Name: Alie Kamara
ID No: 21286
Name: Mariama Oya Kamara
ID No: 55745

Project Report - 1: https://github.com/okamay/Facial-Recognition-Based-Biometric-Access-Control-System.git

Project Report -2 : https://github.com/okamay/MyHealth-Tracker-A-React-Native-Based-Personal-Wellness-Monitoring-App.git

Project Link:

https://github.com/okamay/Facial-Recognition-Based-Biometric-Access-Control-System.git

Project Report - 1

Title: Facial Recognition-Based Biometric Access Control System

1. **Abstract:**

This project presents the design and implementation of a facial recognition-based biometric database system aimed at enhancing access control security. Developed during Semester 3 of the Biometric Database Development module, the system integrates dual authentication mechanisms—facial biometrics and traditional password login—supported by a secure, real-time web interface. The system effectively captures, stores, processes, and authenticates biometric data using modern encryption and computer vision techniques, ensuring both usability and data protection.

2. **Objective:**

To develop a secure and efficient biometric database system utilizing facial recognition to control access and manage user authentication, while meeting the core requirements of data integrity, user privacy, real-time interaction, and auditability.

3. **Motivation and Rationale:**

Conventional authentication systems relying solely on passwords are vulnerable to breaches. The growing demand for secure, non-intrusive, and user-friendly authentication methods has led to the adoption of biometrics. Facial recognition stands out for its contactless nature and rapid identification capabilities. This project harnesses this potential to build a robust, encrypted, and scalable access control system fit for academic, corporate, and institutional environments.

4. **Unique Features:**

- **Dual Authentication**: Combines facial biometrics with traditional password login for enhanced security.
- **User Registration with Live Camera**: Real-time image capture and processing through webcam.
- **Comprehensive Access Logs**: Detailed logging of access events with timestamps.
- **End-to-End Security**: Encryption of biometric data using the Fernet protocol and password hashing with bcrypt.
- **Professional Web Interface**: Responsive, user-centric interface built with modern frontend tools.
- **Admin Dashboard**: Centralized management of users and logs.
- **Local Facial Recognition Processing**: Ensures privacy and independence from third-party services.

5. **Technology Stack:**

Backend:

- **Python 3.11+**: Main development language.
- **Flask 3.1.1**: Web server and API backend.
- **SQLAlchemy**: ORM for structured database operations.
- **OpenCV**: Facial recognition and computer vision.
- **Cryptography (Fernet)**: Encrypts sensitive facial encodings.
- **bcrypt**: Secure password hashing.

Frontend:

- **HTML5 / CSS3 / JavaScript (ES6)**: Responsive design and interactivity.
- **WebRTC**: Direct camera access in the browser.
- **CSS Grid & Flexbox**: Intuitive UI layout techniques.

Database:

- **SQLite** (with PostgreSQL/MySQL compatibility): Lightweight and efficient storage.

6. **Implementation Process:**

1. **Project Setup**:
    - Built a virtual Python environment and installed all dependencies.
    - Configured Flask routing and database models.
2. **Facial Capture & Encoding**:
    - Integrated OpenCV for live camera access.
    - Captured multiple facial images to improve model reliability.

    - Encoded facial data and encrypted it using Fernet.
3. **Authentication System**:
    - Developed login and facial authentication APIs.
    - Combined image recognition with secure password verification.
4. **Web Interface**:
    - Developed user-friendly registration and login forms.
    - Implemented live camera feed and UI interaction using WebRTC and JavaScript.
5. **Access Logging**:
    - Logged every access attempt (success/failure) with user ID, method, and timestamp.
6. **Admin Tools**:
    - Enabled user deletion, stats viewing, and audit log access.

7. Testing and Evaluation:

- Manual testing was conducted via the local server (localhost:5002) to verify registration, authentication, and error handling.
- Automated test scripts (test_basic.py) ensured application integrity.
- Security tests confirmed biometric encryption, password hashing, and data minimization compliance.

8. Challenges and Solutions:

- **Challenge**: Low facial recognition accuracy under poor lighting.
    - **Solution**: Required multiple images during registration and implemented image preprocessing.
- **Challenge**: Browser-based camera permission issues.
    - **Solution**: Provided clear user instructions and fallback authentication options.
- **Challenge**: Storing sensitive biometric data securely.
    - **Solution**: Used Fernet encryption and local-only processing.

