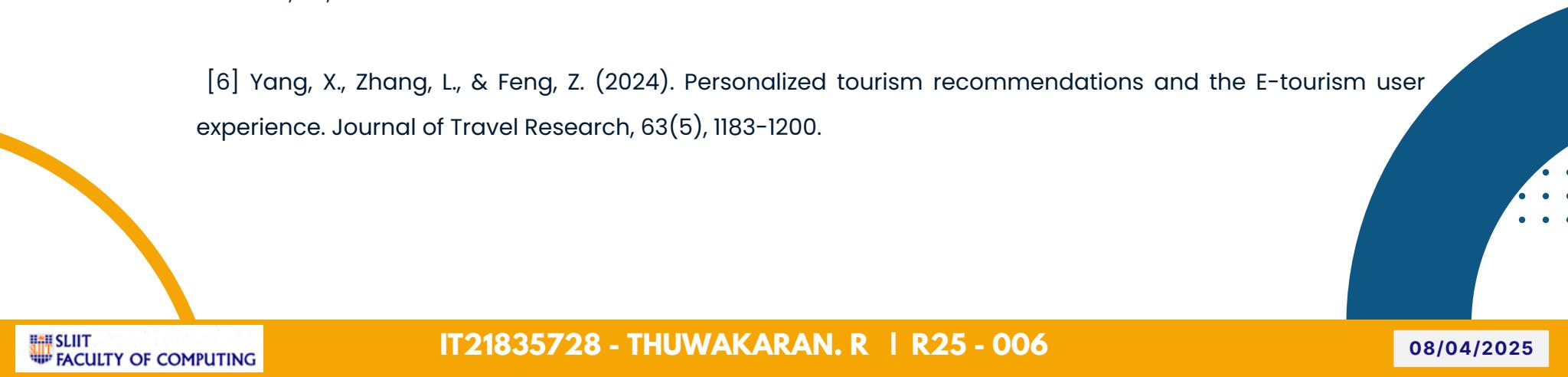
- Backend API Development
- Frontend User Interface
- Cross-System Integration
- Testing & Performance Optimization
- Cloud Deployment





REFERENCES



- [1] Overgoor, G., Chica, M., Rand, W., & Weishampel, A. (2019). Letting the computers take over: Using AI to solve marketing problems. *California Management Review*, 61(4), 156–185.
 - [2] Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1), 44–56.
 - [3] Duarte, L., Torres, J., Ribeiro, V., & Moreira, I. (2020). Artificial Intelligence Systems applied to tourism: A Survey. arXiv preprint arXiv:2010.14654.
 - [4] <https://wttc.org/LinkClick.aspx?fileticket=2VluL2e-nAA%3D&portalid=0&utm>
 - [5] Gretzel, U., Sigala, M., Xiang, Z., & Koo, C. (2015). Smart tourism: foundations and developments. *Electronic markets*, 25, 179–188.
 - [6] Yang, X., Zhang, L., & Feng, Z. (2024). Personalized tourism recommendations and the E-tourism user experience. *Journal of Travel Research*, 63(5), 1183–1200.
- 



IT21835728 - THUWAKARAN. R

SPECIALIZING - SOFTWARE ENGINEERING





INTRODUCTION

01. BACKGROUND
02. RESEARCH QUESTIONS
03. MAIN AND SUB OBJECTIVES



BACKGROUND



Static Budgeting Tools

- Traditional budgeting tools rely on outdated, static data and manual processes, which don't adjust to fluctuating travel costs or changing user preferences.

Lack of Personalization

- Current platforms offer generalized recommendations that don't meet the unique needs of travelers, such as varying group sizes, budgets, and dynamic itineraries.

Limited Real-Time Adaptation

- Without real-time data integration or dynamic adaptability, platforms fail to adjust to unforeseen changes like weather, price fluctuations, or schedule conflicts, leaving travelers with inefficient plans.

RESEARCH Questions?



01

What are the limitations of current tools for travel budget allocation?

02

How can predictive algorithms and real-time data improve the accuracy and reliability of travel budgeting?

03

How can user-centric design enhance accessibility and usability for a broader audience?

04

How does the proposed system address these gaps and stand out from existing solutions?

OBJECTIVES

- **Develop a Dynamic Budget System**

Create an intelligent Travel Budget Allocation System that adapts to real-time data and user preferences, offering personalized and accurate budget recommendations.

