Address Book API – Approach Document

Title

1. Overview

This document explains the overall approach, design decisions, and implementation strategy used to build the Address Book API using FastAPI and SQLite.

2. Technology Stack

The application is built using Python 3 and FastAPI for building RESTful APIs. SQLite is used as the database for simplicity and ease of setup. SQLAlchemy is used as the ORM, and Pydantic is used for request and response validation.

3. Application Architecture

The project follows a modular structure separating concerns into different files. Database configuration, models, schemas, CRUD logic, utility functions, and logging are all kept separate to improve readability, maintainability, and testability.

4. Database Design

The database consists of a single table named 'addresses' with fields for id, name, latitude, and longitude. SQLite was chosen because it is lightweight and sufficient for a minimal API demonstration.

5. API Design

RESTful principles are followed. Endpoints are provided to create, update, delete, and retrieve addresses. An additional endpoint allows querying addresses within a specified distance from a given latitude and longitude.

6. Validation

All incoming request data is validated using Pydantic schemas. Latitude and longitude values are constrained to valid geographic ranges to ensure data integrity.

7. Distance Calculation

The Haversine formula is used to calculate the distance between two geographic points. This logic is isolated in a utility module to keep the code clean and reusable.

8. Logging

Basic logging is implemented to track important application events such as address creation and updates. This helps in debugging and monitoring application behavior.

9. Documentation & Testing

FastAPI’s built-in Swagger UI is used for API documentation and manual testing. The API can also be tested using tools like curl, Postman, or the Python requests library.

10. Conclusion

This project demonstrates best practices in building a minimal yet scalable FastAPI application with proper structure, validation, and documentation.