TABI3A – Smart Tunisian API for Biodiversity, Intelligence and Awareness

A Backend-Oriented Web Application Project

Eslem Sebri

January 2026

BA/IT Senior Student

TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

Tunis Business School
Ben Arous, TUNISIA.
Academic Year 2025 - 2026

Abstract

Within the context of growing environmental challenges and the increasing importance of sustainable tourism, the preservation and promotion of natural heritage have become essential concerns. Tunisia, despite its rich biodiversity and numerous national parks, lacks a unified digital system capable of organizing and exposing structured information related to its natural ecosystems. In response to this need, the *TABI3A* platform is proposed as a backend-oriented solution dedicated to the management of Tunisian natural parks and eco-tourism data.

TABI3A is implemented as a RESTful API that provides structured access to information about parks, species, geographic locations, and environmentally responsible activities. The platform incorporates secure authentication and role-based access control to ensure that sensitive operations are restricted to authorized users. In addition, the system integrates external weather data to support season-aware and context-sensitive trip recommendations, enhancing the relevance of eco-tourism guidance.

By leveraging modern web technologies and backend development practices, TABI3A demonstrates how digital platforms can contribute to environmental awareness and sustainable tourism initiatives. The project highlights the practical application of web services concepts in a real-world, locally relevant context while providing a flexible foundation for future extensions.

Keywords: Tunisian natural parks, biodiversity, eco-tourism, RESTful API, backend development.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction Content

Tunisia is home to a rich and diverse natural heritage, including national parks, protected ecosystems, endemic species, and unique landscapes that span coastal wetlands, forests, mountains, and desert regions. These natural areas play a crucial role in biodiversity conservation, environmental education, and sustainable tourism. However, access to structured, centralized, and digital information about Tunisian natural parks remains limited, particularly in a format that can be easily reused by applications and digital services.

The project titled **TABI3A – Smart Eco-Tourism and Biodiversity API** aims to address this gap by providing a backend-oriented, web API dedicated to the management and promotion of Tunisian natural parks and their associated fauna and flora. The platform offers structured access to data related to parks, species, locations, and eco-tourism activities, while also integrating external services such as weather data to generate intelligent trip recommendations.

This report presents the design, implementation, and evaluation of the TABI3A backend system, highlighting its technical architecture, functional features, and relevance in promoting sustainable eco-tourism and environmental awareness in Tunisia.

1.2 Project Context

The preservation of natural ecosystems and the promotion of responsible eco-tourism have become global priorities in recent years. In Tunisia, national parks such as Ichkeul, Bou Hedma, and Jebil represent not only ecological treasures but also opportunities for sustainable development through controlled tourism and environmental education. Nevertheless, information related to these parks is often scattered across multiple sources, lacking standardization and digital accessibility.

The TABI3A project is conceived as a backend service that centralizes information about Tunisian natural parks and provides structured access through a RESTful API. By targeting developers, researchers, and digital platforms, TABI3A enables the reuse of environmental data for applications related to tourism, education, and conservation.

In addition to static data management, the platform incorporates dynamic elements such as weather-based trip recommendations, allowing users to receive contextual suggestions adapted to seasons and environmental conditions. Through this approach, TABI3A contributes to the promotion of eco-friendly tourism while encouraging respect for protected natural areas.

1.3 Scope of the Project

The scope of the TABI3A project focuses on the design and implementation of a backend RESTful API with the following key features:

- User authentication and role-based access control using JSON Web Tokens (JWT), supporting different user roles such as visitors, researchers, and administrators [2].
- Management of Tunisian natural parks, including descriptive information, rules, and recommended visiting seasons.
- Management of fauna and flora species associated with national parks, including conservation status and endemicity.
- Geographical localization of parks through dedicated location entities.
- Generation of smart eco-tourism trip recommendations based on park data and external weather information.
- Full API documentation using Swagger (OpenAPI) for ease of testing and integration [5].