- **Implement Predictive Algorithms**

Integrate predictive models to estimate travel costs (accommodations, transport, meals) using historical data and real-time inputs for accurate forecasts.

- **Design a User-Centric Interface**

Build an intuitive interface allowing travelers to customize budget preferences and receive tailored recommendations based on their travel style and goals.

- **Integrate APIs & Validate System**

Integrate external APIs for real-time cost data and conduct rigorous testing to ensure system accuracy, reliability, and adaptability to changing market conditions and user needs.

SMART BUDGET ALLOCATION & ADAPTIVE PLANNING

"Our budget system tailors recommendations based on traveler type and package, offering options like affordable hostels for solo travelers and resorts for families, ensuring a personalized, budget-friendly experience."

Customizable Packages: Basic, Moderate, and Premium options tailored to different traveler types, offering varying levels of amenities and services.

Diverse Travel Companions: Plans customized for Solo, Couple, Family, and Friends, based on group size and preferences.

Dynamic Cost Predictions: Predicts breakdowns for Transport, Stay, Activities, Food, and Accommodation, adjusting based on real-time pricing and user preferences.

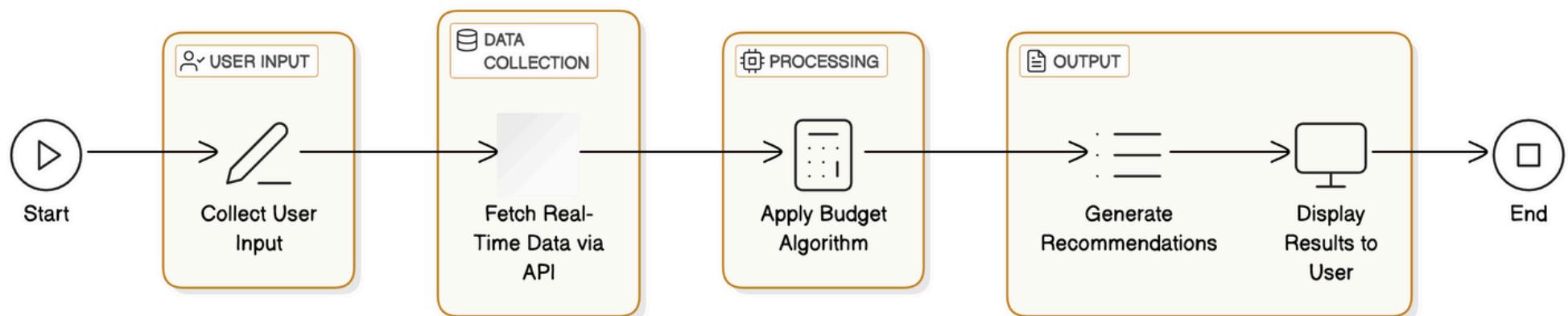


METHODOLOGY

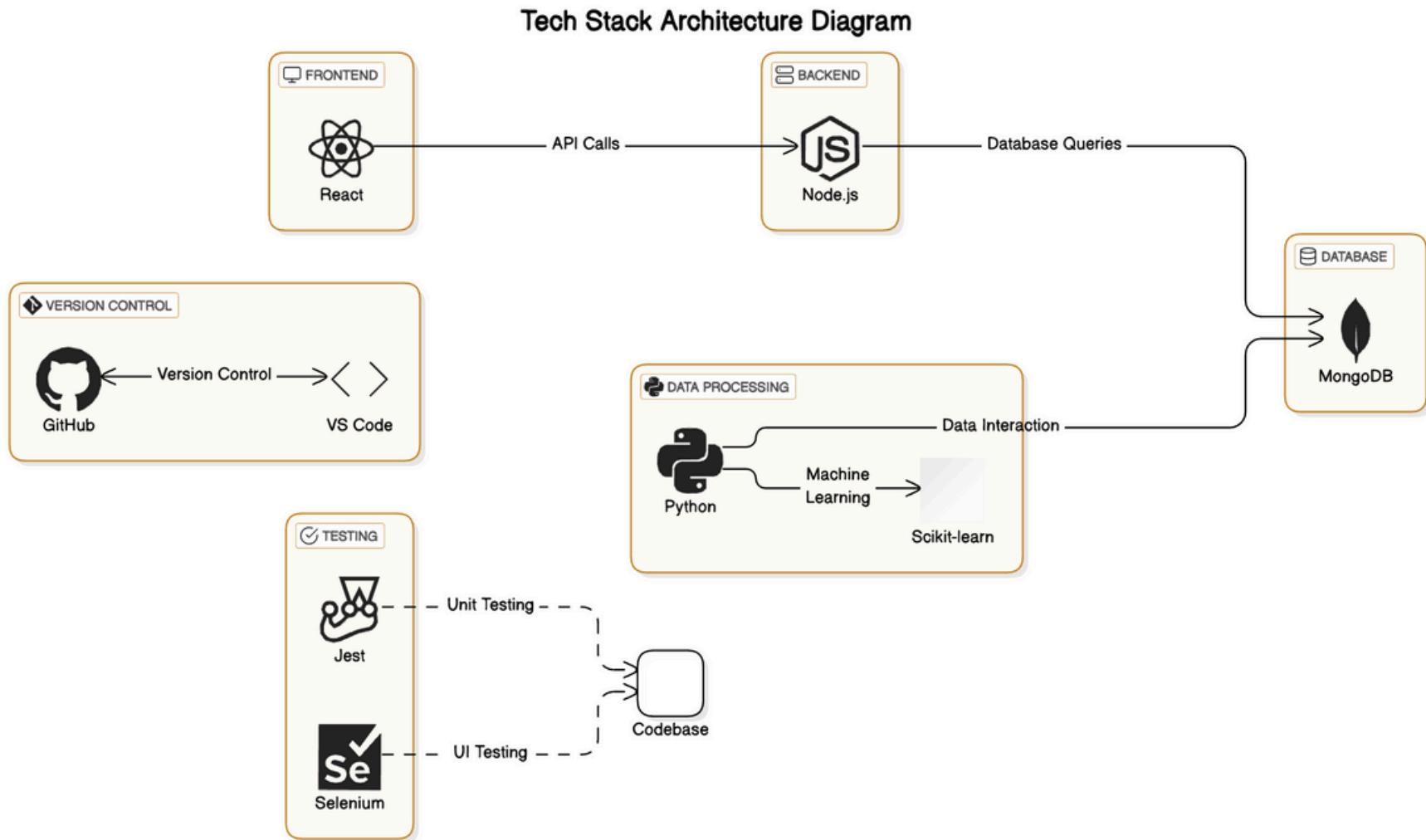
01. SYSTEM DIAGRAM
02. TOOLS AND TECHNOLOGIES
03. REQUIREMENTS
04. GITHUB COMMITS



BUDGET PREDICTION SYSTEM DIAGRAM



TOOLS & TECHNOLOGIES



REQUIREMENTS

FUNCTIONAL

- Real-Time Data Integration
- Accurate Predictions
- Customizable Suggestions
- Notification System
- Collaboration

NON-FUNCTIONAL

- Reliability of Predictions
- Fast Response Times
- User-Friendly Interface
- Maintainability
- Localization

GITHUB COMMITS

Commits on Mar 18, 2025			
generated file set 02		9109f66	 
 thuwakaranR committed 3 weeks ago			
generated file set 01		50d7f34	 
 thuwakaranR committed 3 weeks ago			
file updatios and modal creatiion starting with metrices		9e4db10	 
 thuwakaranR committed 3 weeks ago			
working with user demographic data set		32c2f49	 
 thuwakaranR committed 3 weeks ago			
similarity metrices implementations		48c3238	 
 thuwakaranR committed 3 weeks ago			
Commits on Mar 16, 2025			
generated csv file v2		229f6f2	 
 thuwakaranR committed 3 weeks ago			
generated csv file v1		f8216f3	 
 thuwakaranR committed 3 weeks ago			
hybrid and other implementation 02		f28933f	 
 thuwakaranR committed 3 weeks ago			
hybrid and other implementation 01		61f7547	 
 thuwakaranR committed 3 weeks ago			



