

INSTITUT D'ENSEIGNEMENT SUPÉRIEUR DE RUHENGERI

Accredited by Ministerial Order N°005/2010/MINÉDUC of 16 June 2010



Scientia et Lux

11th December, 2024

TECHNICAL REPORT

Faculty: AFS

Department: Computer Science

class: SWE A year3

Course: Agile Software Development

Project Name: Rwd-Travelagency Website

Group Members:

ABARI ILIOR Aichetou

CHOL Adut Gai

AbdulAzeez Abubakar

Uwera Denyse

B.P. 155

Ruhengeri
Rwanda

T : +250 788 90 30 30

F : +250 788 90 30 32

W : www.ines.ac.rw

I. INTRODUCTION:

This report details the development of the Rwd-Travelagency website, designed to showcase Rwandan culture and offer streamlined travel services. The project utilized software reuse techniques to accelerate development and ensure reliability by integrating pre-existing components. The website provides users with essential travel information, and booking capabilities ensuring a seamless experience. Utilizing Agile methodologies, the project was executed iteratively to adapt to evolving requirements and ensure user satisfaction. Key features include multilingual support to enhance user engagement, and the integration of Tkinter GUI forms for booking activities, flight tickets to Kigali, and hotel accommodations. These forms are connected to the website using Flask and supported by an SQL database for reliable data management. delivering a seamless user experience.

II. PROJECT PLANNING:

- **Objective:**
 - Provide users with a seamless platform for exploring and booking travel services.
 - Highlight Rwanda's cultural heritage to attract global travelers.
 - Enable robust and efficient booking through GUI-based forms integrated into the website.
- **User stories:**
 - "As a traveler, I want to book activities, flights, and hotels easily through an online platform."
 - "As a customer, I want to select my preferred language to ensure smooth communication during my journey."
 - "As a user, I want reliable data storage to ensure my bookings are secure."
 - "As a potential customer, I want to explore travel offers and learn about Rwandan culture on the website."
- **Sprint Planning:**
 - **Sprint 1:** Integration of pre-existing design templates for the static pages (Home, offers, ...).
 - **Sprint 2:** Implementation of Tkinter form for booking and language selection, Develop and connect Tkinter forms using Flask.
 - **Sprint 3:** Implement SQL database integration and perform testing.

- **Scrum Practices:**
Daily stand-ups focused on progress updates and identifying bottlenecks. Sprint reviews ensured constant feedback incorporation.

III. SYSTEM DESIGN:

- **Website Structure:**
 - **Pages:** History, Offers, Contact Us, Reservation.
 - **Reservation Forms:**
 - Activities Booking
 - Flight Tickets to Kigali
 - Hotel Accommodations
- **Architecture:**
 - **Front-end:** reusable HTML, CSS, and JavaScript were adapted for static content and responsiveness, and Flask templates for dynamic content.
 - **Back-end:** Flask framework to manage dynamic features like forms and database interactions and to handle server-side logic and integrate Tkinter forms.
 - **Database:** SQL for data storage, managing bookings, and user information.
- **Reused Components:**
 - Pre-designed CSS frameworks like Bootstrap for layout styling.
- **Integration:**
Tkinter GUIs for reservation forms were connected to Flask endpoints, ensuring smooth data flow between the forms and the website.

IV. IMPLEMENTATION:

- **Key Features:**
 - About us page to know our agency's history and mission.
 - Offers page presenting promotions and activities.
 - Contact Us page for inquiries and support.
 - Reservation page with three forms connected via Flask:
 - Activity Booking Form
 - Flight Ticket Booking Form
 - Hotel Accommodation Form

- **Tools and Technologies:**
 - Development: Python(Tkinter, Flask), HTML, CSS, Bootstrap.
 - Database: SQL for structured data storage.
 - Frameworks: Flask for back-end integration and for integration with the HTML file.
 - Reuse Sources: open-source libraries, and templates.
- **Challenges and Solutions:**
 - **Challenge:** Adapting reusable components to meet specific requirements.
Solution: Customized components while retaining the core reusable functionality.
 - **Challenge:** Connecting Tkinter GUI to the web application.
Solution: Used Flask's routing and data handling capabilities to integrate Tkinter forms.
 - **Challenge:** Ensuring smooth data flow between forms and the database.
Solution: Implemented structured SQL queries and Flask ORM for reliable data management.
 - **Challenge:** Ensuring compatibility across reused modules.
Solution: Conducted thorough testing to identify and resolve conflicts.
 - **Challenge:** Ensuring form validations and error handling were user-friendly.
Solution: Flask's built-in validation features with customized error messages were used.

