

INES-RUHENGERI
COMPUTER SCIENCE/SWE
ARTIFICIAL INTELLIGENT
CLASS A/ GROUP #10 Assignment

2/24/2025

Group member Names	RegNo
Blessing Allison	23/20940
Juma Filberto George Andrea	23/21433
Elvis T. Harmon	23/23787
Allen Albert A	23/19354
UWERA Denye	23/22375

Task 1

Tourism Site Recommendation platform in Rwanda

1. Use Case Definition

Problem Definition

Many tourists struggle to choose travel destinations that align with their interests, budget, and preferences. This challenge is particularly established in regions like Rwanda and Africa, where the diversity of tourist attractions is huge but often underpromoted. A **Tourism Site Recommendation System** is an expert system designed to help travelers discover suitable destinations based on their input. For example, a tourist interested in wildlife might not be aware of Akagera National Park in Rwanda, while someone seeking cultural experiences might overlook the King's Palace Museum. This system aims to bridge this gap by providing personalized recommendations tailored to individual preferences. By leveraging user inputs such as travel interests, budget, and duration, the system ensures that tourists can make informed decisions about their travel plans. Additionally, it promotes lesser-known destinations, thereby supporting local economies and encouraging sustainable tourism practices. Ultimately, this system enhances the overall tourist experience while contributing to the growth of the tourism industry in Rwanda and across Africa.

Relevance in the Rwandan/African Context

Tourism is a growing industry in Rwanda and across Africa, contributing significantly to economic development. Rwanda, for example, is known for its gorilla trekking, national parks, and cultural heritage sites. However, many tourists are unaware of the diverse options available, such as the serene beaches of Lake Kivu, the historical significance of the King's Palace Museum, or the breathtaking views of the Virunga Mountains. This lack of awareness often leads to underutilization of these attractions, limiting their economic potential. A **Tourism Site Recommendation System** addresses this issue by providing tailored suggestions based on user preferences, ensuring that tourists can explore the full range of experiences Rwanda and Africa have to offer. By promoting lesser-known destinations, the system not only enhances the tourist experience but also supports local communities and businesses. Furthermore, it encourages sustainable tourism by guiding visitors to eco-friendly destinations and reducing overcrowding at

popular sites. In a continent rich with natural beauty, cultural heritage, and adventure opportunities, this system plays a crucial role in unlocking the full potential of African tourism.

This system will:

- 1. **Promote lesser-known destinations** like Nyungwe Forest (Rwanda) for its canopy walks and chimpanzee tracking.
- 2. **Boost local economies** by encouraging visits to community-based tourism sites such as Iby'Iwacu Cultural Village (Rwanda).
- 3. **Enhance the tourist experience** by recommending personalized destinations like Akagera National Park (Rwanda) for wildlife enthusiasts.
- 4. **Support sustainable tourism** by guiding visitors to eco-friendly lodges like Singita Kwitonda Lodge (Rwanda).
- 5. **Highlight cultural heritage** by suggesting visits to the Ethnographic Museum (Rwanda) or the Genocide Memorial.
- 6. **Encourage adventure tourism** by promoting activities like hiking the Virunga Mountains or kayaking on Lake Kivu.
- 7. **Promote relaxation tourism** by recommending serene destinations like Gisenyi's hot springs (Rwanda).
- 8. **Support conservation efforts** by directing tourists to eco-tourism initiatives like the Dian Fossey Gorilla Fund.

2. Key Stakeholders

The following stakeholders will benefit from this system:

Stakeholder	Role
1. Tourists	Travelers seeking personalized recommendations for destinations.
2. Tour Operators	Companies that organize tours and rely on tourist interest.
3. Local Communities	Communities near tourist sites that benefit from tourism revenue.
4. Government Tourism Boards	Organizations promoting tourism and managing destinations.