COMPLETION OF THE PROJECT

DATA COLLECTION

2617	LOC_76	PACK_76Z3-2	Premium	Sunrise Visit, Attend Thaipoosam Festival with Private Viewing, Explore Temple with Private Guide	1	Luxury Resort	All-Inclusive, Private Transport	20,500	Exclusive Sunrise Visit, Festival Experience	3.9	70	Couple
2618	LOC_76	PACK_76Z2-3	Moderate	Visit Temple, Attend Special Worship, Visit the Garden and Pond	1	Boutique Hotel	Full Board, Shared Transport	15,300	Worship, Spiritual Serenity	3.6	55	Couple
2619	LOC_76	PACK_76Z3-3	Premium	Private Cultural Experience: Attend Special Worship, Explore Temple Architecture, Visit the Pond	1	Luxury Resort	All-Inclusive, Private Transport	21,500	Tailored Experience, Private Guide	2.7	80	Couple
2620	LOC_76	PACK_76Z2-2	Moderate	Attend the Annual Festival, Explore Temple Parades with Music & Dance	1	Boutique Hotel	Full Board, Shared Transport	15,000	Festival Atmosphere, Parades	3.9	50	Couple
2621	LOC_76	PACK_76Z3-4	Premium	Full Day Experience: Nallur Kandaswamy Temple, Lord Muruga's Worship, Special Blessings	1	Luxury Boutique Stay	All-Inclusive, VIP Transport	22,500	Exclusive Experience, VIP Worship	4.9	90	Couple
2622	LOC_1	PACK_1Z1-1	Basic	Surfing (Beginner), Beach Walks	1	Budget Guesthouse	Breakfast Only, Local Transport	34,500	Public Beach Access, Local Culture	2.8	95	Family
2623	LOC_1	PACK_1Z1-2	Basic	Lagoon Safari, Elephant Rock Visit	1	Homestay	Breakfast & Dinner, Shared Transport	38,500	Wildlife Spotting, Scenic Views	3.2	80	Family
2624	LOC_1	PACK_1Z2-1	Moderate	Surfing (Intermediate), Jeep Safari in Kumana	2	Beachfront Hotel	Half Board, Private Transport	51,000	Beachfront View, Guided Tours	4.1	120	Family
2625	LOC_1	PACK_1Z2-2	Moderate	Snorkeling, Yoga, Local Cuisine Tour	2	Boutique Villa	Full Board, Shared Transport	54,000	Traditional Food, Cultural Immersion	3.7	110	Family
2626	LOC_1	PACK_1Z3-1	Premium	Private Surfing Lessons, Private Boat Tours	2	Luxury Resort	Full Board, Private Chauffeur	81,000	Private Beach Access, Luxury Spa	4.6	150	Family