The project intentionally excludes frontend interfaces, mobile applications, and geographic information systems (GIS), focusing instead on building a robust, reusable, and well-documented backend service. These excluded aspects are considered potential extensions for future work.

1.4 Research on Similar APIs

In order to design an effective and relevant API, existing platforms and APIs related to nature conservation, tourism, and environmental data were studied. Several notable examples include:

- **National Park Service API (USA)**: This API provides structured access to information about U.S. national parks, including locations, activities, and alerts. While comprehensive, it is limited to a specific geographic context and does not address Tunisian ecosystems.
- **GBIF (Global Biodiversity Information Facility)**: GBIF offers extensive biodiversity datasets from around the world. Despite its richness, it is primarily research-oriented and lacks a simplified interface tailored for eco-tourism and local conservation use cases.

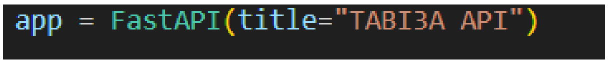
Originality of the Project: Tunisia currently lacks a centralized and structured digital system that aggregates information related to its natural heritage and national parks. In contrast to the APIs discussed above, TABI3A is specifically designed to address the Tunisian context by focusing on national parks, biodiversity, and eco-tourism. The originality of the project lies in its combination of well-structured park and species data with intelligent trip recommendations influenced by seasonal conditions and real-time weather information. By adopting a lightweight, backend-only architecture, the platform offers a flexible and extensible foundation for future applications while actively promoting sustainable tourism and environmental awareness within Tunisia.

Chapter 2

Technologies and Features

This chapter presents the technologies and tools used in the development of the TABI3A backend system. Each technology was selected to ensure scalability, security, maintainability, and ease of deployment. The combination of modern backend frameworks, database management tools, API documentation solutions, and containerization techniques contributes to the robustness and professionalism of the project.

2.1 FastAPI Contribution



```
app = FastAPI(title="TABI3A API")
```

Figure 2.1: FastAPI framework used as the core backend engine of the TABI3A API

FastAPI serves as the core framework of the TABI3A backend application [1]. It is a modern, high-performance web framework designed for building RESTful APIs using Python. FastAPI was chosen for its speed, automatic data validation, and native support for asynchronous operations.

In the TABI3A project, FastAPI is responsible for handling HTTP requests, routing API endpoints, validating input data through schemas, and generating structured responses. Its strong integration with Python type hints enables early detection of errors and improves code reliability. Additionally, FastAPI automatically generates interactive API documentation through Swagger, which significantly simplifies testing and external integration.

The modular architecture supported by FastAPI allows the application to be organized into logical components such as authentication, parks, species, users, and recommendations, ensuring a clean and maintainable codebase.