3. IF-THEN Rules

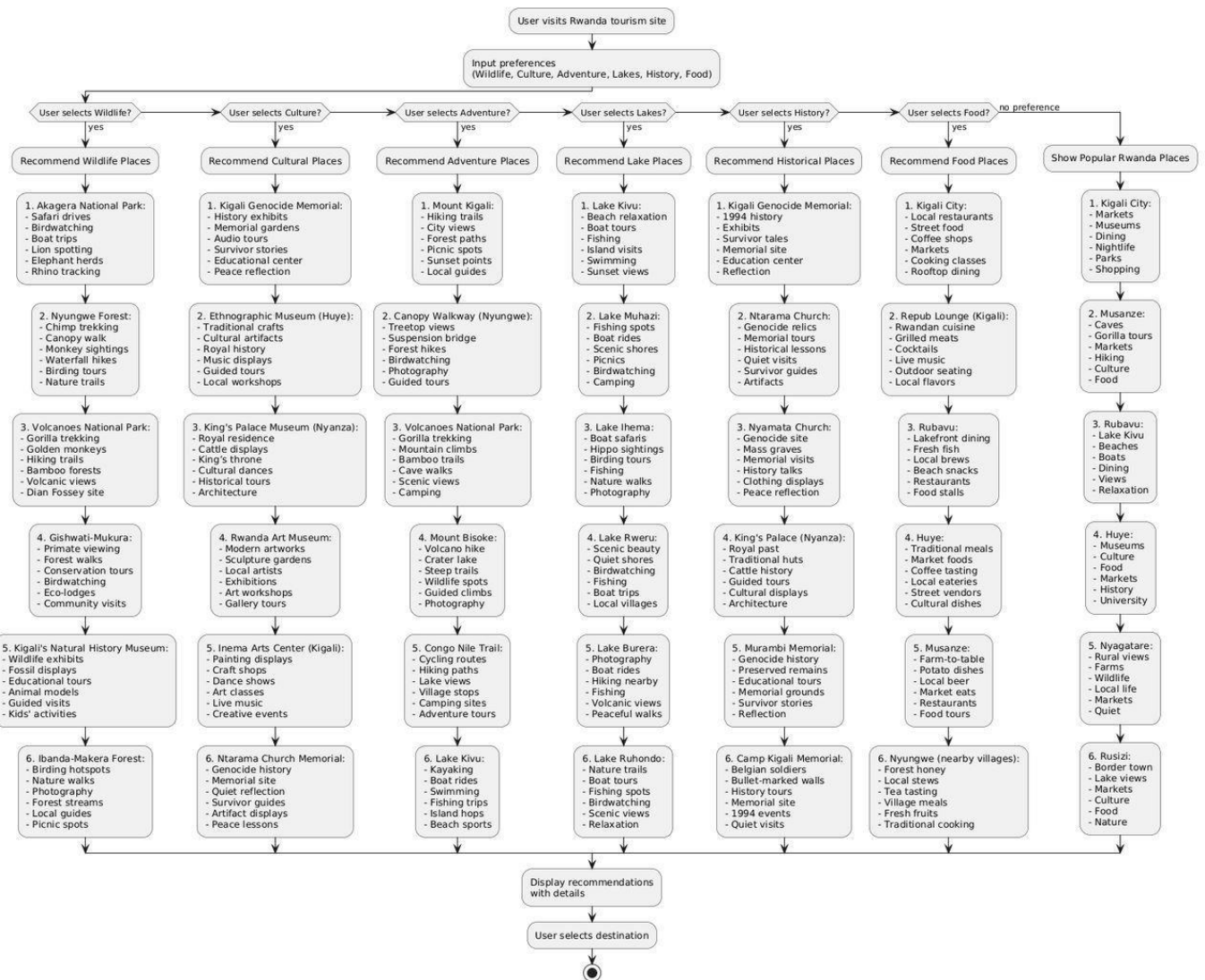
The system uses the following **IF-THEN rules** to recommend destinations:

Rule	Recommendation
IF the user prefers adventure tourism .	THEN recommend destinations like Volcanoes National Park, Mount Kigali, Canopy Walkway, Mount Bisoke, Congo Nile Trail, Lake Kivu (Rwanda).
IF the user prefers cultural tourism ,	THEN recommend destinations like the King's Palace Museum, Kigali Genocide Memorial, Ethnographic Museum (Huye), Rwanda Art Museum, Irema Arts Center (Rwanda).
IF the user prefers wildlife tourism ,	THEN recommend destinations like Akagera National Park, Nyungwe Forest, Gishwati-Mukura, Kigali's Natural History Museum, Ibanda-Makera Forest (Rwanda).
IF the user prefers History tourism ,	THEN recommend Ntarama Church, Nyamata Church, Murambi Memorial, Camp Kigali Memorial (Rwanda).
IF the user prefers relaxation tourism ,	THEN recommend destinations like Lake Kivu, Lake Muhazi, Lake Ihema, Lake Rweru, Lake Burea (Rwanda).
IF the user prefers food tourism ,	THEN recommend food destinations like Kigali City, Rubavu, Musanze, Huye (Rwanda).

Task 2.

