Pattern Recognition System Final Report
Group 7 -Yoga Master AI

Group Member:
Dong Yuantong/e1337515@u.nus.edu/A0292041N(Group Leader)
Li Qiuxian / e1351736@u.nus.edu /   A0297598X
Gao Yunjia / e1349640@u.nus.edu / A0295502E
Zhao Lanting / e1351814@u.nus.edu / A0297676A
Liu Weixuan /e1349820@u.nus.edu/ A0295682M

CONTENTS

1.Introduction

As modern life becomes increasingly fast-paced and stressful, more people are turning their attention to mental and physical well-being. Yoga, as a holistic mind-body exercise, has gradually gained widespread attention. It not only enhances flexibility and strength but also effectively alleviates stress, improves sleep quality, and helps manage chronic pain, making it popular among a growing number of practitioners. However, many yoga enthusiasts lack professional guidance, leading to frequent posture inaccuracies that can diminish the effectiveness of their practice or, worse, result in discomfort or injury.

This project aims to develop an intelligent yoga recommendation and evaluation system that combines Natural Language Processing (NLP) and computer vision technologies to offer users personalized yoga posture recommendations and real-time feedback. Based on the user's yoga level, goals, and health conditions, the system provides safe and tailored posture suggestions. Additionally, the posture recognition feature analyzes and assesses the accuracy of the user's poses in real time, allowing users to correct mistakes promptly, ensuring both safe and effective practice. For users with specific health requirements, such as back pain or knee issues, the system prioritizes poses that are body-friendly and avoids movements that may cause discomfort.

With its advanced posture recognition and personalized recommendation capabilities, this system not only enhances the yoga experience but also provides users with a scientific, safe, and goal-aligned yoga practice solution. In the rapidly growing global yoga market, this project fills the gap for personalized guidance and posture evaluation tools, offering users an innovative and intelligent yoga solution.

2.Market Research

2.1 Industry Overview

In recent years, the global yoga market has experienced significant growth. According to market research analysis, the global yoga market is expected to reach $107.1 billion by 2023, with projections to exceed $200.35 billion by 2030, marking a compound annual growth rate (CAGR) of 9.4%. This growth is mainly attributed to increasing awareness of mental and physical well-being, especially post-COVID-19, as more people opt for home workouts. According to Statista, around 36 million adults in the United States practiced yoga in 2021, with this number expected to rise to 48 million by 2024. This trend reflects consumers' broad acceptance of yoga as a tool for relaxation and health management.

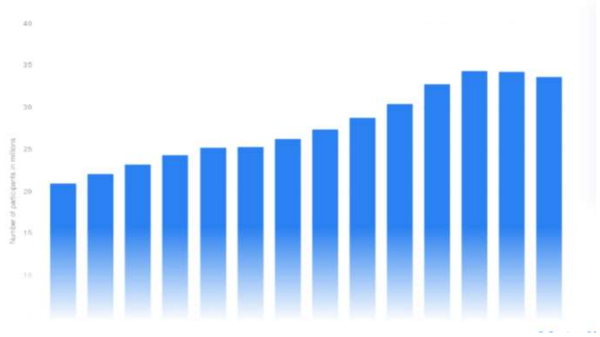


Figure1. Number of yoga participants in the United States from 2010 to 2023

2.2 Current Market Trends

1. Balance Between Online and Offline Classes: Although offline yoga classes still dominate the market (74.3%), the growth rate of online yoga courses is noteworthy. The online market is projected to achieve a CAGR of 10.5% in the coming years. The pandemic prompted many yoga studios to shift to online teaching, allowing users to practice at more flexible times. Additionally, online classes offer a variety of options, from beginner to advanced levels, catering to users of different skill levels. Popular online platforms like Alo Moves and Gaia offer a range of yoga styles that users can choose from according to their needs.

2. **Gender and Age Distribution:** Female users dominate the yoga market, accounting for approximately 72%. However, the proportion of male participants has increased significantly, as awareness of yoga's health benefits encourages more men to join. According to Yoga Journal, the percentage of male yoga practitioners rose from 17.8% in 2012 to 28% in 2021. Additionally, people aged 30-50 represent the largest user group (43.5%), often viewing yoga as an effective way to relieve stress and maintain health. This group typically seeks to improve flexibility, strength, and mental balance through yoga.

