PROJECT REPORT

YatraSathi -Tourist Guide for Chaardham pilgrims

Subject Name: Project Based Learning in Java

Subject Code : 23CSH-304

Submitted to:

Er. Deep Prakash Gupta

(E18557)

Submitted by:

Name: Akshara Chauhan

UID: 23BCS11410

Section: KRG_2B

Abstract:

YatraSathi is a travel companion platform designed to simplify trip planning and enhance the travel experience for users.
The backend, developed using **Spring Boot**, provides secure APIs for **user authentication**, **travel data retrieval** via the **Amadeus API**, and **media management** through **Cloudinary** integration.
This backend acts as the central service layer connecting the Flutter-based frontend with external services and databases, ensuring a seamless, secure, and efficient user experience..

Objectives:

The main objectives of this project are:

1. To design and develop a **secure RESTful backend** using Spring Boot.
2. To use **MongoDB** as the primary NoSQL database for efficient and scalable data storage.
3. To implement **JWT-based authentication** for secure access control.
4. To integrate **Amadeus API** for retrieving travel data such as destinations and flights.
5. To manage user-uploaded travel photos using **Cloudinary API**.
6. To provide REST endpoints for a **Flutter-based frontend application**.

Problem Definition

Travelers often face challenges in finding accurate travel information, managing plans, and storing memories in one place.
Existing apps may lack a unified and efficient backend that connects travel data, user information, and media storage securely.
YatraSathi addresses this by providing a single backend solution combining **authentication**, **travel APIs**, and **media management** with a modern and scalable technology stack.

System Requirements

Hardware Requirements

- Processor: Intel i5 or higher
- RAM: Minimum 8 GB
- Hard Disk: Minimum 500 GB
- Internet Connection: Required for API access

Software Requirements

- Operating System: Windows / Linux / macOS
- Backend Framework: Spring Boot 3.x
- Database: MongoDB (NoSQL)
- Programming Language: Java 17+
- Build Tool: Maven
- External APIs: Cloudinary API, Amadeus API
- IDE: IntelliJ IDEA / Eclipse / VS Code

System Design

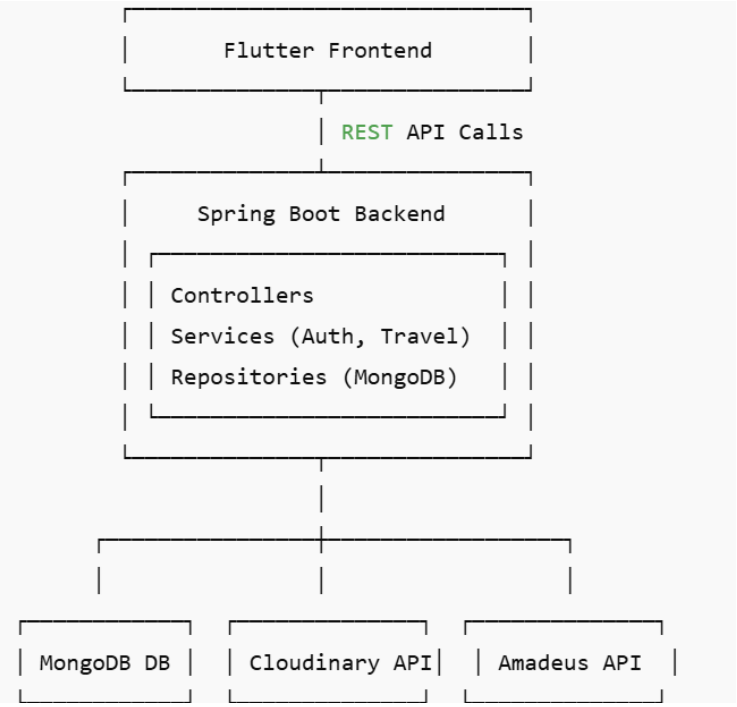
I. Architecture Overview

YatraSathi follows a **three-tier architecture**:
Presentation Layer: Flutter frontend
Application Layer: Spring Boot RESTful backend
Data Layer: MongoDB database.

- External Services:

Amadeus API for travel information
Cloudinary API for image storage

System Architecture Diagram



Modules

Module Description

1. Authentication Module

- Allows user registration and login.
- Implements JWT-based authentication for token validation.
- Passwords are encrypted using BCrypt.
- Endpoints: /api/auth/register, /api/auth/login.

2. Travel Module

- Connects to Amadeus API to fetch travel details.
- Provides endpoints like /api/travel/search?destination=Paris.
- Uses service layer for API integration and JSON data parsing.

3. Gallery Module

- Manages user travel photos using Cloudinary.
- Stores image URL and metadata in MongoDB.
- Endpoints: /api/gallery/upload, /api/gallery/all, /api/gallery/{id}.

4. Security Module

- Implemented using Spring Security and JWT.
- Protects all API routes except authentication endpoints.
- Uses a custom SecurityConfig and JwtUtil for token management**.**

5. Database Module (MongoDB)

- Stores user and gallery documents.
- Collections:
    - users → user details and authentication info.
    - gallery → image data (URLs, captions, upload date).
- No need for predefined schema; flexible document structure allows scalability.

Results and Output

The backend successfully performs:

✅ User registration and authentication using JWT
✅ Secure token validation on protected endpoints
✅ Travel data retrieval from Amadeus API
✅ Image upload and retrieval through Cloudinary
✅ Storage and retrieval of documents from MongoDB

Example API Endpoints

Method Endpoint Description

POST /api/auth/register Register a new user

POST /api/auth/login Login and receive JWT token

GET /api/travel/search?destination=Paris Fetch travel data

POST /api/gallery/upload Upload image to Cloudinary

GET /api/gallery/all Retrieve all gallery images

Conclusion

The **YatraSathi Spring Boot Backend with MongoDB** demonstrates the development of a modern, scalable, and secure backend system for a travel application.
By integrating **MongoDB**, **Spring Security**, **Cloudinary**, and **Amadeus APIs**, it provides a comprehensive backend capable of handling real-world travel application demands.
This project successfully meets its objectives, showcasing proficiency in backend development, API integration, and cloud-based data handling using Java.