

Project Team : CSM-C_B 06

21BQ1A42F1 – R. PAVAN KUMAR

21BQ1A42G6 – S. NANDINI

22BQ5A4217 – SK. ABDUL NAYEEM

22BQ5A4218 – SK. BAJI BABA

Project Name: Yogo Pose Detection System

Abstract:

The **Yogo Pose Detection System** is a cutting-edge web-based platform that utilizes **Deep Learning** to accurately detect and analyze yoga poses. The system leverages powerful deep learning models, including **MobileNetV2**, **VGG-16**, **DenseNet-121**, to provide real-time pose prediction and feedback, ensuring that users receive instant corrections for improved alignment. This innovative solution aims to enhance the practice of yoga by offering users an interactive and dynamic platform to learn, practice, and track their yoga progress.

Key features of the system include secure **user and admin authentication** with **role-based access**, **pose prediction via image uploads**, and **real-time yoga pose detection using webcam input**. The system also incorporates a **pose search interface** that provides detailed instructions and related content for specific yoga poses. Additionally, **sentiment analysis** of user feedback is employed to gather insights into user satisfaction, while an **admin dashboard** offers real-time tracking and metrics for better system management.

The project uses advanced technologies, including **TensorFlow/Keras** for model deployment and **SQLite3** for data storage. By offering personalized **pose recommendations** based on health goals, and integrating 3rd party APIs, the Yogo Pose Detection System provides a seamless experience for users and a robust platform for administrators. This system is designed to support both beginner and experienced yoga practitioners, promoting overall health and well-being.

Title of the Project :

Yogo Pose Detection System

Client

There is no explicit mention of a specific client in the document. However, the system is designed for individuals practicing yoga, yoga instructors, and fitness organizations looking to integrate technology into their training methods.

Objectives of the System

The main goals of the **Yoga Pose Detection System** include:

1. **Real-Time Yoga Pose Detection**
 - Uses deep learning models (MobileNetV2, VGG-16, DenseNet-121) to provide instant feedback on yoga poses.
2. **Pose Prediction via Image Upload**
 - Allows users to upload images of their yoga poses for classification and correction.
3. **Enhancing Yoga with AI Assistance**
 - Provides Yoga Pose Details to help users improve their posture.
4. **Interactive and User-Friendly Interface**
 - Includes features like a **pose search engine** and **detailed instructions** for yoga poses.
5. **User and Admin Authentication**
 - Secure login system with role-based access.
6. **Integration of Sentiment Analysis**
 - Collects and analyzes user feedback to improve the platform.
7. **User FeedBack System:**
 - Feedback will be analyzed using sentiment analysis to understand user satisfaction.
8. **Admin Dashboard & Analytics**
 - Provides real-time tracking of user engagement and system performance.

Users of the System

The primary users of the Yogo Pose Detection System include:

1. **General Users (Yoga Practitioners)**
 - Individuals looking to improve their yoga poses with AI-driven feedback.
2. **Yoga Instructors & Trainers**
 - Coaches who can use the system to guide students remotely and provide pose corrections.
3. **Health & Fitness Enthusiasts**
 - Users interested in improving flexibility, balance, and mindfulness through yoga.
4. **System Administrators**
 - Responsible for managing users, analyzing feedback, and ensuring system functionality.

Functional Requirements

These are the core functionalities that the system must provide:

1. **User Authentication System**
 - Secure login with role-based access (User/Admin).
2. **Yoga Pose Prediction**
 - Image-based pose classification using AI models.
3. **Real-Time Pose Detection**
 - Uses a webcam to analyze poses and provide instant feedback.
4. **Yoga Pose Search Engine**
 - Users can search for specific poses and get detailed instructions.
5. **Sentiment Analysis on Feedback**
 - Evaluates user feedback to measure satisfaction and improve the system.
6. **Personalized Pose Recommendations**
 - Suggests yoga poses based on fitness goals.
7. **Admin Dashboard & Performance Metrics**
 - Displays real-time analytics and user engagement trends.
8. **Yoga Blog Section**
 - Contains articles, videos, and expert tips on yoga practice.
9. **Database Integration & Management**
 - Stores user data, pose classification results, and feedback.

Non-Functional Requirements

These define the quality and operational aspects of the system:

1. **Performance & Speed**
 - The system should provide real-time feedback with minimal latency.
2. **Scalability**
 - Should handle a growing number of users and pose detection requests.
3. **Security**
 - Ensures user data privacy with encrypted authentication and storage.
4. **User-Friendly Interface**
 - Simple, intuitive, and visually appealing design.
5. **Cross-Platform Compatibility**
 - Should work on different devices (PCs, tablets, and mobile phones).
6. **Reliability & Accuracy**
 - The deep learning models must ensure high pose classification accuracy.
7. **Integration with 3rd Party APIs**
 - Allows expansion of features, such as additional fitness tracking tools.

Software Requirements

The system relies on several technologies:

Backend

- **Python (Flask or Django)** – Handles API requests and data processing.
- **TensorFlow/Keras** – Deep learning framework for pose classification.
- **SQLite3** – Lightweight database for storing user and system data.

Frontend

- **React.js or Angular.js** – Builds the user interface.
- **HTML, CSS, JavaScript** – For styling and interactivity.

Other Dependencies

- **OpenCV** – For image processing and pose analysis.
- **NumPy & Pandas** – For handling data.

Hardware Requirements

To run the Yoga Pose Detection System effectively, the following hardware is needed:

1. **For End Users (Yoga Practitioners & Instructors)**
 - Any device with a **webcam** (PC, tablet, or smartphone).
2. **For Server Hosting**
 - **Processor:** Intel i5/i7 or AMD Ryzen 5/7 (or higher).
 - **RAM:** Minimum 8GB (16GB recommended for real-time performance).
 - **GPU:** NVIDIA RTX 3060 or higher (for deep learning model inference).
 - **Storage:** SSD (256GB minimum, 512GB recommended).
3. **For Development & Testing**
 - **High-performance PC with GPU** (for training models efficiently).