Overview

The Facial Recognition Attendance System is a Python-Flask-based application for managing student attendance via facial recognition.

It integrates AWS Rekognition for facial detection, Firebase Firestore for database management, and features a responsive front-end

with dynamic updates. The system supports multi-role user management, enabling admins, teachers, and students to interact effectively.

System Architecture

- **Frontend**: HTML templates styled with Bootstrap, supporting dynamic updates with AJAX.
- **Backend**: Flask-based API handling routing, database operations, and integrations.
- **Database**: Firebase Firestore for managing users, subjects, students, and attendance.
- **Facial Recognition**: AWS Rekognition for indexing and identifying faces.
- **Chatbot**: Gemini AI for assisting users with intelligent suggestions and actions.
- **Background Tasks**: Celery with Redis for managing time-intensive operations.

Data Flow:

- 1. User interacts with the UI.
- 2. The backend processes inputs and interacts with Firebase or AWS Rekognition.
- 3. Responses are returned, and Gemini assists with context-aware feedback.

Components and Features

1. Authentication:

- Login with role-based access (admin, teacher, student).
- User session management using Flask-Login.

2. Student Management:

- Register new students with image processing.
- Display registered students in a list with editing capabilities.

3. Subject Management:

- Create, edit, and delete subjects.
- Editable subject list with real-time updates.

4. Attendance Logging:

- Mark attendance via face recognition.
- Display attendance records with filtering and exporting options.

5. Chatbot Integration:

- Interactive guidance via Gemini AI for all operations.
- Automated suggestions based on user actions.

6. Dynamic UI:

- Tabs for navigation between functionalities.
- Progress bars for operations like face recognition.
- Detailed attendance views with photos and records.

Database Schema

Firestore Collections:

- username: Unique identifier. - role: Role type (admin, teacher, student). - classes: Associated classes for teachers. 2. Subjects: - name: Subject name. - code: Unique subject code. 3. Students: - name: Full name. - student_id: Unique ID. - image_path: AWS Rekognition image reference. 4. Attendance: - student_id: ID of the student. - timestamp: Time of logging. - subject: Subject details. - status: Attendance status.

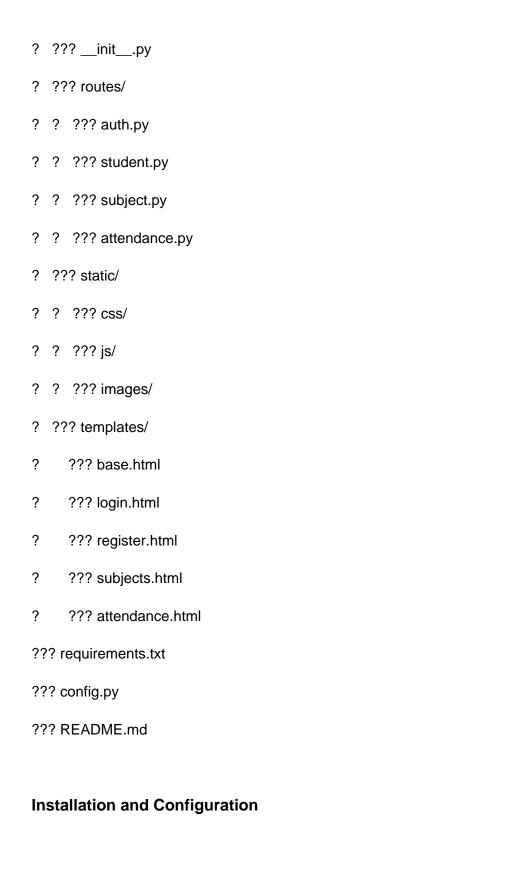
Key Functionalities

1. Users:

1. User Management:

- Admin: Create, edit, delete users, assign classes.

| - Teacher: Manage attendance for assigned classes. |
|--|
| - Student: View personal attendance records. |
| 2. Student Registration: |
| - Input validation and image enhancement. |
| - AWS Rekognition indexing for facial recognition. |
| 3. Attendance Logging: |
| - Real-time recognition, logging attendance with details. |
| - Dynamic records view with filters and exports. |
| 4. Subject Management: |
| - Add, edit, delete subjects with real-time updates. |
| 5. Chatbot Assistance: |
| - Context-aware feedback for user actions. |
| 6. Enhanced Front-End: |
| - Tab-based navigation, modern visuals, and progress bars. |
| File Structure |
| project/ |
| ??? app/ |



1. Clone the Repository:

git clone <repo_url>

| cd project/ |
|--|
| 2. Install Dependencies: |
| pip install -r requirements.txt |
| pip inotali i roquiromonto.txt |
| 3. Set Up Environment Variables: |
| - AWS_ACCESS_KEY_ID |
| - AWS_SECRET_ACCESS_KEY |
| - FIREBASE_ADMIN_CREDENTIALS_BASE64 |
| - GEMINI_API_KEY |
| |
| 4. Run the Application: |
| flask run |
| |
| Future Development Directions |
| |
| Extend Chatbot Capabilities: |
| - Automate subject management and analytics using natural language commands. |
| |
| 2. Improve Error Handling: |
| - Add detailed error pages for better debugging. |
| |
| 3. Real-Time Notifications: |
| - Notify students and teachers on attendance updates. |

| 4. | Additional | Security: | |
|----|------------|-----------|--|
|----|------------|-----------|--|

- Enhance rate limiting and data validation.

5. Advanced Analytics:

- Provide graphical insights into attendance trends.