9. Project Outcomes:

- Fully functional facial recognition system for secure access control.
- Encrypted, privacy-respecting biometric data management.
- Responsive web platform for user and admin interactions.
- Comprehensive access audit trail and security features.
- Thoroughly documented system with step-by-step setup guide and API references.

10. Future Improvements:

- Integration with multi-camera systems for broader coverage.

- Real-time face spoof detection (e.g., using liveness detection).
- Integration with enterprise-level identity management systems (LDAP, OAuth).
- Deployable cloud-ready version with Docker and Kubernetes.

11. Conclusion:

This biometric access control system demonstrates the practical application of facial recognition technology for secure, user-friendly authentication. Built with scalability, security, and privacy in mind, it serves as a model for real-world implementations in education, healthcare, government, and enterprise sectors. The project aligns with modern biometric trends and proves the potential of using computer vision to improve digital security systems.

Project Link:

https://github.com/okamay/MyHealth-Tracker-A-React-Native-Based-Personal-Wellness-Monitoring-App.git

Project Report - 2

Title: MyHealth Tracker – A React Native-Based Personal Wellness Monitoring App

1. Introduction

In an era where personal health and wellness have become critical to quality of life, mobile health (mHealth) solutions offer individuals tools to proactively manage their health. This project proposes the development of a cross-platform mobile application, **MyHealth Tracker**, designed to help users monitor and improve their daily wellness by tracking water intake, sleep, physical activity, and medication adherence.

2. Objectives

The main objectives of this project are:

- To develop a **user-friendly mobile app** for personal wellness tracking.
- To enable **daily health monitoring** including water consumption, steps walked, sleep patterns, and medication schedules.
- To implement **local data persistence** using AsyncStorage for privacy and offline functionality.
- To present an **interactive analytics dashboard** that visualizes user progress and promotes healthy habits.

3. Problem Statement

Modern lifestyles often lead to neglect in essential health habits such as staying hydrated, sleeping adequately, and adhering to medication. While various health apps exist, they often lack simplicity, data privacy, or integration of multiple metrics. There is a need for a lightweight, intuitive, and private solution that consolidates wellness monitoring into one mobile platform.

4. Scope of the Project

The scope of the "MyHealth Tracker" app includes:

- Cross-platform compatibility (iOS and Android).
- Tracking of:
    - **Water intake**
    - **Step count**
    - **Sleep duration and quality**
    - **Medication reminders**
- User goal setting and customization.
- Local storage using AsyncStorage (no cloud dependency).
- Basic analytics and data visualization for user insights.

Future scope may include:

- Integration with wearable devices.
- Cloud-based backup and synchronization.
- Advanced analytics and AI-driven recommendations.

5. Methodology

The application will be developed using **React Native** with the help of **Expo** for rapid development and deployment. The development will follow the **Agile methodology** with iterative releases and testing.

Tools & Technologies:

- **Programming Language**: JavaScript (React Native)
- **Framework**: Expo
- **State Management**: Context API
- **Data Storage**: AsyncStorage
- **Charting**: Custom components for visual analytics
- **Version Control**: Git/GitHub

Key Modules:

1. **Dashboard Screen**
2. **WaterTrackerScreen**
3. **StepTrackerScreen**
4. **SleepTrackerScreen**
5. **MedicationScreen**
6. **AnalyticsScreen**
7. **SettingsScreen**

6. Deliverables

- Fully functional mobile app (APK/IPA for Android/iOS).
- Source code and documentation (GitHub or ZIP).
- User manual and installation guide.

- Final project report.

7. Timeline

Week	Activity
1-2	Project setup, UI wireframes
3-4	Implement water and step tracking modules
5-6	Develop sleep and medication modules
7	Integrate analytics and dashboard
8	Testing and debugging
9	User testing and feedback implementation
10	Final report and submission

8. Expected Outcome

- A working mobile app that helps users track and visualize their daily wellness habits.
- Improved awareness and consistency in health practices among users.
- A scalable codebase that can be extended to include additional health metrics or wearable device integration.

9. Conclusion

"MyHealth Tracker" is an innovative approach to making health tracking simple, efficient, and private. By consolidating multiple wellness metrics into a single mobile platform and using local storage for data privacy, this project aims to empower users to take charge of their health in an informed and convenient way.