2.2 JWT Authentication and Security

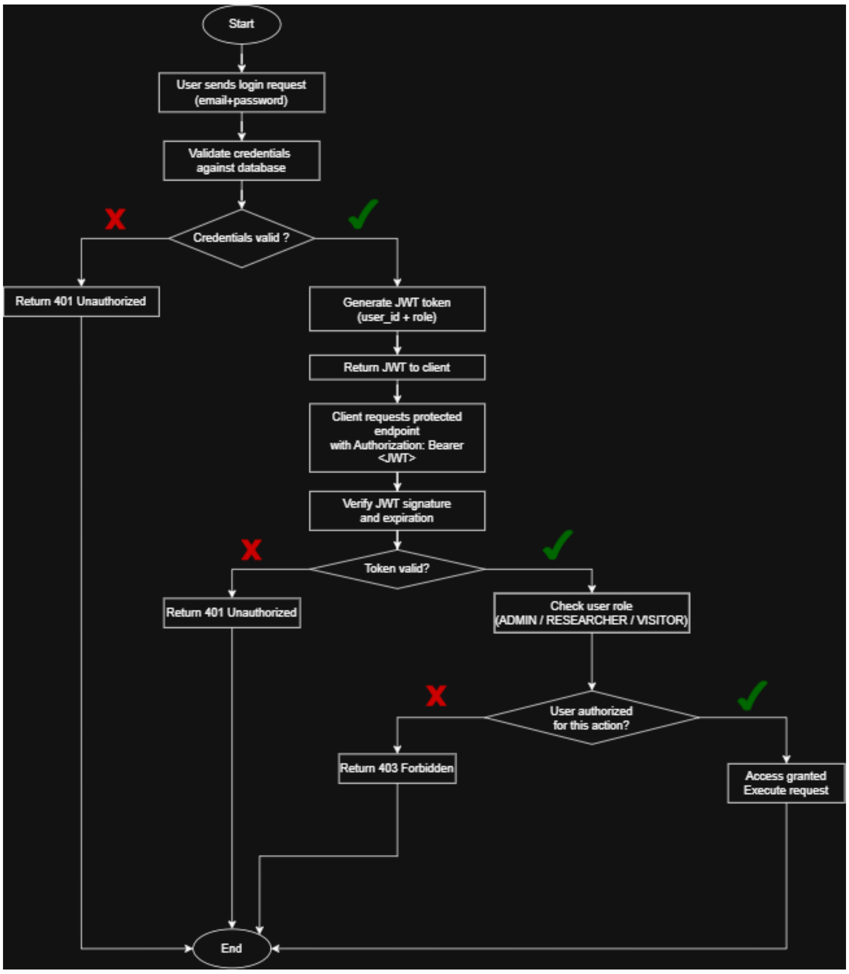

Start
User sends login request (email+password)
Validate credentials against database
Credentials valid ?
Return 401 Unauthorized
Generate JWT token (user_id + role)
Return JWT to client
Client requests protected endpoint with Authorization: Bearer <JWT>
Verify JWT signature and expiration
Token valid?
Return 401 Unauthorized
Check user role (ADMIN / RESEARCHER / VISITOR)
User authorized for this action?
Return 403 Forbidden
Access granted Execute request
End

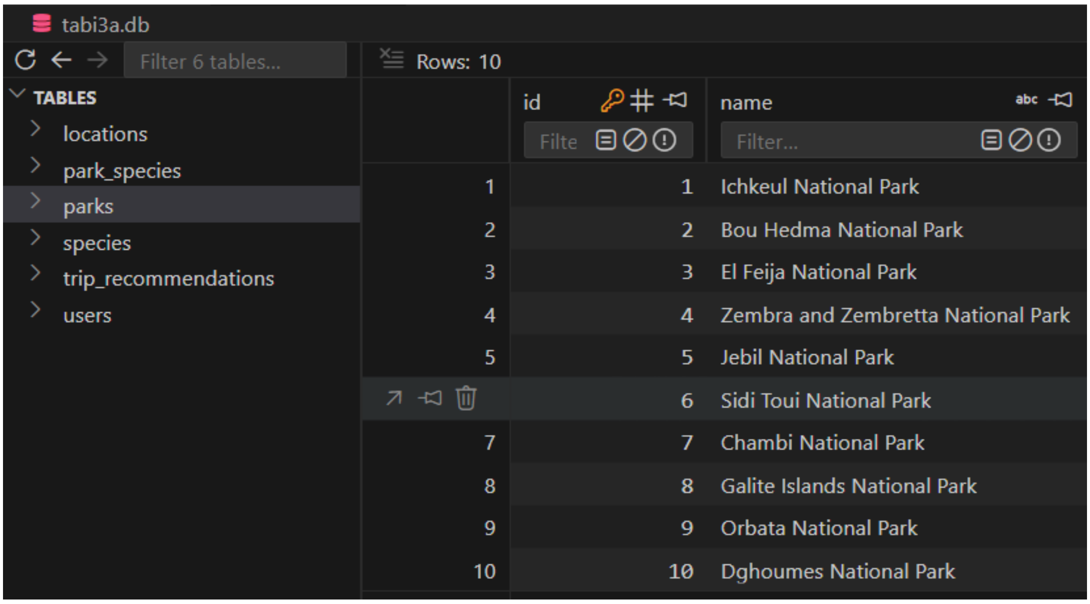
Figure 2.2: JWT-based authentication and role-based authorization flow in the TABI3A platform *(draw.io)* [11]

Security is a critical aspect of the TABI3A platform, particularly due to the presence of role-based access control. JSON Web Tokens (JWT) are used to implement secure authentication and authorization mechanisms [2].