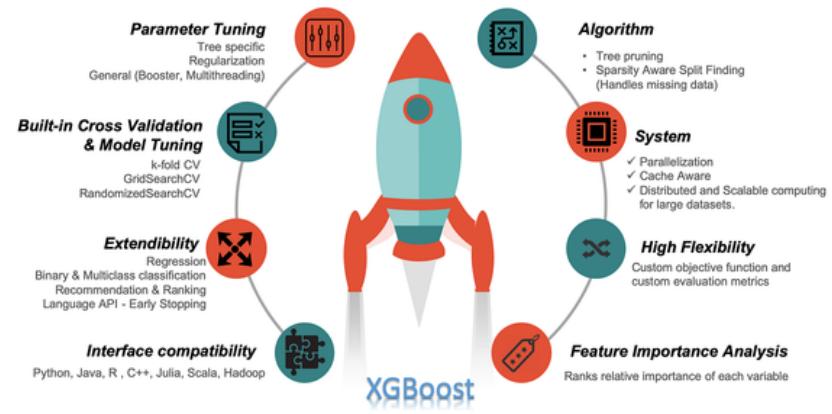


	Location_ID	Package_ID	Package_Type	Activities	Duration (Days)	Accommodation	Food & Transport	Budget (LKR)	Location_Features	Avg_Rating	Review_Count	Travel_Companion
1	LOC_1	PACK_1Z1-1	Basic	Surfing (Beginner), Beach Walks	1	Budget Guesthouse	Breakfast Only, Local Transport	23,500	Public Beach Access, Local Culture	2.8	95	Solo
2	LOC_1	PACK_1Z1-2	Basic	Lagoon Safari, Elephant Rock Visit	1	Homestay	Breakfast & Dinner, Shared Transport	27,500	Wildlife Spotting, Scenic Views	3.2	80	Solo
3	LOC_1	PACK_1Z2-1	Moderate	Surfing (Intermediate), Jeep Safari in Kumana	2	Beachfront Hotel	Half Board, Private Transport	40,000	Beachfront View, Guided Tours	4.1	120	Solo
4	LOC_1	PACK_1Z2-2	Moderate	Snorkeling, Yoga, Local Cuisine Tour	2	Boutique Villa	Full Board, Shared Transport	43,000	Traditional Food, Cultural Immersion	3.7	110	Solo
5	LOC_1	PACK_1Z3-1	Premium	Private Surfing Lessons, Private Boat Tours	2	Luxury Resort	Full Board, Private Chauffeur	70,000	Private Beach Access, Luxury Spa	4.6	150	Solo
6	LOC_1	PACK_1Z3-2	Premium	Helicopter Tour, Exclusive Beachfront Dining	3	5-Star Resort	All-Inclusive, Luxury Transport	90,000	VIP Beachfront, Exclusive Experience	4.3	130	Solo
7	LOC_1	PACK_1Z3-3	Premium	Yala Safari Camping, Traditional 'Oruwa' Rafting	2	Luxury Safari Camp	Full Board, Private Guide	100,000	Unique Wildlife Encounters	4.8	140	Solo
8	LOC_1	PACK_1Z1-3	Basic	Beach Volleyball, Turtle Watching	1	Hostel Dorm	Breakfast Only, Shared Transport	22,500	Public Beach Access, Marine Life	2.5	70	Solo
9	LOC_1	PACK_1Z1-4	Basic	Mangrove Kayaking, Fishing Experience	1	Eco Lodge	Breakfast & Dinner, Local Transport	25,000	Scenic Views, Local Culture	3.1	65	Solo
10	LOC_1	PACK_1Z2-3	Moderate	Stand-up Paddleboarding, Cultural Dance Show	2	Eco Boutique Hotel	Full Board, Private Transport	41,500	Traditional Food, Cultural Immersion	4.4	100	Solo
11	LOC_1	PACK_1Z2-4	Moderate	Hiking, Tea Plantation Visit	2	Hillside Retreat	Half Board, Shared Transport	39,500	Scenic Views, Nature Trails	3.8	105	Solo
12	LOC_1	PACK_1Z2-5	Moderate	Sunset Cruise, Local Handicraft Workshop	2	Riverside Bungalow	Full Board, Local Transport	42,000	Traditional Crafts, River Views	4	90	Solo

MODEL SELECTION

-XGBoost Regressor with RandomizedSearchCV-

- Explored XGBoost Regressor to predict the budget for a given travel input setup using customized features.
- Used RandomizedSearchCV for tuning hyperparameters to improve prediction accuracy and model generalization.



- Fine-tuned Performance
- Better Accuracy
- Flexible and Powerful
- Efficient and Fast