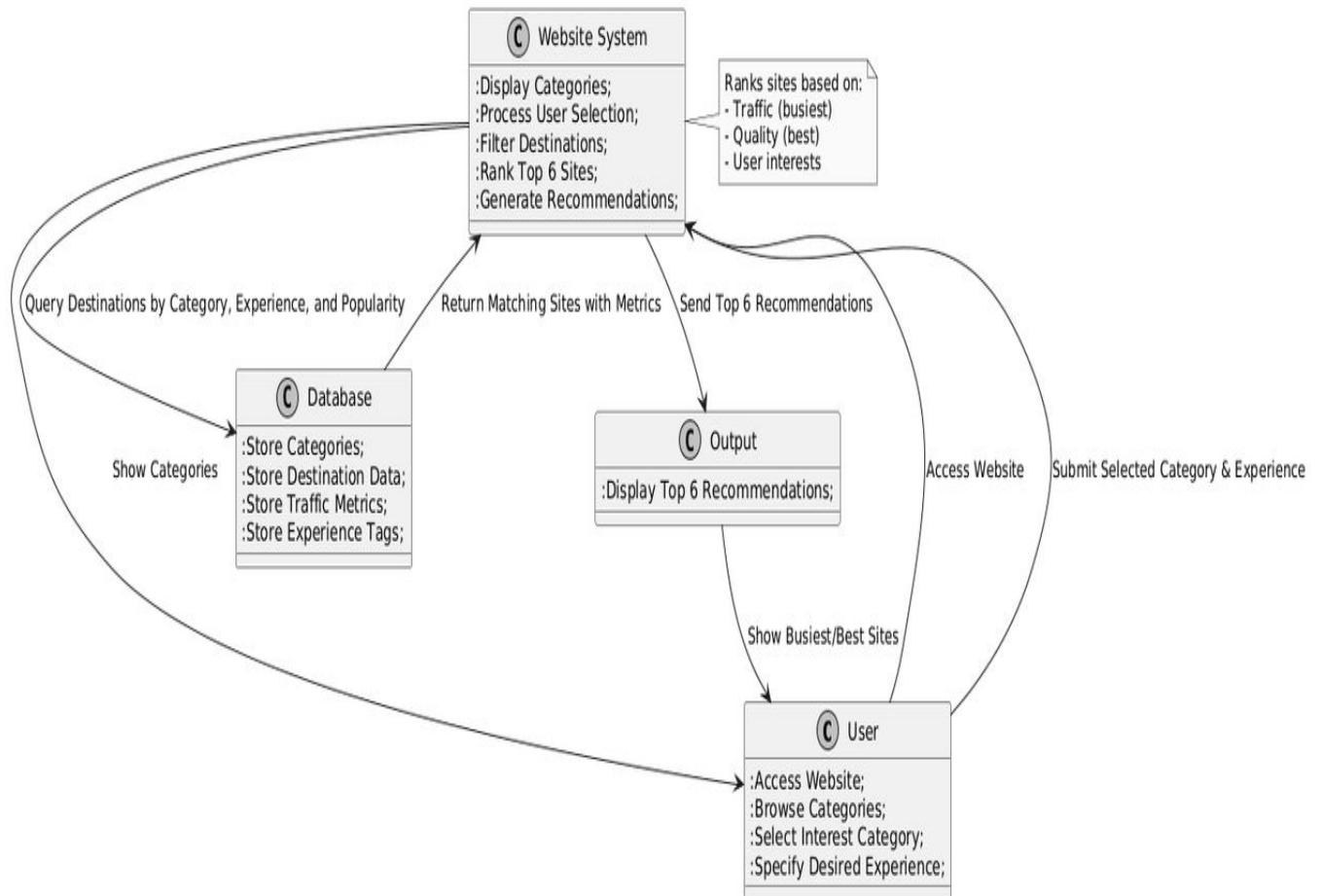
4. Flowchart

Flowcharts are visual representations of the steps or decisions your system will make. These help in understanding how various processes within the system are connected.



5. Data Flow Diagram (DFD)

A DFD shows how data moves through your system. It's a top-down view of how information will be processed, stored, and transferred. DFDs typically include



User Requirements Document for Rwanda Tourism Site

1. System Purpose:

The Rwanda Tourism Site is a web-based platform designed to provide users with detailed information about Rwanda's tourist attractions, accommodations, events, and other travel-related services. The site allows users to plan their trips, discover different tourist destinations, book tours, and access helpful travel resources.

2. User Inputs:

The users will provide the following inputs through various forms on the website:

- **Search Parameters:**
 - Destination (e.g., Kigali, Volcanoes National Park, Nyungwe Forest, Lake Kivu)
 - Date range for travel (start and end dates)
 - Number of travelers
 - Preferences for accommodation (e.g., luxury, budget, eco-friendly)
 - Tour type (e.g., adventure tours, cultural tours, safaris)
- **Personal Information:**
 - Name
 - Email address
 - Contact details for booking confirmations

3. Expected Outputs:

The system will return different outputs based on the user's input. These include:

- **Search Results:**
 - List of available destinations, accommodations, and tours based on the search criteria (e.g., available tours in Volcanoes National Park from March 1st to March 7th).
 - Display relevant details such as location, pricing, available amenities, and booking options.
- **Booking Confirmation:**
 - After a successful booking, the system will provide confirmation details, including a summary of the booking (e.g., tour dates, accommodation booked, total price).
- **Recommendation Engine:**
 - Based on the user's preferences or past behavior, recommend activities or destinations (e.g., "Based on your interest in wildlife, we recommend visiting Akagera National Park").
- **Travel Information:**
 - Provide general travel tips, visa information, and health guidelines for visiting Rwanda.
- **Error Messages:**
 - If a user inputs invalid data (e.g., dates are in the past), an error message should be displayed: "Please select valid travel dates."
- **Booking Error:**
 - If there are issues with payment, a message should appear: "Payment could not be processed. Please check your payment details."

4. System Features:

The system have several key features to enhance user experience and meet their needs. Some of the essential features are:

- **Tourist Information Pages:**
 - Detailed pages about various tourist attractions in Rwanda (e.g., Volcanoes National Park, Kigali city, Lake Kivu, Nyungwe Forest, and cultural landmarks).
- **Search Functionality:**
 - A search tool allowing users to search for tours and accommodations based on destination, travel dates, and preferences.
- **Tour Booking System:**
 - Users should be able to view available tours and book directly through the website, including the ability to select different packages (e.g., guided tours, private tours).
- **Accommodation Booking System:**
 - Users can view and book hotels, lodges, and other accommodation options, including details on price, location, and amenities.
- **User Account System:**
 - Users should be able to create an account to track their bookings, preferences, and past activities on the site.
- **Contact Form/Support:**
 - A contact form where users can reach customer service for inquiries or booking issues.