JWT authentication enables users to log in and receive a signed access token, which must be included in subsequent requests to protected endpoints. The platform supports multiple user roles, including visitors, researchers, and administrators. Each role has different access privileges, ensuring that sensitive operations such as creating parks, managing species, or modifying user roles are restricted to authorized users.

By adopting a stateless authentication model, JWT improves scalability and reduces server-side session management overhead, making the system suitable for future expansion.

2.3 SQLite and SQLAlchemy Contribution



	id	name
1	1	Ichkeul National Park
2	2	Bou Hedma National Park
3	3	El Feija National Park
4	4	Zembra and Zembretta National Park
5	5	Jebil National Park
	6	Sidi Toui National Park
7	7	Chambi National Park
8	8	Galite Islands National Park
9	9	Orbata National Park
10	10	Dghoumes National Park

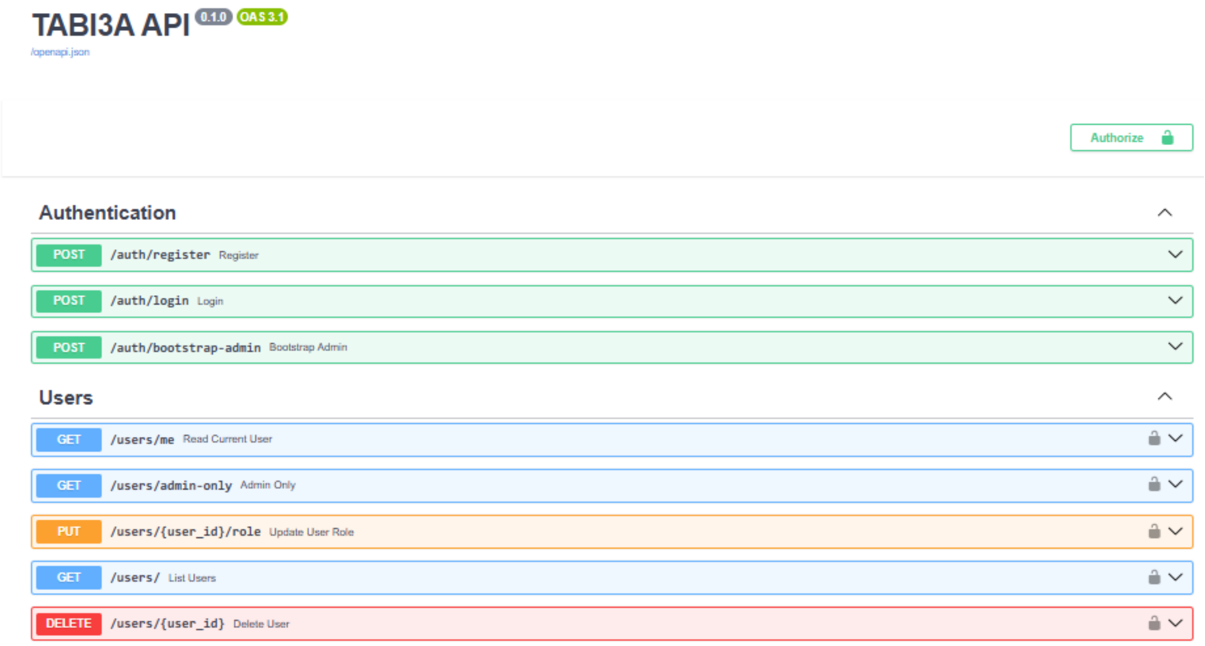
Figure 2.3: SQLite database tables storing parks, species, and trip recommendations

SQLite is used as the database solution for the TABI3A project due to its lightweight nature and ease of integration [4]. It is particularly well-suited for academic projects and backend prototypes, as it does not require a separate database server while still offering reliable relational data storage.