BUDGET PREDICTION – CODE EXECUTION FLOW

```
[ ] df = pd.concat([df, df_activities], axis=1)
df.drop(columns=['Activities'], inplace=True) # Drop original activities column

# Add Travel_Companion column and encode it
travel_companion_map = {'Solo': 0, 'Couple': 1, 'Family': 2, 'Friends': 3}
df['Travel_Companion'] = df['Travel_Companion'].map(travel_companion_map)

# Define Rating Ranges (0-1, 1-2, 2-3, 3-4, 4-5)
def categorize_rating(rating):
    return f'{int(rating)}-{int(rating) + 1}'

df['Rating_Range'] = df['Avg_Rating'].apply(categorize_rating)

[ ] # Select features and target variable
X = df.drop(columns=['Package_ID', 'Accommodation', 'Food & Transport', 'Budget (LKR)', 'Rating_Range'])
y = df['Budget (LKR)']

# Split dataset into training (80%) and testing (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ] # Initialize XGBoost regressor model
xgb_model = XGBRegressor(random_state=42)

# Train the model
xgb_model.fit(X_train, y_train)
```

```
if selected_packages:
    final_results = pd.DataFrame(selected_packages)
    final_results = final_results[['Location', 'Package_ID', 'Package_Type', 'Days', 'Food & Transport', 'Avg_Rating', 'Activities']]
]

# Format the predicted budget to two decimal places
final_results['Predicted_Budget'] = final_results['P']

total_duration = final_results['Days'].sum()
total_budget = round(total_budget, 2)
possible_combinations.append((final_results, total_d

return possible_combinations
```

```
XGBOOST
dict(X_test)

# evaluate
r2_tuned = r2_score(y_test, y_pred_tuned)
rmse = np.sqrt(mean_squared_error(y_test, y_pred_tuned)) # fixed RMSE calculation

# Output
print("\nBest Parameters Found:")
print(random_search.best_params_)
print(f"R² Score: {r2_tuned:.4f}")
print(f"RMSE: {rmse:.4f}")

# Plotting predicted vs actual
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred_tuned, alpha=0.6, color='dodgerblue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='darkred', linestyle='--')
plt.title(f"Predicted vs Actual | R²: {r2_tuned:.3f}")
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
[ ] # Make predictions
y_pred = xgb_model.predict(X_test)

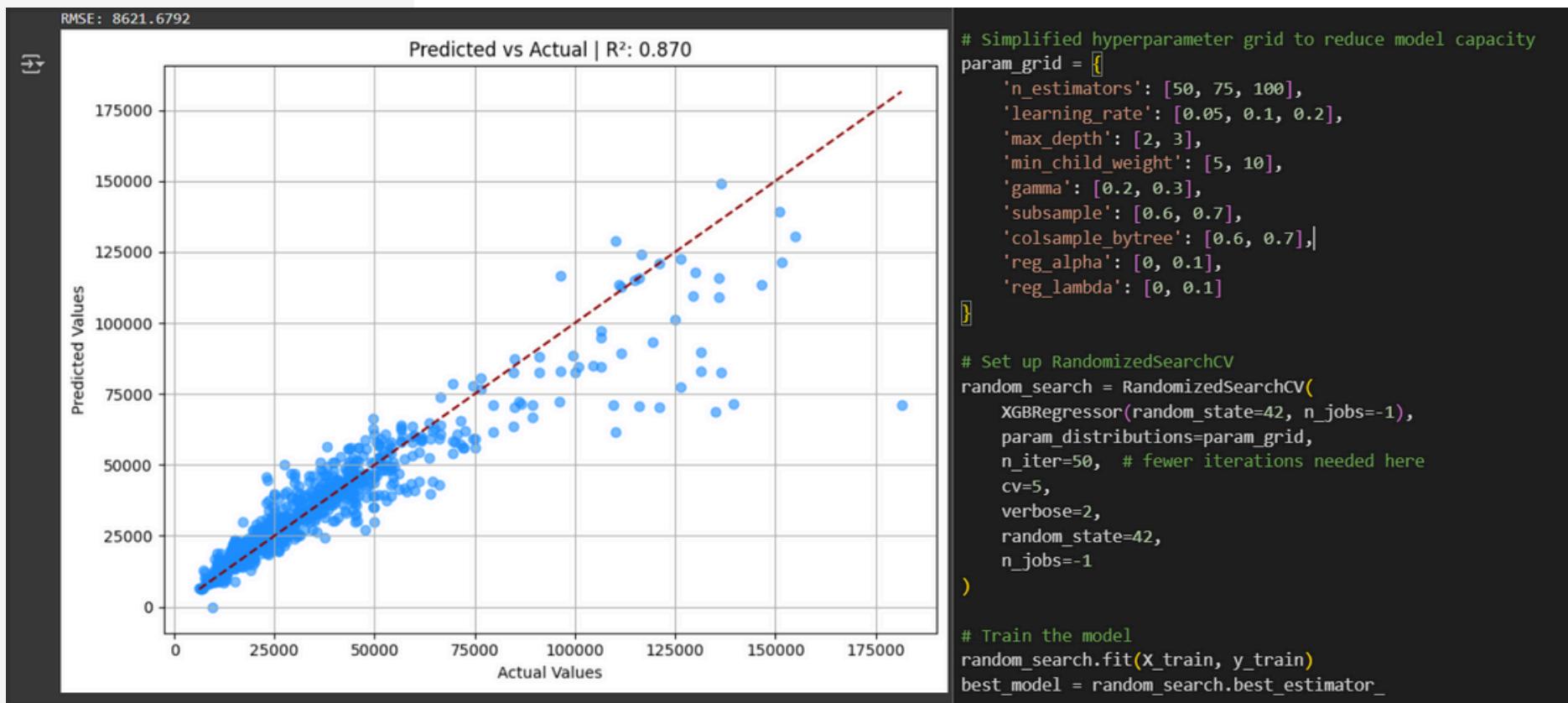
# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred) ** 0.5
r2 = r2_score(y_test, y_pred)

# Display evaluation results
print(f"✅ XGBoost Model Evaluation:\nMAE: {mae}\nRMSE: {rmse}\nR²: {r2}")

→ ✅ XGBoost Model Evaluation:
MAE: 2769.10 LKR
RMSE: 5242.96 LKR
R²: 0.72
```