5. Error Handling:

The system should handle errors gracefully to improve user experience and ensure smooth navigation. Some possible error scenarios include:

- **Invalid Inputs:**
 - If a user selects travel dates that have already passed, an error message should appear: “Please choose a future date for your trip.”
- **Empty Fields:**
 - If required fields like name, email, or destination are missing, show a message: “Please complete all required fields.”
- **Unavailable Tours/Accommodations:**
 - If the selected tour or accommodation is no longer available, the system will notify the user: “This option is no longer available for your selected dates.”

6. User Interface (UI) Requirements:

The platform is a user-friendly, visually appealing, and responsive. Key UI elements of this platform include:

- **Homepage:**
 - A clean and inviting homepage with a search and navigation bar with different section of the content on site that allow user to navigate between content like, services, contact and booking tours.
 - Featured attractions and promotions, like "Top 10 Destinations in Rwanda."
- **Tour/Accommodation Listings:**
 - A listing page that displays search results, including filters for pricing, availability.
- **Booking Form:**
 - A simple form for users to input their personal details to book accommodation.
- **Mobile-Friendly Design:**
 - This web platform is a fully responsive web platform that work across different devices such as tablets, mobile phone, and desktop.

7. User Feedback:

- After a user completes an action (such as booking a tour or submitting a form), the system should provide clear feedback, such as:
 - **Success Messages:** "Your tour has been booked successfully!"
 - **Error Messages:** "Sorry, there was an issue with your payment. Please try again."

Example:

Let's say a user visits the Rwanda Tourism Site and is interested in a safari in Akagera National Park from 1st March to 7th March. They would:

1. **Input Information:**
 - Destination: Akagera National Park
 - Travel Dates: 1st March - 7th March
 - Number of travelers: 2
2. **Output from System:**
 - The system lists available tours in Akagera National Park, showing prices, tour dates, and available accommodation.
 - If the user proceeds, the system would prompt them to enter their personal and payment details.
3. **Post-booking:**
 - After booking, the user receives a confirmation email and a booking summary on the site: "Your safari booking is confirmed for 1st March to 7th March at Akagera National Park."