SQLAlchemy is employed as the Object-Relational Mapping (ORM) tool, enabling seamless interaction between the database and the application logic [3]. Through SQLAlchemy, database tables are mapped to Python classes, simplifying CRUD operations and enforcing data consistency.

The database schema includes entities such as users, parks, species, locations, and trip recommendations, along with appropriate relationships such as one-to-one and many-to-many associations. This structured approach ensures data integrity while allowing flexible querying and future extensibility.

2.4 Swagger API Documentation



TABI3A API 0.1.0 OAS 3.1

/openapi.json

Authorize

Authentication

POST /auth/register Register

POST /auth/login Login

POST /auth/bootstrap-admin Bootstrap Admin

Users

GET /users/me Read Current User

GET /users/admin-only Admin Only

PUT /users/{user_id}/role Update User Role

GET /users/ List Users

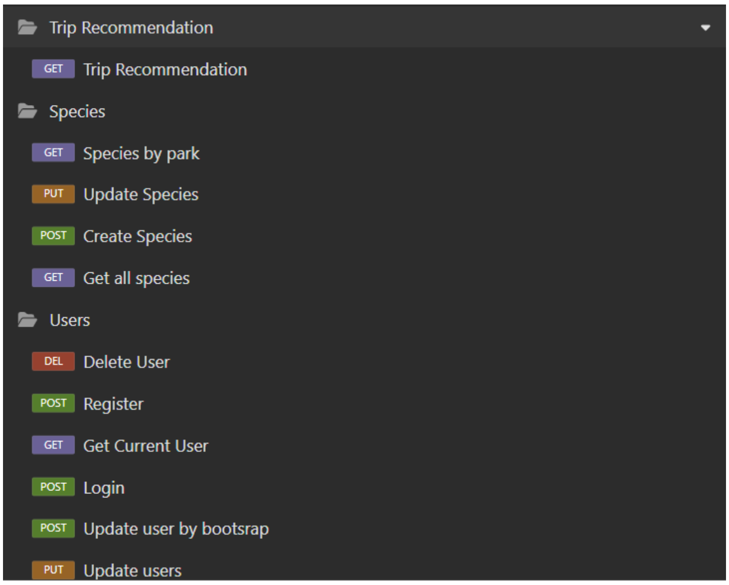
DELETE /users/{user_id} Delete User

Figure 2.4: Swagger (OpenAPI) interactive documentation interface for the TABI3A API

Swagger, integrated through FastAPI's OpenAPI support, plays a central role in documenting the TABI3A API [5]. It provides an interactive web interface that allows developers and testers to explore available endpoints, view request and response schemas, and execute API calls directly from the browser.

The Swagger interface significantly enhances the usability of the API by offering clear documentation without requiring additional manual effort. It also serves as a validation tool, ensuring that endpoints behave as expected and comply with their defined contracts.

2.5 Insomnia for API Testing



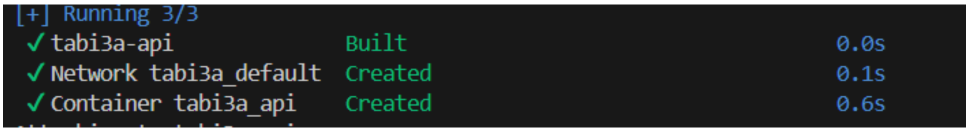
Trip Recommendation
GET Trip Recommendation
Species
GET Species by park
PUT Update Species
POST Create Species
GET Get all species
Users
DEL Delete User
POST Register
GET Get Current User
POST Login
POST Update user by bootsrap
PUT Update users

Figure 2.5: API request testing and response validation using Insomnia

Insomnia is used extensively during the development and testing phases of the TABI3A project [6]. It provides a user-friendly environment for sending HTTP requests, managing authentication tokens, and validating API responses.