V. TESTING AND DEPLOYMENT:

- **Testing:**
 - Functional testing to verify form submissions, database interactions, and multilingual options.
 - Usability testing to ensure the interface was intuitive for users.
 - Integration testing ensured seamless communication between Tkinter, Flask, and the SQL database.
- **Feedback:**

User feedback led to refined the user interface, and improved error messages for clarity.
- **Deployment:**

Hosted on a local server during development; planned deployment on a cloud platform for live access.

VI. REFLECTION AND LESSONS LEARNED:

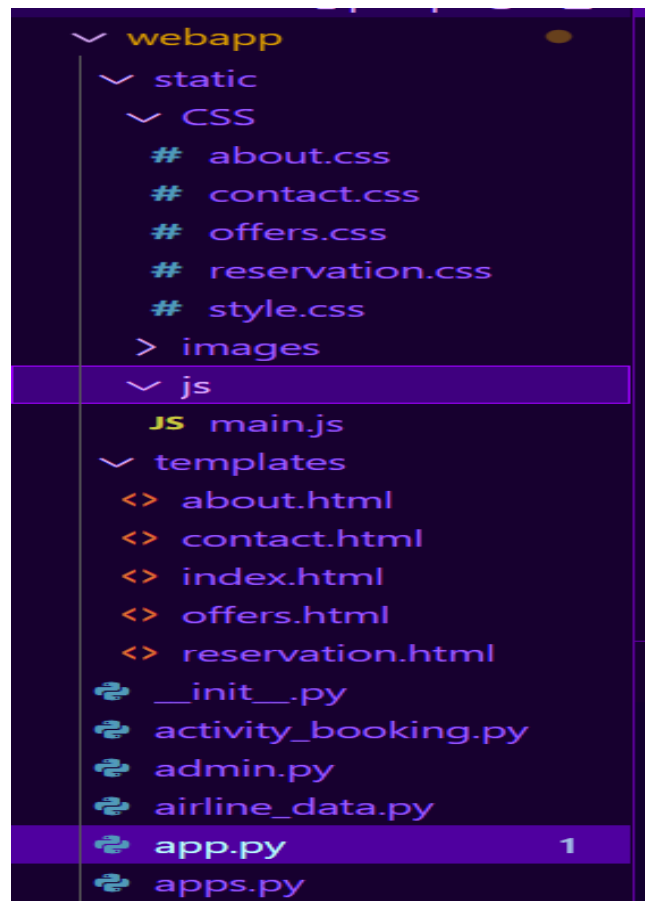
This project highlighted the importance of software **reuse** in accelerating development and ensuring reliability, while also demonstrating the value of integrating desktop and web technologies, such as Tkinter and Flask, to create a seamless user experience. Agile practices facilitated iterative feedback and user-focused customization, allowing for effective iterative development. Additionally, the use of SQL databases ensured secure and reliable data management. Challenges in integration were addressed through structured planning and testing, providing valuable lessons for future projects.

VII. CONCLUSION:

The Rwd-Travelagency website successfully combines dynamic features like Tkinter-based reservation forms, Flask integration, and SQL data management. By promoting Rwandan culture and offering user-friendly booking tools, the platform meets its objective of attracting global travelers while maintaining a reliable and efficient backend system. These elements highlight Rwanda's cultural richness and provide seamless user experiences.

