**Face Recognition Attendance System Documentation**

**Table of Contents**

* Introduction
* 1.1 Purpose
* 1.2 Technologies Used
* 1.3 Project Overview
* System Architecture
* 2.1 Frontend (ReactJS and TailwindCSS)
* 2.2 Backend (FastAPI)
* 2.3 Face Recognition Library (Insightface)
* Project Execution
* 3.1 User Input
* 3.2 Backend Processing
* 3.3 Results Storage
* Data Structures and Algorithms
* 4.1 Indexing for Efficient Retrieval
* 4.2 Face Recognition Algorithm in Insightface
* Results and Storage
* 5.1 JSON Storage
* 5.2 CSV Storage
* Frontend Functionality
* 6.1 Attendance Retrieval
* 6.2 Aggregate Attendance Display
* Conclusion
* 7.1 Achievements
* 7.2 Future Improvements

**1. Introduction**

**1.1 Purpose**

This documentation serves as a comprehensive guide to the Face Recognition Attendance System developed as an assignment for the Algorithms and Data Structures course. The project employs modern technologies and focuses on the practical application of algorithms and data structures.

**1.2 Technologies Used**

* **Backend:** FastAPI
* **Face Recognition Library:** Insightface
* **Frontend:** ReactJS
* **Frontend Design:** TailwindCSS

**1.3 Project Overview**

The system captures attendance through face recognition, using a group image uploaded by the user. It then processes and stores the results in JSON and CSV formats.

**2. System Architecture**

**2.1 Frontend (ReactJS and TailwindCSS)**

The ReactJS frontend provides a user-friendly interface for uploading group images, specifying lecture details, and retrieving attendance information. TailwindCSS enhances the design aesthetics, creating an intuitive user experience.

**2.2 Backend (FastAPI)**

FastAPI serves as the backend framework, handling data processing and communication between the frontend and the face recognition library (Insightface). It also manages the storage of attendance results.

**2.3 Face Recognition Library (Insightface)**

Insightface, an open-source image detection and recognition library, is integrated into the system to perform face detection and recognition against individual student images stored in the server directory.

**3. Project Execution**

**3.1 User Input**

The process begins with the user uploading a group image, specifying the date of the lecture and course through a frontend form. This information is then sent to the FastAPI backend for further processing.

**3.2 Backend Processing**

The backend utilizes Insightface to detect and recognize faces in the uploaded image. The recognition process involves matching faces with individual student images stored on the server.

**3.3 Results Storage**

The results, including roll numbers and attendance status, are stored in JSON format on the local file system. Additionally, a CSV file is stored in the frontend for easy retrieval and analysis.

**4. Data Structures Used**

**4.1 Indexing for Efficient Retrieval**

To optimize attendance retrieval, the system employs indexing and mapping, enabling quick access to attendance details for a particular date.

**4.2 Arrays for Client-Server Interaction**

To reciprocate data between client and server on API endpoints, they system uses arrays wrapped in JSON file structure.

**4.3 Face Recognition Algorithm in Insightface**

* **Hash Tables:** Insightface uses them for storing and accessing pre-computed features or for mapping facial features to unique identifiers.
* **Queues and Stacks:** Queues and stacks could be used in various stages of the recognition pipeline, managing the order of processing or storing temporary data.
* **Matrices:** They represent pixel values and be used in various mathematical operations during feature extraction and comparison.
* **Sets:** Sets might be used to store unique facial features or descriptors, facilitating efficient comparisons and eliminating duplicates.

**5. Results and Storage**

**5.1 JSON Storage**

Attendance results are stored in JSON format on the local file system, using roll numbers as keys and attendance status (0-1 boolean) as values.

**5.2 CSV Storage**

In addition to JSON, a CSV file is stored in the frontend, facilitating easy data analysis and visualization.

**6. Frontend Functionality**

**6.1 Attendance Retrieval**

The frontend provides functionality to fetch attendance details for a particular date. The indexed format ensures quick and efficient retrieval.

**6.2 Aggregate Attendance Display**

A dedicated route in the frontend displays aggregate attendance in a visually appealing format, such as a table with partially filled horizontal bars next to the roll number and name columns.

**7. Conclusion**

**7.1 Achievements**

The Face Recognition Attendance System successfully integrates cutting-edge technologies and algorithms to automate the attendance tracking process, providing a reliable and efficient solution.

**7.2 Future Improvements**

Future improvements may include database integration for scalable data storage and further enhancements to the user interface for an even more intuitive experience.