Screenshots of working Python code

```
import tkinter as tk
from tkinter import tk, messagebox
from PIL import Image, ImageTk
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from tkinter import Canvas
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.pdfgen import pdfgen
from datetime import datetime

# Data structures (experiences and destination info unchanged)
experiences = {
    "Wildlife": {
        "destinations": [
            "Akagera National Park": ["safari", "bird watching", "game drives"],
            "Nyungwe Forest National Park": ["chimp trekking", "canopy walk", "birding"],
            "Volcanoes National Park": ["gorilla trekking", "hiking", "wildlife"],
            "Gishwati-Mukura National Park": ["forest walks", "primate tracking", "nature"],
            "Kigali's Natural History Museum": ["exhibits", "education", "history"],
            "Banda-Makera Forest": ["nature walks", "bird watching", "exploration"]
        ]
    },
    "Cultural": {
        "destinations": [
            "Kigali Genocide Memorial": ["history", "education", "reflection"],
            "Ethnographic Museum (Huye)": ["artifacts", "culture", "history"],
            "King's Palace Museum (Nyanza)": ["royalty", "traditions", "history"],
            "Rwanda Art Museum": ["art", "culture", "exhibits"],
            "Inema Arts Center (Kigali)": ["contemporary art", "workshops", "culture"],
            "Ntarama Church Memorial": ["history", "memorial", "reflection"]
        ]
    },
    "Adventure": {
        "destinations": [
            "Mount Kigali": ["hiking", "views", "nature"],
            "Canopy Walkway (Nyungwe)": ["suspension bridge", "forest views", "adventure"],
            "Volcanoes National Park": ["gorilla trekking", "hiking", "climbing"],
            "Mount Bisoke": ["volcano hike", "crater lake", "adventure"],
            "Congo Nile Trail": ["trekking", "hiking", "scenic views"],
            "Lake Kivu": ["kayaking", "boat trips", "swimming"]
        ]
    },
    "Rwanda's Lakes": {
        "destinations": [
            "Lake Kivu": ["boating", "fishing", "relaxation"],
            "Lake Muhazi": ["fishing", "bird watching", "boat rides"],
            "Lake Ihema": ["boat tours", "wildlife", "fishing"],
            "Lake Rwenzori": ["scenic views", "fishing", "nature"],
            "Lake Burera": ["canoeing", "views", "relaxation"],
            "Lake Ruhondo": ["boat trips", "scenic beauty", "nature"]
        ]
    },
    "Rwanda's History": {
        "destinations": [
            "Kigali Genocide Memorial": ["history", "education", "memorial"],
            "Ntarama Church": ["history", "memorial", "reflection"],
            "Nyanza Church": ["history", "memorial", "education"],
            "King's Palace (Nyanza)": ["royalty", "history", "culture"],
            "Murambi Memorial": ["history", "education", "memorial"],
            "Camp Kigali Memorial": ["history", "military", "memorial"]
        ]
    },
    "Rwanda Food": {
        "destinations": [
            "Kigali City": ["restaurants", "local cuisine", "markets"],
            "Repub Lounge (Kigali)": ["dinner", "cocktails", "atmosphere"],
            "Rubavu": ["lakefront dining", "fresh fish", "local food"],
            "Huye": ["traditional meals", "markets", "culture"],
            "Musanze": ["local cuisine", "fresh produce", "dining"],
            "Nyungwe (Nearby Villages)": ["village food", "traditional cooking", "culture"]
        ]
    }
}

destination_info = {
    "Akagera National Park": ("Eastern Rwanda", "A savannah park with diverse wildlife including lions and rhinos. Expect thrilling game drives and bird watching."),
    "Nyungwe Forest National Park": ("Southwestern Rwanda", "A lush rainforest with chimpanzee trekking and a famous canopy walkway offering stunning views."),
    "Volcanoes National Park": ("Northern Rwanda", "Home to mountain gorillas and volcanic landscapes. Experience gorilla trekking and scenic hikes."),
    "Gishwati-Mukura National Park": ("Western Rwanda", "A regenerating forest with primate tracking and peaceful nature walks."),
    "Kigali's Natural History Museum": ("Kigali", "Learn about Rwanda's biodiversity and geological history through engaging exhibits."),
    "Banda-Makera Forest": ("Southern Rwanda", "A hidden gem for nature walks and bird watching in a tranquil setting."),
    "Kigali Genocide Memorial": ("Kigali", "A poignant site to reflect on Rwanda's history with educational exhibits."),
    "Ethnographic Museum (Huye)": ("Huye, Southern Rwanda", "Explore Rwanda's cultural heritage with artifacts and traditional displays."),
    "King's Palace Museum (Nyanza)": ("Nyanza, Southern Rwanda", "Step into royal history with a replica palace and cultural insights."),
    "Rwanda Art Museum": ("Kigali", "Admire Rwandan art in a historic setting, once a presidential palace."),
    "Inema Arts Center (Kigali)": ("Kigali", "Vibrant contemporary art and workshops showcasing local talent."),
    "Ntarama Church Memorial": ("Near Kigali", "A somber monument of the genocide with a powerful historical narrative."),
    "Mount Kigali": ("Kigali", "Hike for panoramic city views and a refreshing nature escape."),
    "Canopy Walkway (Nyungwe)": ("Nyungwe Forest", "Walk a suspension bridge high above the forest for breathtaking views."),
    "Mount Bisoke": ("Northern Rwanda", "Climb a volcano to see a stunning crater lake and rugged terrain."),
    "Congo Nile Trail": ("Western Rwanda", "Cycle or hike along Lake Kivu with scenic villages and landscapes."),
    "Lake Kivu": ("Eastern Rwanda", "Enjoy boating and relaxation on one of Africa's great lakes."),
    "Lake Muhazi": ("Eastern Rwanda", "A serene lake perfect for fishing and bird watching."),
    "Lake Burera": ("Eastern Rwanda, Akagera", "Boat tours with hippos and crocodiles in a wildlife-rich setting."),
    "Lake Rwenzori": ("Northern Rwanda", "Quiet waters with scenic beauty and fishing opportunities."),
    "Lake Burera": ("Northern Rwanda", "Canoe among volcanic views and tranquil waters."),
    "Lake Ruhondo": ("Northern Rwanda", "A peaceful lake with boat trips and natural beauty."),
    "Nyanza Church": ("Near Kigali", "A genocide memorial site offering historical reflection."),
    "Murambi Memorial": ("Southern Rwanda", "A stark reminder of the genocide with preserved history."),
    "Camp Kigali Memorial": ("Kigali", "Honors Belgian peacekeepers with historical significance."),
    "Kigali City": ("Kigali", "Savor local cuisine and vibrant markets in the capital."),
    "Repub Lounge (Kigali)": ("Kigali", "Upscale dining with cocktails and a lively atmosphere."),
    "Rubavu": ("Western Rwanda, Lake Kivu", "Fresh fish and lakefront dining."),
    "Huye": ("Southern Rwanda", "Traditional meals with a cultural market vibe."),
    "Musanze": ("Northern Rwanda", "Treat local dishes near volcanic landscapes."),
    "Nyungwe (Nearby Villages)": ("Southwestern Rwanda", "Authentic village food and traditions.")
}

class TourismApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Rwanda Tourism Planner")
        self.root.geometry("1200x900")
        self.root.config(bg="#F0F4F8")
        self.bookings = []

        self.main_frame = tk.Frame(root, padding=20)
        self.main_frame.pack(fill="both", expand=True)

        self.left_frame = tk.Frame(self.main_frame, relief="groove", borderwidth=2)
        self.left_frame.grid(row=0, column=0, sticky="nsew", padx=10, pady=10)
        self.create_info_form()

        self.right_frame = tk.Frame(self.main_frame, relief="groove", borderwidth=2)
        self.right_frame.grid(row=0, column=1, sticky="nsew", padx=10, pady=10)
        self.create_recommendation_section()

        self.main_frame.columnconfigure(0, weight=1)
        self.main_frame.columnconfigure(1, weight=2)
        self.main_frame.rowconfigure(0, weight=1)

        style = tk.Style()
        style.configure("Accent.Button", foreground="white", background="#4CAF50", font=("Arial", 10, "bold"))
        style.configure("Title.Label", background="#4CAF50", color="white")
        style.configure("Title.Label.Frame", background="#F0F4F8")

        self.activity_vars = {}
        self.image_label = None
        self.book_button = None
        self.plot_canvas = None

    def create_info_form(self):
        tk.Label(self.left_frame, text="Your Information", font=("Arial", 14, "bold")).pack(pady=10)
        tk.Label(self.left_frame, text="Full Name:").pack(pady=5)
        self.name_entry = tk.Entry(self.left_frame)
        self.name_entry.pack(fill="x", pady=10)
        tk.Label(self.left_frame, text="Email:").pack(pady=5)
        self.email_entry = tk.Entry(self.left_frame)
        self.email_entry.pack(fill="x", pady=10)