Using Insomnia, all major API functionalities were tested, including user authentication, role-based access control, park and species management, and trip recommendation retrieval. The tool facilitated rapid debugging and verification of request flows, contributing to the stability and correctness of the backend system.

2.6 Docker Contribution



```
[+] Running 3/3
 ✓ tabi3a-api               Built                                    0.0s
 ✓ Network tabi3a_default   Created                                  0.1s
 ✓ Container tabi3a_api     Created                                  0.6s
```

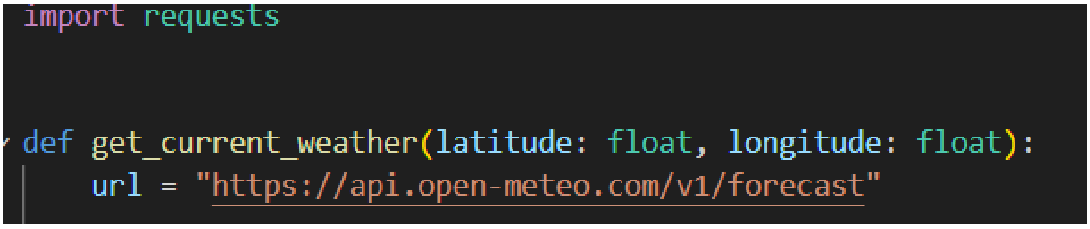
Figure 2.6: Docker containerization and execution of the TABI3A backend application

Docker is utilized to containerize the TABI3A backend application, ensuring consistent execution across different environments [7]. By packaging the application and its dependencies into a container, Docker eliminates issues related to system configuration and dependency mismatches.

The project includes a Dockerfile and a Docker Compose configuration, allowing the entire backend to be built and executed using a single command. This approach simplifies deployment, improves reproducibility, and aligns with modern software engineering practices.

Containerization also prepares the project for potential future deployment on cloud platforms or integration within larger systems.

2.7 External Weather API Integration



```
import requests


def get_current_weather(latitude: float, longitude: float):
    url = "https://api.open-meteo.com/v1/forecast"
```

Figure 2.7: Integration of an external weather API for generating eco-tourism trip recommendations

To enhance the eco-tourism aspect of the platform, TABI3A integrates an external weather API [9]. This integration enables the application to retrieve real-time weather data based on the geographic location of national parks.

Weather information is used as an input for generating smart trip recommendations, allowing users to receive suggestions that take environmental conditions into account. This feature adds dynamic behavior to the system and demonstrates how external APIs can enrich backend services.

The integration highlights the practical use of third-party APIs in building intelligent and context-aware applications.