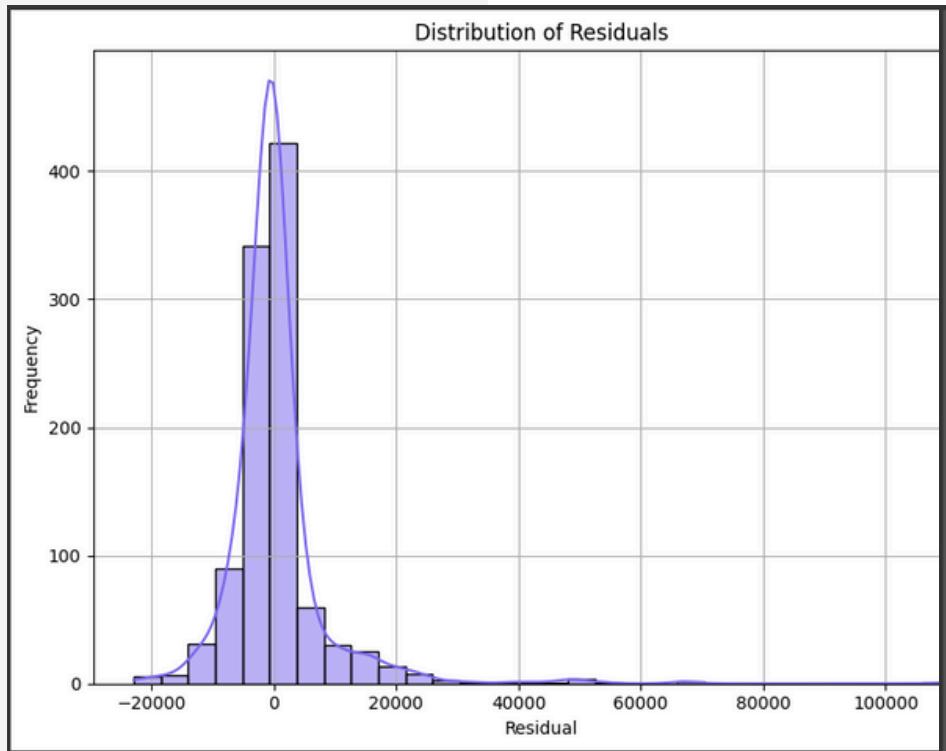
ACCURACY & ANALYSIS

- Plotting



ACCURACY & ANALYSIS

- Plot Histogram



```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate residuals
residuals = y_test - y_pred_tuned

# Plot histogram of residuals
plt.figure(figsize=(8, 6))
sns.histplot(residuals, kde=True, bins=30,
             plt.title("Distribution of Residuals")
             plt.xlabel("Residual")
             plt.ylabel("Frequency")
             plt.grid(True)
             plt.tight_layout()
             plt.show()
```

BUDGET PREDICTION & ITERATIVE RESULTS

"HERE'S HOW OUR BUDGET SYSTEM FITS IN. IT PULLS LIVE COST DATA, APPLIES PREDICTION LOGIC, AND RETURNS AN OPTIMIZED, ADAPTIVE BUDGET PLAN BASED ON USER INPUTS AND PREFERENCES."

```
locations = ["LOC_11", "LOC_42", "LOC_6"]
package = "Moderate"
total_days = 8
rating_range = "4-5"
travel_companion = "solo"

predicted_packages = predict_budget_multiple_options(locations, package, total_days, rating_range, travel_companion, df, best_model, le_location, le_package)

# Display the results
for i, (packages, total_duration, total_budget) in enumerate(predicted_packages, 1):
    print("+" * 72)
    print(f"{'*' * 72} Trip Plan {str(i)} {'*' * 72}") # Center-aligned title
    print("+" * 72)

    # Display the package details
    print(packages)
    |
    # Display travel companion
    print("-" * 72)
    print(f"💡 Travel Companion: {travel_companion}")

    print("-" * 72)
    print(f"⌚ Total Duration: {total_duration} days")
    print(f"💰 Total Budget: {total_budget:.2f} LKR")
    print("-" * 72, "\n")
    print("\n")
```