3. **Demand for Personalization:** Users increasingly expect personalized experiences. Systems that can provide tailored recommendations based on users' health conditions and yoga goals will have a competitive advantage. For example, for users with specific health needs (such as pregnant women or those with chronic illnesses), a system offering safe and suitable posture recommendations is highly valued. Market analysis shows a rising demand for intelligent recommendation systems based on personal health data, particularly in applications related to disease prevention and health management.

2.3 Competitive Landscape

Currently, several yoga apps (e.g., Glo, CorePower Yoga) exist in the market, with most relying on pre-recorded classes and lacking real-time feedback and personalized health assessment functions. This gap presents an entry point for our yoga AI application, which leverages advanced posture recognition technology and a personalized recommendation system to meet user needs. Additionally, some emerging companies are combining AI and computer vision technology, enhancing user experience through real-time posture detection and personalized feedback. Products like YogiFi and Mirror use smart devices and applications to provide personalized yoga guidance, promoting innovation in the market.

2.4 Opportunities and Challenges

2.4.1 Opportunities:

Technological Innovation: With rapid advancements in AI, deep learning, and natural language processing, the yoga market is undergoing a tech-driven transformation. By using these technologies, our system can provide real-time posture evaluation and personalized recommendations, significantly enhancing user experience. For instance, deep learning models like PoseNet enable accurate posture recognition, offering real-time feedback on users' practice.

Market Education and Promotion: Although yoga’s health benefits are well-known, awareness of technology-driven yoga guidance tools needs to be raised among a broader audience. Educating users on how to effectively use AI technology to enhance their yoga experience will be a key aspect of market promotion. Through community activities, online promotion, and social media, user acceptance of new technology can be increased, thereby boosting market participation.

Policy Support and Industry Standards: With government support for yoga promotion, industry standardization is becoming increasingly important. For example, initiatives like India’s Fit India movement serve as effective driving forces. Such policy support helps promote the healthy development of the yoga industry. Additionally, certification standards for yoga instructors and related professionals are gradually forming, enhancing professionalism and credibility within the industry.

2.4.2 Challenges:

Market Competition: Despite wide-ranging opportunities, competition is intense. Several well-established yoga apps (such as Glo and CorePower Yoga) have strong user bases due to their rich content and excellent user experience. New entrants must overcome the challenge of standing out in this competitive landscape.

Changing User Habits: Although demand for online yoga courses is rising, many users still prefer face-to-face instruction, especially beginners who require precise guidance. Convincing users to adopt digital yoga instruction is a significant challenge.

Technical Complexity: While technological innovation provides opportunities, successfully integrating these new technologies into existing systems also presents numerous challenges. Considerable resources are required to ensure system accuracy and reliability, with continuous technical updates and maintenance necessary to maintain competitiveness.

3.Business Value

The yoga AI application project offers significant business value by integrating real-time posture recognition and personalized recommendations, catering to the growing demand for customized fitness experiences. This system addresses diverse user needs, from beginners to those with specific health requirements, by enhancing user acquisition and retention through tailored engagement and satisfaction. Leveraging data-driven optimization, the application refines recommendations based on user behavior, and community-building efforts foster user loyalty, encouraging interaction, sharing, and recommendations. Such a strategy not only improves user retention but also reduces new user acquisition costs.

In addition to its core offerings, the project's educational potential promotes user loyalty and health awareness through accessible content on yoga techniques and overall wellness. Furthermore, with expansion opportunities into related wellness fields, such as meditation and nutrition management, the application could serve as a comprehensive wellness solution. This multi-dimensional approach establishes a competitive edge in the health tech market, enhancing both immediate and long-term user engagement and amplifying brand influence.

4.Objectives of Project

The primary objective of this project is to develop an intelligent yoga recommendation and evaluation system that provides users with personalized yoga pose suggestions and real-time feedback on posture accuracy. By leveraging advanced technologies like Natural Language Processing (NLP) and computer vision, the system aims to enhance user engagement, ensure safe practice, and support individual wellness goals. The specific objectives are as follows:

Yoga Pose Recommendation: Offer personalized yoga pose suggestions based on each user's health conditions, yoga level, and specific practice goals, such as flexibility improvement, stress relief, or strength building.

