Backend documentation

Jarpula Bhanu Prasad
AI21BTECH11015

Kola Akshitha
AI21BTECH11017

1 Overview

The backend system is built using the `FastAPI` framework. It supports features such as user registration, login, voice-based authentication, and anti-spoofing detection. Additionally, it includes functionalities for audio encryption and feature extraction using TenSEAL and Librosa.

2 Core Modules

The backend is organized into several core modules:

- `main.py`: The main entry point for the FastAPI application.
- `database.py`: Contains database connection logic.
- `models.py`: Defines the database schema using SQLAlchemy.
- `crud.py`: Implements database operations.
- `encryption.py`: Handles voice data encryption and feature extraction.
- `antispoof.py`: Uses the anti-spoofing model for detecting fake audio.

3 API Endpoints

3.1 User Registration

- **Endpoint:** `/api/register`
- **Method:** `POST`
- **Description:** Registers a new user with username, email, password, and voice data.
- **Input Parameters:**

  - `username`: User's name.
  - `email`: User's email.
  - `password`: User's password.
  - `audio`: Voice data..

- **Error Handling:**
  - Throws `HTTP 409` if the user already exists.
  - Throws `HTTP 400` if audio processing or anti-spoofing detection fails.

3.2 User Login

- **Endpoint:** `/api/login`
- **Method:** `POST`
- **Description:** Authenticates a user using email and password.
- **Input Parameters:**
  - `email`: User's email.
  - `password`: User's password.
- **Error Handling:**
  - Throws `HTTP 401` if authentication fails.

3.3 Voice-Based Login

- **Endpoint:** `/api/login/voice`
- **Method:** `POST`
- **Description:** Authenticates a user based on voice data.
- **Input Parameters:**
  - `voice_file`: Audio file for voice-based login.

4 Encryption

The backend uses `TenSEAL` for encryption. Voice data is processed as follows:

- Extract features (MFCC or Log-Mel Spectrogram).
- Encrypt the feature vector using TenSEAL's CKKS scheme.
- Store encrypted data in the database.

5 Anti-Spoofing Detection

The anti-spoofing module uses a pre-trained TensorFlow model to classify audio as `REAL` or `FAKE`.

- Audio is preprocessed using Librosa to extract MFCC features.
- The model outputs a label (`REAL` or `FAKE`) with a confidence score.

6 Database Configuration

6.1 Connection

The database uses PostgreSQL. The connection details are loaded from environment variables using the `dotenv` package.

6.2 Models

The `User` model includes the following fields:

- `username`: String
- `email`: String
- `password_hash`: Encrypted password
- `voice_data`: Encrypted voice features

7 Logging

The application logs authentication activities using the `python-json-logger` package. Logs are stored in `auth_activity.log`.