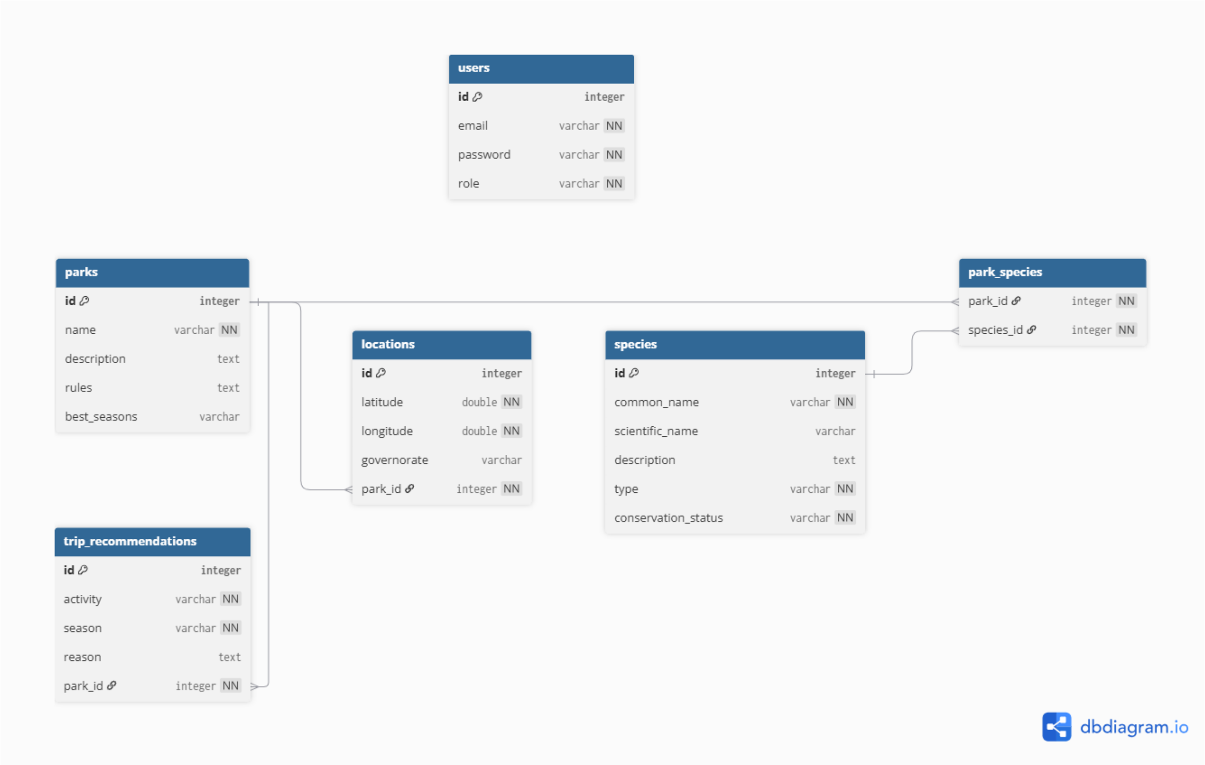
2.8 HTTP Methods Overview

The TABI3A API follows RESTful design principles and makes use of standard HTTP methods to ensure clarity and consistency:

- **GET:** Used to retrieve information about parks, species, users, and trip recommendations.
- **POST:** Used to create new resources such as user accounts, parks, species, and locations.
- **PUT:** Used to update existing resources, including park information, species attributes, and user roles.
- **DELETE:** Used to remove resources such as parks or user accounts when authorized.

This standardized use of HTTP methods ensures predictable behavior and simplifies client-side integration.

2.9 Database Structure



users		
id	integer	
email	varchar	NN
password	varchar	NN
role	varchar	NN

parks		
id	integer	
name	varchar	NN
description	text	
rules	text	
best_seasons	varchar	

locations		
id	integer	
latitude	double	NN
longitude	double	NN
governorate	varchar	
park_id	integer	NN

species		
id	integer	
common_name	varchar	NN
scientific_name	varchar	
description	text	
type	varchar	NN
conservation_status	varchar	NN

park_species		
park_id	integer	NN
species_id	integer	NN

trip_recommendations		
id	integer	
activity	varchar	NN
season	varchar	NN
reason	text	
park_id	integer	NN

Figure 2.8: Database schema and entity relationships of the TABI3A system *(dbdiagram.io)* [10]

The database structure of the TABI3A platform is designed to model the main components involved in managing Tunisian natural parks and eco-tourism data. The schema is organized around the following key entities and relationships:

- **Parks:** Represents Tunisian national parks and serves as the central entity of the system, storing descriptive information and seasonal characteristics.
- **Locations:** Linked to parks through a one-to-one relationship, ensuring that each park is associated with precise geographic coordinates and administrative information.
- **Species:** Stores fauna and flora data, including scientific classification and conservation status.
- **Park–Species Association:** Implemented as a many-to-many relationship, allowing multiple species to exist in multiple parks while maintaining data normalization.
- **Trip Recommendations:** Associates eco-tourism activities with specific parks and seasons, supporting the generation of context-aware and environmentally responsible visit suggestions.