Posture Assessment: Utilize a pose estimation model to detect and analyze the user's body pose through uploaded photos, comparing it with standard pose references to identify deviations and ensure accuracy.

Real-Time Feedback: Provide detailed, real-time feedback on specific body parts that require adjustment, offering corrective suggestions to help users maintain proper form and reduce the risk of injury.

Personalized Health Considerations: Integrate health condition assessments to filter and recommend yoga poses that are safe and appropriate for users with specific health needs, such as back pain or knee issues.

Continuous Learning and Optimization: Use user data to optimize recommendation algorithms and continuously improve the accuracy of posture recognition, adapting to the unique needs of each individual over time.

By meeting these objectives, the system aims to provide a scientific, safe, and tailored yoga experience that aligns with the needs of a wide range of users, from beginners to advanced practitioners, promoting sustained engagement and long-term health benefits.

5.Overview of Dataset

5.1 Dataset Structure

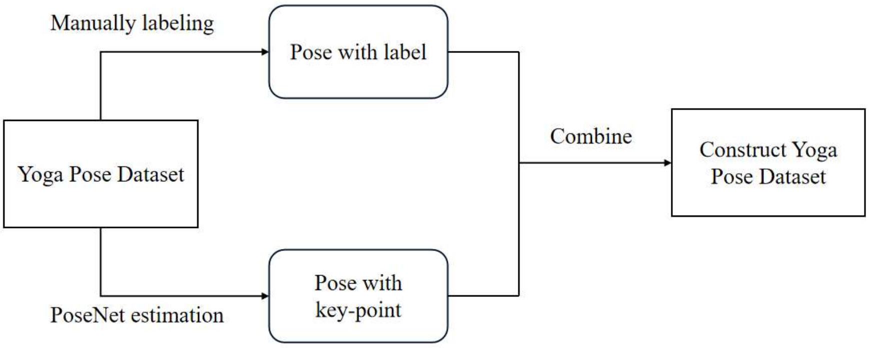


Figure2. Flowchart of data processing

This flowchart illustrates the process of constructing a yoga pose dataset. Initially, the base yoga pose dataset undergoes two types of processing: one part is manually labeled as "Pose with Label," while the other part uses the YOLO V11 model for key-point estimation, creating a "Pose with Key-Point" dataset. Then, the labeled poses and key-point data are combined to form a comprehensive yoga pose dataset. This final dataset includes each pose's name along with its corresponding key points, providing a high-quality data foundation for pose recognition and alignment analysis.

5.2 Dataset Collection and Preparation

5.2.1 Data Collection:

A robust and diverse dataset is essential for accurately training and testing the pose estimation system. The following data collection steps are proposed:

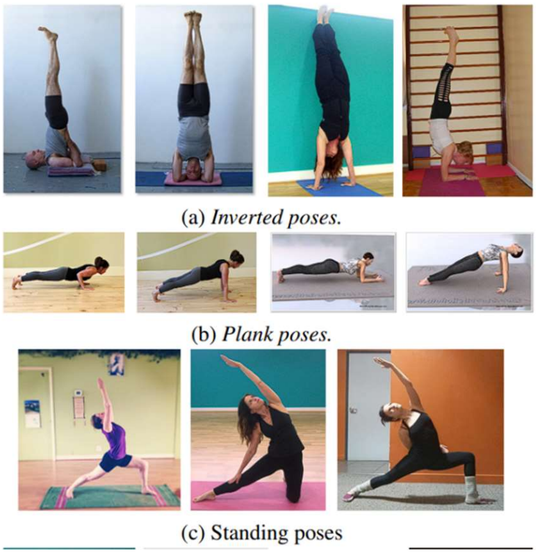
·Yoga Pose Dataset: Use publicly available datasets like Yoga-82 or collect custom data. The dataset should cover a wide range of yoga postures (e.g., Tree Pose, Warrior II, Downward Dog).

·Diversity of Data:

Body Types and Flexibility Levels: Ensure the dataset includes individuals of different body types, flexibility levels, and skill sets to make the system generalizable.

Angles and Lighting: Capture poses from multiple angles and under varying lighting conditions to improve model robustness.

Camera Positioning: Data should be collected using both front-facing and side-facing camera angles to cover the most relevant views for pose correction.