```

```

    tk.Label(self.left_frame, text="Age:").pack(pady=5)
    self.age_entry = tk.Entry(self.left_frame)
    self.age_entry.pack(fill="x", pady=10)
    tk.Label(self.left_frame, text="Departure Date:").pack(pady=5)
    self.depart_cal = Calendar(self.left_frame, selectmode="day", date_pattern="yyyy-mm-dd")
    self.depart_cal.pack(pady=10)
    tk.Label(self.left_frame, text="Return Date:").pack(pady=5)
    self.return_cal = Calendar(self.left_frame, selectmode="day", date_pattern="yyyy-mm-dd")
    self.return_cal.pack(pady=10)
    tk.Button(self.left_frame, text="View Active Schedules", command=self.show_active_schedules).pack(pady=10)

def create_recommendation_section(self):
    tk.Label(self.right_frame, text="Recommendation System", font=("Arial", 14, "bold")).pack(pady=10)
    self.exp_frame = tk.LabelFrame(self.right_frame, text="Choose Your Experience", padding=10)
    self.exp_frame.pack(fill="x", pady=10)
    tk.Label(self.exp_frame, text="Select Experience Type:").grid(row=0, column=0, padx=5, pady=5)
    self.experience_var = tk.StringVar()
    self.experience_combo = tk.Combobox(self.exp_frame, textvariable=self.experience_var,
                                       values=list(experiences.keys()), state="readonly")
    self.experience_combo.grid(row=0, column=1, padx=5, pady=5)
    self.experience_combo.bind("<<ComboboxSelected>>", self.load_activities)

    self.activities_frame = tk.LabelFrame(self.right_frame, text="Select Activities", padding=10)
    self.activities_frame.pack(fill="both", expand=True, pady=5)

    self.recommend_frame = tk.Frame(self.right_frame)
    self.recommend_frame.pack(fill="x", pady=5)
    tk.Button(self.recommend_frame, text="Get Recommendation", command=self.recommend_destination).pack(pady=5)
    self.result_var = tk.StringVar()
    self.result_label = tk.Label(self.recommend_frame, textvariable=self.result_var, wraplength=600, justify="left")
    self.result_label.pack(pady=5)

def load_activities(self, event=None):
    for widget in self.activities_frame.winfo_children():
        widget.destroy()
    self.activity_vars.clear()
    experience = self.experience_var.get()
    if not experience:
        return
    all_activities = set()
    for activities in experiences[experience]["destinations"].values():
        all_activities.update(activities)
    for i, activity in enumerate(all_activities):
        tk.Checkbutton(self.activities_frame, text=activity, variable=self.activity_vars[activity]).grid(row=i//2, column=i%2, sticky="w",
        padx=5)

def recommend_destination(self):
    # Input validation checks
    if not all([self.name_entry.get(), self.email_entry.get(), self.age_entry.get(),
               self.depart_cal.get_date(), self.return_cal.get_date()]):
        messagebox.showwarning("Input Error", "Please fill in all personal information first.")
        return
    experience = self.experience_var.get()
    if not experience:
        messagebox.showwarning("Input Error", "Please select an experience.")
        return
    selected_activities = [act for act, var in self.activity_vars.items() if var.get()]
    if not selected_activities:
        messagebox.showwarning("Input Error", "Please select at least one activity.")
        return
    # START OF RULE BASED LOGIC
    rule_based_system = RecommendationSystem(selected_activities, experiences["destinations"])
    best_match = None
    max_score = 0
    scores = {}
    for destination, activities in destinations.items():
        # Rule 1: Count matching activities
        score = sum(1 for act in selected_activities if act in activities)
        # Rule 2: Apply weights based on destination type
        weight = 1.0 if "Beach" in destination else 0.8 if "Hill" in destination else 0.5 if "Mountain" in destination else 0.3
        weighted_score = score * weight
        scores[destination] = weighted_score
    # Rule 3: Select the destination with the highest weighted score
    if scores:
        best_match = max(scores, key=scores.get)
    # END OF RULE BASED LOGIC
    # Process the result of the rule-based recommendation
    if best_match:
        location, desc = destination_info[best_match]
        match_count = sum(1 for act in selected_activities if act in destinations[best_match])
        self.result_var.set(f"Recommended: {best_match}\nLocation: {location}\nExperience: {desc}\nMatches: {match_count}/{len(selected_activities)}")
        self.selected_destination = best_match
        if self.image_label:
            self.image_label.destroy()
        try:
            img = Image.open(f"images/{best_match}.jpg")
            img = img.resize((200, 200), Image.Resampling.LANCZOS)
            photo = ImageTk.PhotoImage(img)
            self.image_label = tk.Label(self.recommend_frame, image=photo)
            self.image_label.image = photo
            self.image_label.pack(pady=5)
        except FileNotFoundError:
            self.result_var.set(self.result_var.get() + "\n(Image not available)")
        self.plot_scores(scores)
        if self.book_button:
            self.book_button.destroy()
        self.book_button = tk.Button(self.recommend_frame, text="Book Now", command=self.create_ticket,
        style="Accent.Tuple")
        self.book_button.pack(pady=10)
    else:
        self.result_var.set("No suitable destination found.")

def plot_scores(self, scores):
    if self.plot_canvas:
        self.plot_canvas.get_tk_widget().destroy()
    fig = plt.figure(figsize=(6, 3))
    ax = fig.add_subplot(111)
    ans.barplot(x=list(scores.keys()), y=list(scores.values()), palette="viridis", ax=ax)
    ax.set_xlabel("Destinations", fontsize=10)
    ax.set_ylabel("Score", fontsize=10)
    ax.set_title("Match Scores", fontsize=12)
    fig.tight_layout()
    self.plot_canvas = FigureCanvasTkAgg(fig, master=self.recommend_frame)
    self.plot_canvas.draw()
    self.plot_canvas.get_tk_widget().pack(fill="x", pady=5)

def create_ticket(self):
    user_data = {
        "full_name": self.name_entry.get(),
        "email": self.email_entry.get(),
        "age": self.age_entry.get(),
        "departure_date": self.depart_cal.get_date(),
        "return_date": self.return_cal.get_date(),
        "destination": getattr(self, "selected_destination", None),
        "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    }
    if not all(user_data.values()):
        messagebox.showwarning("Input Error", "Please complete all fields and get a recommendation.")
        return
    try:
        depart_date = datetime.strptime(user_data["departure_date"], "%Y-%m-%d")
        return_date = datetime.strptime(user_data["return_date"], "%Y-%m-%d")
        user_data["duration"] = (return_date - depart_date).days
    except ValueError:
        messagebox.showwarning("Input Error", "Dates must be in YYYY-MM-DD format.")
        return
    self.bookings.append(user_data)
    filename = f"ticket_{user_data['full_name']}_{datetime.now().strftime('%Y%m%d_%H%M%S')}.pdf"
    self.create_pdf_ticket(user_data, filename)
    self.show_booking_info(user_data, filename)

def create_pdf_ticket(self, user_data, filename):
    c = Canvas(Canvas(filename, pagesize=letter))
    c.setFont("Helvetica-Bold", 10)
    c.drawString(100, 70, f"Name: {user_data['full_name']}")
    c.drawString(100, 75, f"Email: {user_data['email']}")
    c.drawString(100, 80, f"Age: {user_data['age']}")
    c.drawString(100, 85, f"Departure: {user_data['departure_date']}")
    c.drawString(100, 90, f"Return: {user_data['return_date']}")
    c.drawString(100, 95, f"Duration: {user_data['duration']} days")
    c.drawString(100, 100, f"Destination: {user_data['destination']}")
    c.drawString(100, 105, f"Experience: {destination_info[user_data['destination']]['desc']}")
    # Activity Schedule
    c.setFont("Helvetica-Bold", 14)
    c.drawString(100, 110, "Activity Schedule")
    c.setFont("Helvetica", 10)
    v_pos = 120
    activities = experiences[self.experience_var.get()]["destinations"][user_data['destination']]
    start_time = "08:00 AM"
    for i, activity in enumerate(activities):
        time = f"{int(start_time[:2]) + 1:02d}:{int(start_time[3:5]) + 1:02d}:00 PM" if i < 5 else f"{int(start_time[:2]) + 1:02d}:{int(start_time[3:5]) + 1:02d}:00 PM"
        c.drawString(120, v_pos, f"{i+1}: {activity} {time}")
        v_pos += 15
    # Travel Tips
    c.setFont("Helvetica-Bold", 14)
    c.drawString(100, v_pos, "Travel Tips")
    c.setFont("Helvetica", 10)
    c.drawString(120, v_pos, "1. Weather: Expect warm days (20-30°C), pack light clothing & rain gear.")
    c.drawString(120, v_pos, "2. Currency: Rwandan Franc (RWF), ATMs widely available in Kigali.")
    c.drawString(120, v_pos, "3. Pack: Sunscreen, insect repellent, comfortable shoes.")