VIII. APPENDICES:

- Code snippets:
 - Structure of the system:



➤ connection:

webapp >  urls.py > ...

```
1 from django.urls import path
2 from . import views
3
4
5 urlpatterns = [
6     path('', views.index, name='index'), # Home page
7     path('about/', views.about, name='about'), # About page
8     path('offers/', views.offers, name='offers'), # Offers page
9     path('reservation/', views.reservation, name='reservation'), # Reservation
10    path('contact/', views.contact, name='contact'),
11    path('reservation1/', views.reservation1, name='reservation1'),
12    path('gui/', views.gui_view, name='gui'),
13    path('open_gui/', views.open_gui, name='open_gui'),
14
15 ]
16
```

webapp >  views.py > ...

```
1 # views.py
2 from django.shortcuts import render
3 from django.http import HttpResponse
4
5 def index(request):
6     return render(request, 'index.html')
7
8 def about(request):
9     return render(request, 'about.html')
10
11 def offers(request):
12     return render(request, 'offers.html')
13
14 def reservation(request):
15     return render(request, 'reservation.html')
16
17 def contact(request):
18     return render(request, 'contact.html')
19
20 def reservation1(request):
21     return render(request, 'reservation1.html')
22
23
```

```

<section class="header">
  <nav class="navbar">
    <ul class="nav-list">
      <li><a href="{{ url_for('index') }}">Home</a></li>
      <li><a href="{{ url_for('about') }}">About</a></li>
      <li><a href="{{ url_for('offers') }}">Offers</a></li>
      <li><a href="{{ url_for('reservation') }}">reservation</a></li>
      <li><a href="{{ url_for('contact') }}">contact</a></li>
    </ul>
  </nav>
</section>

```

```

Agile_assignment > settings.py > ...
74 ROOT_URLCONF = 'Agile_assignment.urls'
75 INSTALLED_APPS = [
76     'django.contrib.admin',
77     'django.contrib.auth',
78     'django.contrib.contenttypes',
79     'django.contrib.sessions',
80     'django.contrib.messages',
81     'django.contrib.staticfiles',
82     'webapp',
83 ]
84 TEMPLATES = [
85     {
86         'BACKEND': 'django.template.backends.django.DjangoTemplates',
87         'DIRS': [],
88         'APP_DIRS': True,
89         'OPTIONS': {
90             'context_processors': [
91                 'django.template.context_processors.debug',
92                 'django.template.context_processors.request',
93                 'django.contrib.auth.context_processors.auth',
94                 'django.contrib.messages.context_processors.messages',
95             ],
96         },
97     },
98 ]
99 # Database
100 # https://docs.djangoproject.com/en/5.1/ref/settings/#databases
101 STATIC_URL = 'static/'
102
103 DATABASES = {
104     'default': {
105         'ENGINE': 'django.db.backends.sqlite3',
106         'NAME': BASE_DIR / 'db.sqlite3',
107     }
108 }
109

