Technical Report: Virtual Tour API for Ennejma Ezzahra Palace

Dhia Eddine Zouari
Senior BA-IT Student
Tunis Business School

January 21, 2025

Abstract

This technical report provides an in-depth analysis of the **Virtual Tour API**, a project designed to offer a **360° interactive virtual tour** of the **Ennejma Ezzahra Palace** in Tunisia. The system is built using **Django REST Framework**, integrating **GoMaps.Pro API** for immersive virtual tours. Security is ensured via **JWT-based authentication**, and **Docker** is used for **scalability and deployment**. The report details **system architecture, API structure, security mechanisms, testing strategy, and deployment**.

Contents

1 Context and Scope Analysis

1.1 Introduction

With the increasing demand for **digital tourism**, historical landmarks are shifting towards **virtual experiences**. This project aims to **digitally preserve and promote** the **Ennejma Ezzahra Palace**, offering an immersive **360° virtual tour**.

1.2 Overview of Ennejma Ezzahra Palace

Ennejma Ezzahra Palace is a **historical monument, museum, and a cultural venue for classical music events**. Built in the early 20th century, it serves as a center for **heritage preservation** and hosts **annual music festivals**.

1.3 Project Originality and Scope

Why is this project unique?

- **Uses immersive 360° tours** powered by GoMaps.Pro API.
- **Implements secure user authentication**.
- **Allows interactive feedback submission**.
- **Provides a scalable architecture with Docker deployment**.