```

```

tourism_recommender_gui

# Contact Information
c.setFont("Helvetica-Bold", 14)
c.drawString(100, y_pos - 85, "Contact Information")
c.setFont("Helvetica", 10)
c.drawString(120, y_pos - 100, "- Emergency: 112 (Police/Medical)")
c.drawString(120, y_pos - 115, "- Rwanda Tourism: +250 791 815 780")
c.drawString(120, y_pos - 115, "- Rwanda Tourism: +250 792 086 409")

# Map Reference
c.setFont("Helvetica-Bold", 14)
c.drawString(100, y_pos - 135, "Map Reference")
c.setFont("Helvetica", 10)
region = destination_info[user_data['destination']][0].split(',')[1]
c.drawString(120, y_pos - 150, f"- Located in {region}. Check local maps for exact routes.")

# Footer
c.drawString(100, y_pos - 170, f"Issued: {user_data['timestamp']}")
c.save()

def show_booking_info(self, user_data, filename):
    info_window = tk.Toplevel(self.root)
    info_window.title("Your Booking Details")
    info_window.geometry("500x400")
    info_window.configure(bg="#f0f4f8")

    ttk.Label(info_window, text="Your Booking Information", font=("Arial", 14, "bold")).pack(pady=10)
    info_text = (f"Name: {user_data['full_name']}\n"
                f"Email: {user_data['email']}\n"
                f"Age: {user_data['age']}\n"
                f"Departure: {user_data['departure_date']}\n"
                f"Return: {user_data['return_date']}\n"
                f"Destination: {user_data['destination']}\n"
                f"Location: {destination_info[user_data['destination']][0]}\n"
                f"Experience: {destination_info[user_data['destination']][1]}\n"
                f"Ticket Saved As: {filename}")
    ttk.Label(info_window, text=info_text, wraplength=450, justify="left").pack(pady=10)
    ttk.Button(info_window, text="Close", command=info_window.destroy).pack(pady=10)

    self.reset_form()

def show_active_schedules(self):
    schedule_window = tk.Toplevel(self.root)
    schedule_window.title("Active Schedules")
    schedule_window.geometry("600x400")

    ttk.Label(schedule_window, text="Active Bookings", font=("Arial", 14, "bold")).pack(pady=10)
    text = tk.Text(schedule_window, height=20, width=70)
    text.pack(pady=10)

    for booking in self.bookings:
        entry = (f"Name: {booking['full_name']} | "
                f"Destination: {booking['destination']} | "
                f"Depart: {booking['departure_date']} | "
                f"Return: {booking['return_date']}\n")
        text.insert(tk.END, entry)
    text.config(state="disabled")

    ttk.Button(schedule_window, text="Close", command=schedule_window.destroy).pack(pady=10)

def reset_form(self):
    self.name_entry.delete(0, tk.END)
    self.email_entry.delete(0, tk.END)
    self.age_entry.delete(0, tk.END)
    self.depart_cal.selection_clear()
    self.return_cal.selection_clear()
    self.result_var.set("")
    self.experience_var.set("")
    for widget in self.activities_frame.winfo_children():
        widget.destroy()
    if self.image_label:
        self.image_label.destroy()
    if self.book_button:
        self.book_button.destroy()
    if self.plot_canvas:
        self.plot_canvas.get_tk_widget().destroy()
    self.activity_vars.clear()

if __name__ == "__main__":
    root = tk.Tk()
    app = TourismApp(root)
    root.mainloop()