This structured schema reflects real-world relationships, ensures data consistency, and provides a scalable foundation for future extensions of the platform.

Chapter 3

Conclusion

This project, titled *TABI3A – A Backend Platform for Tunisian Natural Parks*, presents a practical and innovative solution for organizing and promoting Tunisia's natural heritage. By centralizing information about national parks, biodiversity, and eco-tourism activities, the platform addresses the lack of a unified digital system dedicated to environmental data management. Leveraging modern backend technologies such as FastAPI, SQLite, JWT authentication, RESTful APIs, and Docker, the project demonstrates how digital tools can support sustainable tourism and environmental awareness.

Working on this project has been an enriching and rewarding experience. Despite technical challenges encountered during development, the process allowed me to strengthen my backend development skills, deepen my understanding of API design, and apply security and architectural concepts to a real-world problem. The successful implementation of TABI3A highlights the importance of innovation, adaptability, and perseverance in software engineering projects.

I am sincerely grateful to my professor for his guidance and support throughout this work. His feedback and encouragement played a significant role in shaping the direction and overall quality of the project.

Eslem Sebri

TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

Bibliography

[1] FastAPI Documentation. `https://fastapi.tiangolo.com/`

[2] JSON Web Token (JWT). `https://en.wikipedia.org/wiki/JSON_Web_Token`

[3] SQLAlchemy ORM. `https://www.sqlalchemy.org/`

[4] SQLite Database Engine. `https://www.sqlite.org/index.html`

[5] Swagger (OpenAPI Specification). `https://swagger.io/`

[6] Insomnia REST Client. `https://insomnia.rest/`

[7] Docker Platform. `https://www.docker.com/`

[8] Git Version Control System. `https://git-scm.com/`

[9] Open-Meteo Weather API. `https://open-meteo.com/`

[10] dbdiagram.io – Database Diagram Tool. `https://dbdiagram.io/`

[11] draw.io – Online Diagram Tool. `https://www.draw.io/`

[12] Python Programming Language. `https://www.python.org/`

Appendix A

This appendix presents representative examples of API requests and responses supported by the TABI3A platform. The purpose of this section is to illustrate the behavior of the system and validate its functionality through practical use cases.

Authentication Requests

POST /auth/login

Response Example:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "bearer"
}
```

User Management Requests

GET /users/me

Response Example:

```
{
  "email": "admin@tabi3a.tn",
  "role": "ADMIN"
}
```

PUT /users/user_id/role

Response Example:

```
{
  "message": "Role updated",
  "role": "RESEARCHER"
}
```

Park Management Requests

POST /parks

Response Example:

```
{
  "id": 3,
  "name": "El Feija National Park",
  "description": "Mountain forest park known for Barbary deer",
  "rules": "No fires, respect wildlife",
  "best_seasons": "Spring, Summer"
}
```

POST /parks/park_id/location

Response Example:

```
{
  "park": "El Feija National Park",
  "latitude": "36.495",
  "longitude": "8.708",
  "governorate": "Jendouba"
}
```

Species Management Requests

POST /species

Response Example:

```
{
  "id": 1,
  "common_name": "Barbary Deer",
  "scientific_name": "Cervus elaphus barbarus",
  "category": "FAUNA",
  "conservation_status": "Critically Endangered",
  "endemic": true
}
```

GET /species

Response Example:

```
[
  {
    "id": 1,
    "common_name": "Barbary Deer",
    "category": "FAUNA"
  },
  {
    "id": 2,
    "common_name": "Aleppo Pine",
    "category": "FLORA"
  }
]
```

Trip Recommendation Requests

GET /recommendations/park/park_id

Response Example:

```
{
  "park": "Ichkeul National Park",
  "season": "Spring",
  "activity": "Birdwatching",
  "reason": "Favorable weather conditions and migratory bird presence"
}
```