```

➤ forms:

```
8bapp / hotel_booking.py / ~
2 from tkinter import messagebox, ttk
3 import mysql.connector
4 from gui_functions import show_calendar
5 from airline_data import hotels
6
7 DB_HOST = "localhost"
8 DB_NAME = "travel_booking" # Replace with your database name
9 DB_USER = "root" # Default username for XAMPP MySQL
10 DB_PASSWORD = "" # Default password for XAMPP MySQL (usually empty)
11
12 def insert_hotel_booking(
13     selected_hotel,
14     room_type1,
15     room_type2,
16     room_type3,
17     num_rooms1,
18     num_rooms2,
19     num_rooms3,
20     check_in,
21     check_out,
22     total_price,
23 ):
24     conn = mysql.connector.connect(
25         host=DB_HOST, database=DB_NAME, user=DB_USER, password=DB_PASSWORD
26     )
27     cursor = conn.cursor()
28     cursor.execute(
29         """
30         INSERT INTO hotel_bookings (selected_hotel, room_type1, room_type2, room_type3, num_rooms1, num_rooms2, num_rooms3, check_in, check_out, total_price)
31         VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
32         """
33     )
34     (
35         selected_hotel,
36         room_type1,
37         room_type2,
38         room_type3,
39         num_rooms1,
40         num_rooms2,
41         num_rooms3,
42         check_in,
43         check_out,
44         total_price,
45     ),
46 )
```

```
50 def open_hotel_form():
51     hotel_form = Tk()
52     hotel_form.title("Hotel Booking Form")
53     hotel_form.geometry("400x400")
54
55     # Room Type Selection
56     room_type_label = tk.Label(hotel_form, text="Room Type Selection")
57     room_type_label.grid(row=0, column=0, padx=10, pady=10, sticky="w")
58
59     # Number of Rooms
60     num_rooms_label = tk.Label(hotel_form, text="Number of Rooms")
61     num_rooms_label.grid(row=1, column=0, padx=10, pady=10, sticky="w")
62
63     num_rooms1_entry = tk.Entry(hotel_form, width=30)
64     num_rooms1_entry.grid(row=1, column=1, padx=10, pady=10)
65
66     num_rooms2_entry = tk.Entry(hotel_form, width=30)
67     num_rooms2_entry.grid(row=1, column=2, padx=10, pady=10)
68
69     num_rooms3_entry = tk.Entry(hotel_form, width=30)
70     num_rooms3_entry.grid(row=1, column=3, padx=10, pady=10)
71
72     # Check-in and Check-out Dates
73     check_in_label = tk.Label(hotel_form, text="Check-in Date")
74     check_in_label.grid(row=2, column=0, padx=10, pady=10, sticky="w")
75
76     check_in_entry = tk.Entry(hotel_form, width=30)
77     check_in_entry.grid(row=2, column=1, padx=10, pady=10)
78
79     check_out_label = tk.Label(hotel_form, text="Check-out Date")
80     check_out_label.grid(row=3, column=0, padx=10, pady=10, sticky="w")
81
82     check_out_entry = tk.Entry(hotel_form, width=30)
83     check_out_entry.grid(row=3, column=1, padx=10, pady=10)
84
85     # Select Date Buttons
86     select_in_button = tk.Button(hotel_form, text="Select Date", command=lambda: show_calendar(check_in_entry))
87     select_in_button.grid(row=2, column=2, padx=10, pady=10)
88
89     select_out_button = tk.Button(hotel_form, text="Select Date", command=lambda: show_calendar(check_out_entry))
90     select_out_button.grid(row=3, column=2, padx=10, pady=10)
91
92     # Total Price Label
93     total_price_label = tk.Label(hotel_form, text="Total Price: $0")
94     total_price_label.grid(row=4, column=0, padx=10, pady=10)
95
96     # Update Room Options Function
97     def update_room_options(event=None):
98         selected_hotel = hotel_dropdown.get()
99         room_types = list(hotels.get(selected_hotel, {}).keys())
100         room1_dropdown["values"] = room_types
101         room2_dropdown["values"] = room_types
102         room3_dropdown["values"] = room_types
103
104     # Update Price Function
105     def update_price(event=None):
106         selected_hotel = hotel_dropdown.get()
107         price_per_room1 = hotels.get(selected_hotel, {}).get(room1_dropdown.get(), 0)
108         price_per_room2 = hotels.get(selected_hotel, {}).get(room2_dropdown.get(), 0)
109         price_per_room3 = hotels.get(selected_hotel, {}).get(room3_dropdown.get(), 0)
110
111         room1_price_label.config(text=f"${price_per_room1}")
112         room2_price_label.config(text=f"${price_per_room2 if price_per_room2 else 0}")
113         room3_price_label.config(text=f"${price_per_room3 if price_per_room3 else 0}")
114
115         try:
116             num_rooms1 = int(num_rooms1_entry.get())
117         except ValueError:
118             num_rooms1 = 0
119
120         try:
121             num_rooms2 = int(num_rooms2_entry.get())
122         except ValueError:
```