```

User Manuel

Tourism Website Overview:

The website features a **search engine** and a **navigation bar** to help users easily find what they need.



Search Engine:

- Users can search for specific content by entering keywords or phrases (e.g., "services").

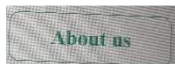
Navigation Bar:



- **Home:** Takes users to the homepage.



- **Services:** Lists different services like luxury hotels, fastest travel, food, recommendations, etc.



- **About Us:** Provides information about the website and what it offers.



- **Places:** Shows various tourist attractions in Rwanda.



- **To-Do:** Allows users to add, edit, or delete places they want to visit.



- **Contact Us:** Displays contact details and company links.

User Requirements for the Tourism Platform:

Functional Requirements:

These describe what the platform should do:

1. **Search and Discovery:**
 - Users can search for destinations, hotels, tours, and activities.
 - They can view details about destinations, hotels, and activities.
2. **Booking and Reservations:**
 - Users can book hotels, flights, tours, and activities.
 - They can view booking details, receive confirmations, and modify or cancel bookings.
3. **Itinerary Planning:**
 - Users can create and save travel itineraries.
 - Users can add or remove activities from their itineraries.
4. **Customer Support:**
 - Users can access a help center or FAQ section.
 - They can contact support via chat, email, or phone.

Non-Functional Requirements:

These describe how the platform should perform:

1. **Performance:**
 - The platform should load quickly (under 3 seconds).
 - It should handle many users at once.
2. **Usability:**
 - The platform should have a user-friendly and easy-to-navigate design.
 - It should work on both desktop and mobile devices.
3. **Reliability:**
 - The platform should have minimal downtime (99.9% uptime).
 - It should provide accurate, up-to-date information.
4. **Compatibility:**
 - The platform should work on major browsers (Chrome, Firefox, Safari, Edge).
 - It should be compatible with Windows, macOS, Android, and iOS.

User Stories:

These describe the experience from the user's perspective.

1. **Traveler:**
 - "I want to search for hotels by location and filter by price/rating to find the best option."
 - "I want to book a hotel and get a confirmation email."
2. **Tourist:**
 - "I want to view details about attractions to decide where to visit." o "I want to create an itinerary to organize my trip." o "I want to read reviews from other travelers."
3. **Business User:**
 - "I want to list my hotel or tour to attract customers." o "I want to manage bookings and availability." o "I want to respond to customer reviews."

Tools for Gathering User Requirements:

These tools help collect user feedback to improve the website:

1. **Surveys and Questionnaires:** Use tools like Google Forms or SurveyMonkey to gather user opinions.
2. **Interviews:** Speak directly with users to understand their needs.
3. **Focus Groups:** Organize discussions with multiple users to gain insights.