Project Report: AI Virtual Personal Fitness Coach

Author: Harshit Shukla **Date:** November 9, 2025

1. Introduction

In today's world, at-home fitness has become extremely popular. However, a major challenge for individuals working out at home is the lack of real-time feedback and progress tracking, which is traditionally provided by a personal trainer. This often leads to improper form, risk of injury, and a lack of motivation.

This project, the "AI Virtual Personal Fitness Coach," aims to solve this problem by creating a full-stack web application. It uses a standard webcam to provide users with a real-time, interactive, and data-driven personal training experience, completely free of cost.

2. Abstract

This project presents a full-stack, AI-powered web application built using Python and Streamlit. The system leverages Google's MediaPipe library for real-time human pose estimation, allowing it to track 33 different body landmarks via a webcam feed. The core logic calculates joint angles to accurately count repetitions and sets for 7 different exercises, including Bicep Curls, Squats, and Push-ups.

The application is a complete full-stack solution, featuring a secure user authentication (Login/Signup) system powered by Firebase Authentication. All user workout data (exercise type, sets, reps, and duration) is persistently stored in a Firebase Firestore database. This was achieved by implementing a robust REST API solution to overcome library dependency conflicts. The Streamlit UI provides a clean sidebar for navigation, exercise selection, and setting personal targets for sets and reps. A multi-language (Hindi/English) Text-to-Speech (TTS) voice assistant provides real-time audible feedback to the user, creating an immersive coaching experience.

3. Tools Used

- **Python:** The core programming language for the entire application.
- **Streamlit:** The primary framework used to build the interactive web application frontend (UI).
- **OpenCV:** Used to capture and process the live video feed from the user's webcam.
- **MediaPipe:** Google's AI library used for real-time human pose estimation to detect body landmarks.
- **NumPy:** Used for high-performance mathematical calculations, specifically for calculating joint angles.
- **Pyrebase4:** A Python wrapper for Firebase, used specifically for the Email/Password Authentication system.
- **Requests:** A Python library used to make HTTP requests to Firebase Firestore's REST API for all database operations (saving and loading logs).

4. Steps Involved in Building the Project

1. **Setup & Prototyping:** The project began with setting up a virtual environment and installing core libraries. Initial prototyping focused on capturing a webcam feed with OpenCV and overlaying MediaPipe's pose landmarks.
2. **Core Logic (Angle Calculation):** A helper function (utils.py) was created using NumPy to calculate the angle between three 3D points (e.g., shoulder, elbow, wrist).
3. **Rep Counting (State Machine):** A "state machine" logic was built. For a bicep curl, the app tracks a 'down' state (angle > 160°) and an 'up' state (angle < 30°). A rep is only counted when the user successfully transitions from 'down' to 'up' and back.
4. **Feature Expansion:** The logic was expanded to include 7 different exercises (Squats, Push-ups, Lunges, Overhead Press, Jumping Jacks, High Knees), each with its own unique angle logic. A "Target Sets & Reps" feature was added to guide the user through a complete workout.
5. **Full-Stack Backend (Authentication):** Firebase was integrated as the backend. Pyrebase4 was used to create a secure Login/Signup system using Email & Password.
6. **Database (REST API Fix):** This was the most critical technical challenge. The google-cloud-firestore library caused severe dependency conflicts with mediapipe (the AttributeError bug). To solve this, the entire database logic was re-architected. The app now uses Python's requests library to communicate directly with the Firebase Firestore REST API, using the user's auth token for security. This fixed all bugs and stabilized the app.
7. **Frontend & UI/UX:** The Streamlit UI was redesigned into a multi-page app (Login vs. Coach). A persistent sidebar was created for all settings, including exercise selection, target goals, voice assistant toggles, and the user's complete, scrollable workout log.
8. **Deployment Preparation:** The code was given a final update to read the Firebase keys from Streamlit's "Secrets" (environment variables) on the server, while still allowing local testing by pasting the keys into the sidebar.

5. Conclusion

This project successfully evolved from a simple rep counter into a complete, full-stack, deployable web application. It solves the real-world problem of at-home fitness by providing a data-driven, interactive coach. The user's workout history is now securely saved and retrieved from a cloud database, making it a persistent and personalized tool.

Future work could involve adding more exercises, creating a social "leaderboard" feature, or building a recommendation engine to suggest workouts based on the user's past activity.