webapp > hotel_booking.py > open_hotel_form

```
50 def open_hotel_form():
112     def update_price(event=None):
129         except ValueError:
130             num_rooms2 = 0
131
132         try:
133             num_rooms3 = int(num_rooms3_entry.get())
134         except ValueError:
135             num_rooms3 = 0
136
137         total_price = (
138             (price_per_room1 * num_rooms1)
139             + (price_per_room2 * num_rooms2)
140             + (price_per_room3 * num_rooms3)
141         )
142         total_price_label.config(text=f"Total Price: ${total_price}")
143
144     hotel_dropdown.bind("<<ComboboxSelected>>", update_room_options)
145     room1_dropdown.bind("<<ComboboxSelected>>", update_price)
146     num_rooms1_entry.bind("<KeyRelease>", update_price)
147     room2_dropdown.bind("<<ComboboxSelected>>", update_price)
148     num_rooms2_entry.bind("<KeyRelease>", update_price)
149     room3_dropdown.bind("<<ComboboxSelected>>", update_price)
150     num_rooms3_entry.bind("<KeyRelease>", update_price)
151
152     def submit_booking():
153         selected_hotel = hotel_dropdown.get()
154         room_type1 = room1_dropdown.get()
155         room_type2 = room2_dropdown.get()
156         room_type3 = room3_dropdown.get()
157         num_rooms1 = num_rooms1_entry.get()
158         num_rooms2 = num_rooms2_entry.get()
159         num_rooms3 = num_rooms3_entry.get()
160         check_in = check_in_entry.get()
161         check_out = check_out_entry.get()
162         total_price = total_price_label.cget("text").split("$")[1]
163
164         if not selected_hotel or not check_in or not check_out:
165             messagebox.showerror("Error", "Please fill in all required fields.")
166             return
167
168         insert_hotel_booking(
169             selected_hotel,
170             room_type1,
```

```

50 def open_hotel_form():
152 def submit_booking():
170     room_type1,
171     room_type2,
172     room_type3,
173     num_rooms1,
174     num_rooms2,
175     num_rooms3,
176     check_in,
177     check_out,
178     total_price,
179 )
180
181 # Display booking details
182 booking_details = (
183     f"Hotel: {selected_hotel}\n"
184     f"Room Type 1: {room_type1} (x{num_rooms1})\n"
185     f"Room Type 2: {room_type2} (x{num_rooms2})\n"
186     f"Room Type 3: {room_type3} (x{num_rooms3})\n"
187     f"Check-in: {check_in}\n"
188     f"Check-out: {check_out}\n"
189     f"Total Price: ${total_price}"
190 )
191 messagebox.showinfo("Booking Details", booking_details)
192 hotel_form.destroy() # Close the form after submission
193
194 tk.Button(hotel_form, text="Book Hotel", command=submit_booking).grid(row=10, column=1, padx=10, pady=10)
195
196 # Start the application
197 if __name__ == "__main__":
198     root = tk.Tk()
199     root.title("Travel Booking System")
200     tk.Button(root, text="Open Hotel Booking Form", command=open_hotel_form).pack(pady=20)
201     root.mainloop()

```

```

# Buttons to open other forms
tk.Button(root, text="Book Hotel", command=open_hotel_form).grid(
    | row=9, column=0, padx=10, pady=10
    | )
tk.Button(root, text="Book Activity", command=open_activity_form).grid(
    | row=10, column=3, padx=10, pady=10
    | )

root.mainloop() # Start the Tkinter main loop

```

➤ Database:

Server: 127.0.0.1 » Database: travel_booking

Structure SQL Search Query Export Import Operations Privileges Routines Events More