Trip Plan 1					
Location	Package_ID	Package_Type	Days	Accommodation	
697	LOC_42	PACK_42Z2-4	Moderate	2.0	Seaside Guesthouse
696	LOC_42	PACK_42Z2-1	Moderate	1.0	Boutique Villa
101	LOC_6	PACK_6Z2-3	Moderate	2.0	4-Star Resort

Food & Transport Avg_Rating \		
697	Full Board, Shared Transport	4.0
696	Half Board, Shared Transport	4.1
101	Full Board, Private Transport	4.3

Activities Predicted_Budget		
697	[Morning Safari, Nature Walk, Spot Flamingos]	25405.94
696	[Birdwatching, Explore Flora & Fauna, Lagoon T...	20413.94
101	[Boat Ride to Trincomalee, Whale Watching (Sea...	51253.43

💡 Travel Companion: Solo

⌚ Total Duration: 5.0 days

💰 Total Budget: 97073.31 LKR

Trip Plan 2					
Location	Package_ID	Package_Type	Days	Accommodation	
200	LOC_11	PACK_11Z2-8	Moderate	3.0	Beachfront Resort
697	LOC_42	PACK_42Z2-4	Moderate	2.0	Seaside Guesthouse
103	LOC_6	PACK_6Z2-5	Moderate	2.0	Beachfront Hotel

Food & Transport Avg_Rating \		
200	Full Board, Private Transport	4.0
697	Full Board, Shared Transport	4.0
103	Full Board, Private Transport	4.1

Activities Predicted_Budget		
200	[Relax on Kalkudah Beach, Watersports at Pasik...	26724.68
697	[Morning Safari, Nature Walk, Spot Flamingos]	25405.94
103	[Diving, Underwater Photography]	43833.84

💡 Travel Companion: Solo

⌚ Total Duration: 7.0 days

💰 Total Budget: 95964.38 LKR

Trip Plan 3					
Location	Package_ID	Package_Type	Days	Accommodation	
200	LOC_11	PACK_11Z2-8	Moderate	3.0	Beachfront Resort
696	LOC_42	PACK_42Z2-1	Moderate	1.0	Boutique Villa
103	LOC_6	PACK_6Z2-5	Moderate	2.0	Beachfront Hotel

Food & Transport Avg_Rating \		
200	Full Board, Private Transport	4.0
696	Half Board, Shared Transport	4.1
103	Full Board, Private Transport	4.1

Activities Predicted_Budget		
200	[Relax on Kalkudah Beach, Watersports at Pasik...	23317.40
696	[Birdwatching, Explore Flora & Fauna, Lagoon T...	16623.75
103	[Diving, Underwater Photography]	28720.92

💡 Travel Companion: Solo

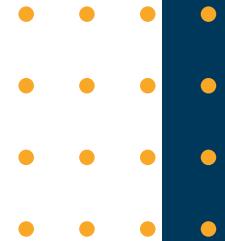
⌚ Total Duration: 6.0 days

💰 Total Budget: 68662.07 LKR



TASK TO BE DONE

- Backend API Development
- Frontend User Interface
- Cross-System Integration
- Testing & Performance Optimization
- Cloud Deployment



REFERENCES

- [1] J. Anderson, M. Smith, and L. Roberts, "Limitations in static budgeting models: Examining traditional travel budgeting tools," *Journal of Travel Management*, vol. 34, no. 2, pp. 45–62, 2020.
- [2] H. Tan, C. Lee, and S. Tan, "Role of real-time data in budget predictions: Integrating APIs for travel budgeting systems," *International Journal of Data Analytics*, vol. 19, no. 1, pp. 30–42, 2021.
- [3] X. Zhang, L. Wang, and Q. Chen, "Importance of user-centric designs in travel budgeting tools," *Journal of User Experience Design*, vol. 12, no. 4, pp. 58–75, 2022.
- [4] J. Smith and K. Thomas, "Machine learning for travel budgeting: Enhancing cost predictions with predictive algorithms," *Journal of Artificial Intelligence in Tourism*, vol. 16, no. 3, pp. 215–228, 2021.
- [5] M. Green and D. Turner, "Real-time data integration: A new frontier in travel budgeting systems," *Journal of Travel Technology*, vol. 21, no. 1, pp. 50–65, 2020.



THANK YOU