2 System Design and Architecture

2.1 Project Structure

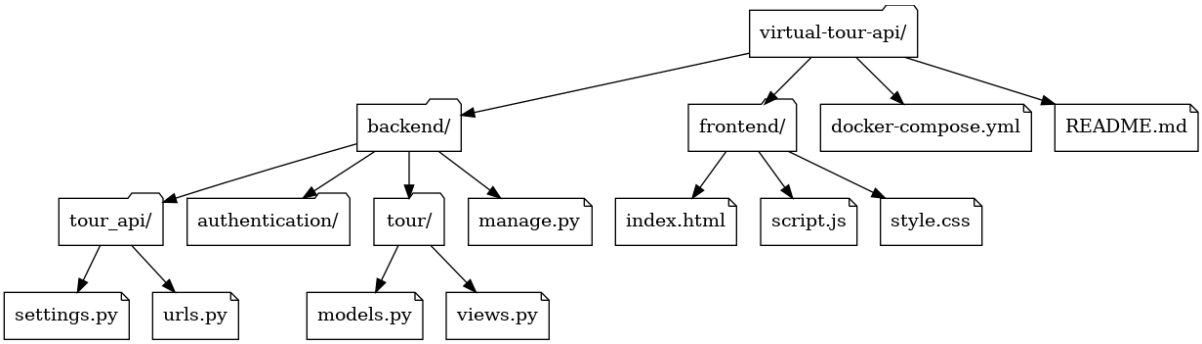

Figure 1: Project Structure Overview

2.2 Database Schema

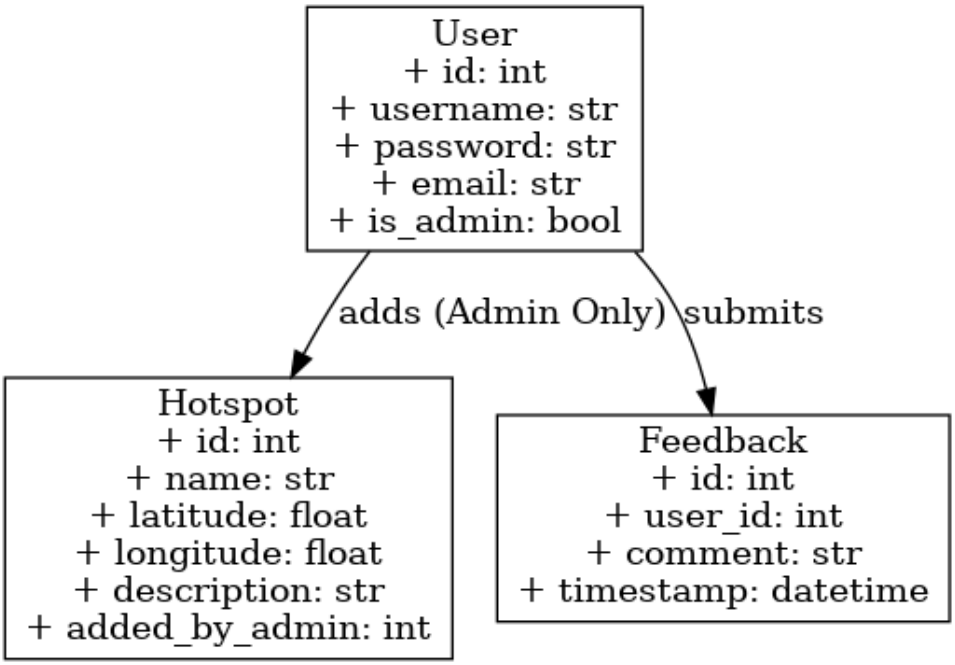

Figure 2: Database Schema and Relationships

3 API Design

3.1 Tour API - Fetching Virtual Hotspots

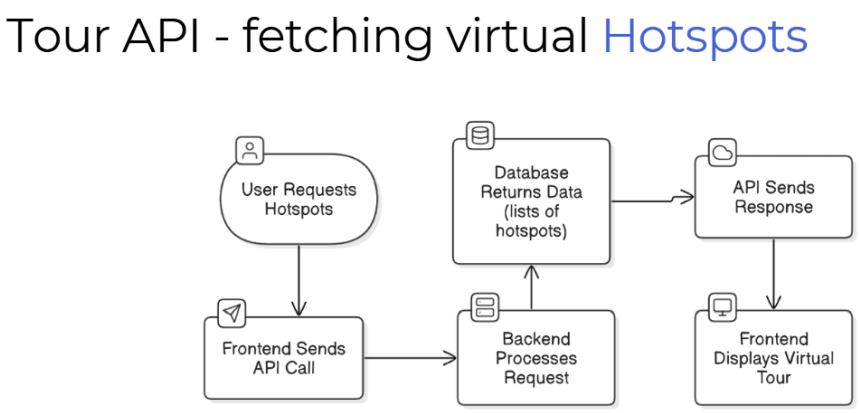

Figure 3: Tour API Flow

3.2 Feedback API - User Engagement

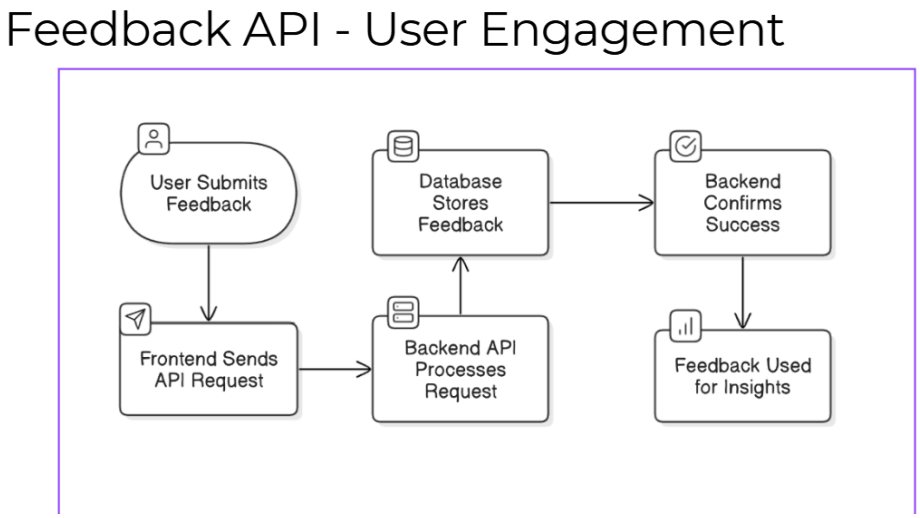

Figure 4: Feedback API Flow

4 Security Implementation

4.1 JWT Authentication Flow

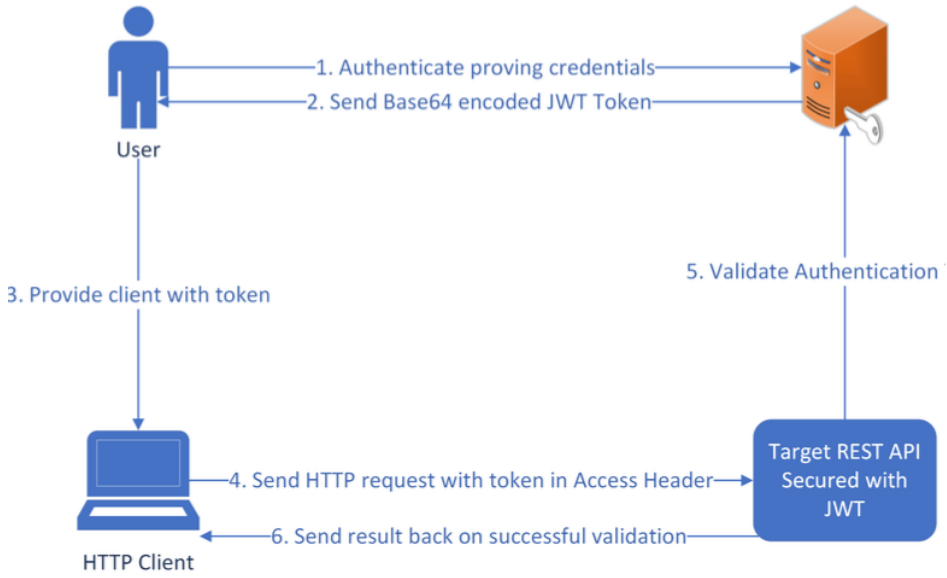

Figure 5: JWT Authentication Workflow

4.2 Advanced Security Measures

- **Password Hashing** - Protects user credentials from exposure.
- **Token Expiry Refresh Mechanism** - Controls user sessions.
- **CORS Policy** - Prevents unauthorized API access.
- **Rate Limiting** - Protects against brute-force attacks.
- **Input Validation** - Prevents SQL Injection and XSS.

5 Deployment Strategy

5.1 Docker-Based Deployment

Why Docker?

- Ensures **consistent environments** across dev and production.
- Facilitates **scalability and CI/CD integration**.

5.2 Docker Compose Configuration

```
version: '3.8'
services:
  backend:
    build: .
    ports:
      - "8000:8000"
    depends_on:
      - db
    environment:
      - DATABASE_URL=sqlite:///db.sqlite3

  db:
    image: postgres
    environment:
      - POSTGRES_DB=virtualtour
      - POSTGRES_USER=admin
      - POSTGRES_PASSWORD=adminpassword
```

6 Testing Strategy

6.1 Testing with Insomnia and Swagger

- **Insomnia** - Used for manual API testing and response validation.
- **Swagger** - Used for auto-generated API documentation and live testing.

7 Conclusion and Future Enhancements

7.1 Future Enhancements

- **AI-Powered Virtual Tour Guide** for enhanced interaction.
- **Multilingual support** to reach a broader audience.
- **Augmented Reality (AR) integration** for real-world engagement.

8 References

- GoMaps.Pro API - https://www.gomaps.pro/
- Django REST Framework - https://www.django-rest-framework.org/
- Docker Documentation - https://docs.docker.com/
- JWT Authentication - https://jwt.io/
- Insomnia API Testing - https://insomnia.rest/
- Swagger API Docs - https://swagger.io/