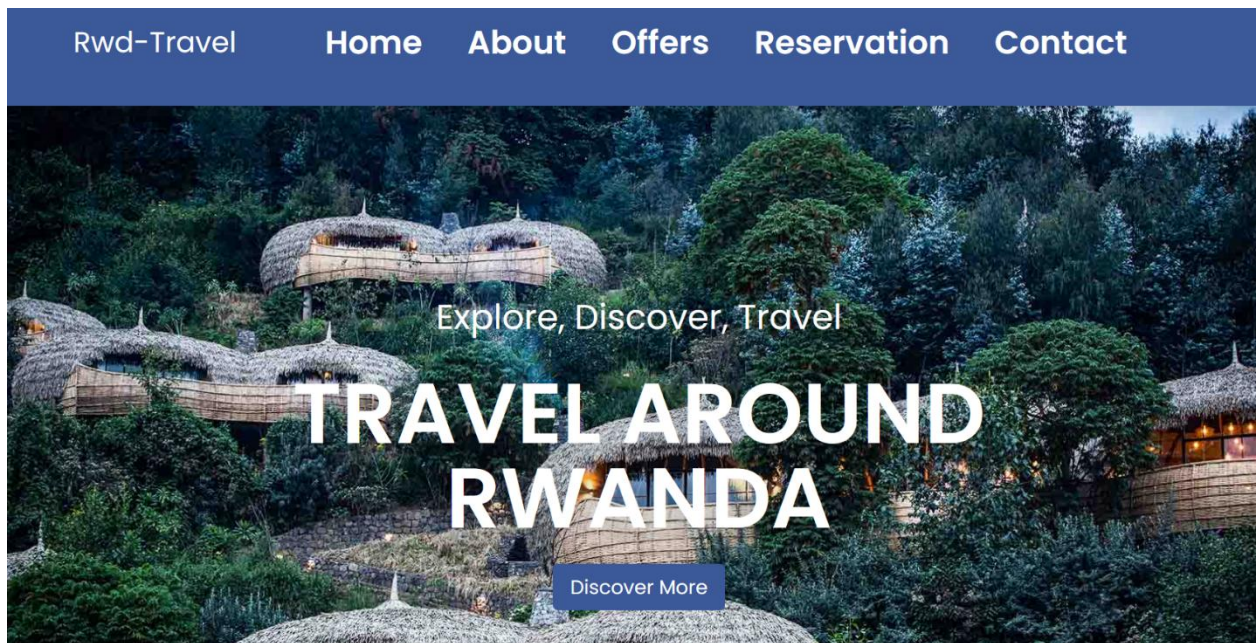
Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> activity_bookings	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KkB	-
<input type="checkbox"/> bookings	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KkB	-
<input type="checkbox"/> hotel_bookings	★ Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	32.0 KkB	-
3 tables	Sum	12	InnoDB	utf8mb4_general_ci	80.0 KkB	0 B

▪ Ui Screenshots:

➤ Home page:



➤ Offers:

Our Current Offers

Black Friday Offer

Enjoy 30% Off On All Destinations. Don't Miss This Limited-Time Offer!

Book Now

Family Packages

Plan A Memorable Family Getaway With Exclusive Discounts On Group Bookings.

Book Now

Express Reservations

Secure Your Spot Quickly With Our Hassle-Free Express Reservation Service.

Book Now

Our Services

Adventure And Wildlife Safaris

• Guided Expeditions To Remote Locations

Cultural Tours

- Guided Visits To Historical Sites And Monuments



Hiking

- Trek Through The Scenic Trails Of The Mountains

➤ Reservation page and forms:

Rwd-Travel

HomeAboutOffersReservationContact

 **Book Your Dream Getaway!** 

🌟🌟🌟 Ready To Plan Your Next Adventure? 🌟🌟🌟

Click The Button Below To Easily Book Hotels, Flights, And Exciting Activities All In One Place! Enjoy A Seamless Reservation Experience Tailored Just For You. Let's Get Started On Your Journey Today!

Open Tkinter GUI

Travel Booking System

Name:

Email:

Destination:

From:

Select Airline:

Travel Date:

Number of People:

Airline Price Per Person: \$0

Activity Booking Form

Select Activities:

☐ Adventure - \$100

☐ Tour Guide - \$150

☐ Hiking - \$80

☐ Camping - \$120

☐ Off-Road - \$200

☐ Water Sports - \$90

☐ Safari - \$100


☐ Cultural Tour - \$30

☐ Beach Visit - \$40

Activity Date:

Number of Participants:

Total Price: \$0

 Hotel Booking Form

Select Hotel:	<div></div>	
Select Room Type 1:	<div></div>	\$0
Number of Room Type 1:	<div></div>	
Select Room Type 2 (Optional):	<div></div>	\$0
Number of Room Type 2:	<div></div>	
Select Room Type 3 (Optional):	<div></div>	\$0
Number of Room Type 3:	<div></div>	
Check-in Date:	<div></div>	<div>Select Date</div>
Check-out Date:	<div></div>	<div>Select Date</div>
Total Price: \$0		
<div>Book Hotel</div>		