(a) *Inverted poses.*

(b) *Plank poses.*

(c) Standing poses

Figure3. Sample of Dataset

5.2.2 Data Preparation:

·Manually labeling: Combined with the recommend system, our system can recommend different yoga poses based on different user conditions and preferences (e.g. sports performance, uncomfortableness, etc.) Therefore, we need to do a manually label with the dataset to construct the poses with different features, which can be utilized by the recommend system.

·Data Augmentation: To enhance model generalization, augment the dataset by rotating, scaling, and flipping images to simulate various camera orientations and scenarios.

·Key-point Detection: Each image or video frame will be annotated with key points representing major body joints (e.g., shoulders, elbows, knees, wrists). Tools like COCO Annotator or LabelMe can be used.

·Convert to required type: Once we got the images and its corresponding json file after labeling with key-points. We need to convert them into COCO and YOLO format to meet the requirement of model training. After converting, we can get text files, which contain the key-points and border box information for each image.

·Training and Validation Splits: Split the dataset into training, validation, and test sets. The training set will be used for model learning, while the validation set will help tune hyperparameters. The test set will evaluate the system's final performance.

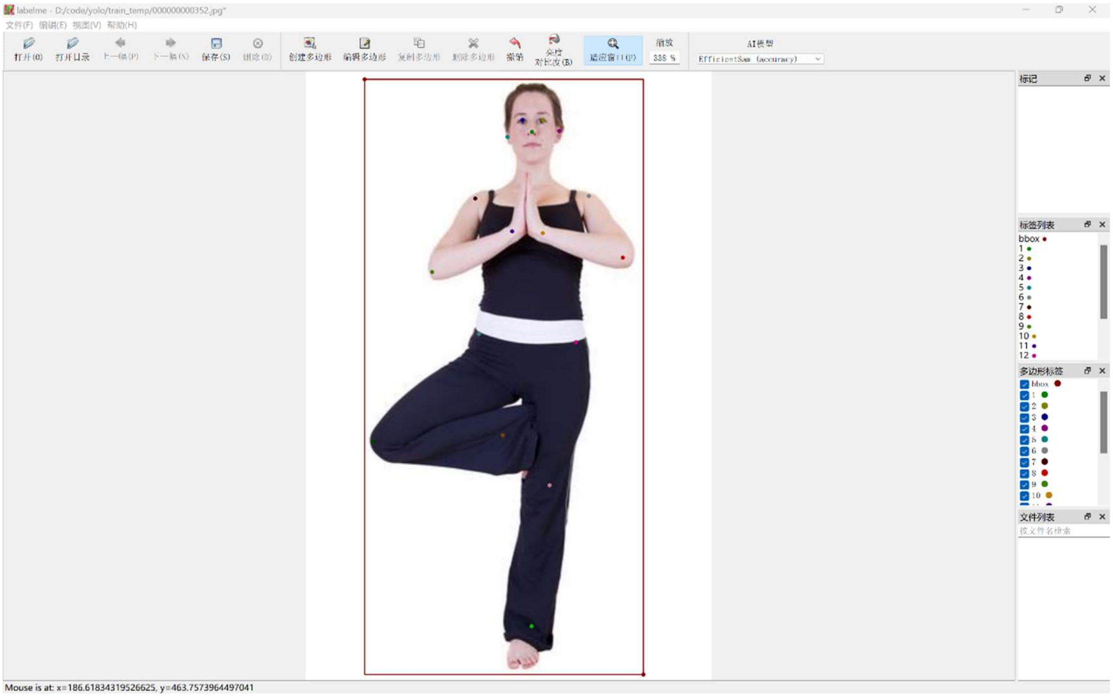


Figure4. Use labelme to label key-points

6.System Design

6.1 System Architecture

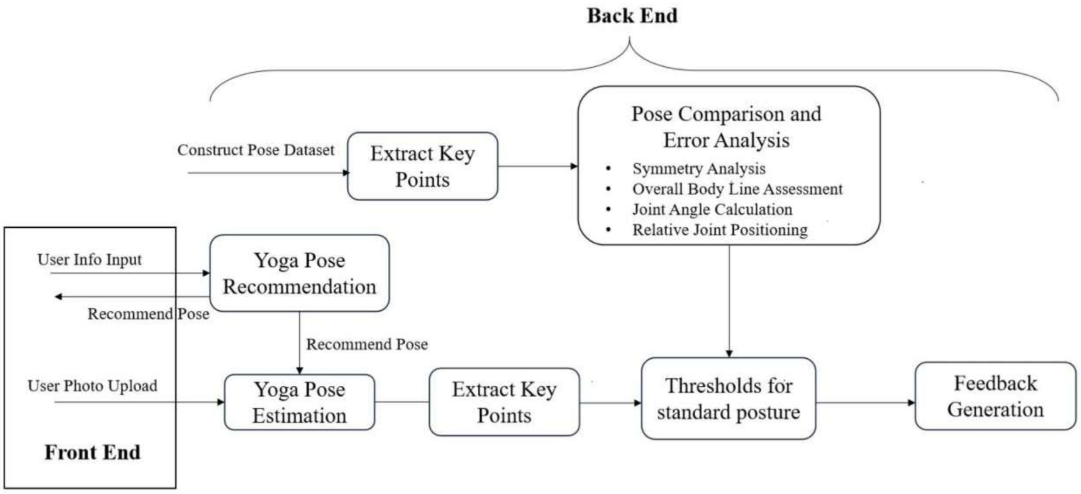


Figure5. Architecture Diagram

1. Front End:
   Users can input personal information and upload yoga posture photos through the front end.
   The front end transmits user data to the backend processing system and receives personalized posture recommendations and feedback to display to users.

2. Yoga Pose Dataset Clustering:
   This module classifies yoga poses into clusters, building a comprehensive pose dataset.
   The constructed dataset provides standard pose references for the recommendation and evaluation system, ensuring a robust library of poses.

3. Yoga Pose Recommendation:

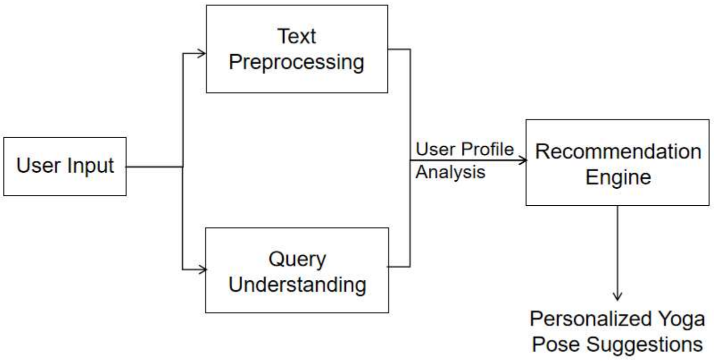


Figure6. Recommendation Flowchart

   Based on the user's health conditions, yoga level, and practice goals, the system generates personalized posture recommendations in the backend.
   Recommended poses are tailored to the user's physical needs and matched with the clustered standard pose dataset.

4. Yoga Pose Estimation:
   The system analyzes the user's uploaded posture photo to extract key body points.
   Through a pose estimation model, the user's pose is identified and compared with the recommended standard pose to ensure accurate practice.

5. Pose Comparison and Error Analysis:
   This module is the core of posture evaluation, analyzing the accuracy of the user's pose through four sub-modules:
   Symmetry Analysis: Evaluates the symmetry of the left and right sides of the body.
   Overall Body Line Assessment: Analyzes the overall body line, including spine alignment and leg extension.
   Joint Angle Calculation: Calculates the angles of key joints using mathematical methods.
   Relative Joint Positioning: Checks the relative positioning of key points, ensuring correct placement of body parts such as shoulders and knees.

6. Feedback Generation:
   Based on the results of the pose error analysis, the system generates specific feedback to guide the user on areas that need adjustment.

This feedback is sent back to the front end, helping users correct their poses in real-time to ensure safe and effective practice.

6.2 Technology Stack

- Back-End Framework: Flask for creating APIs to handle image uploads and process requests from the front end.
- Front-End Framework: Vue.js for a responsive user interface that allows users to upload images and view feedback.
- Axios: Used to send HTTP requests from the Vue.js frontend to the Flask backend, essential for fetching data (e.g., user-specific recommendations) and submitting user inputs to the server.
- Database：MySQL for stores and retrieves structured data, including user data, preferences, recommendation metadata, and other system data.
- Recommendation System

  A. SpaCy is used for text processing tasks such as part-of-speech tagging, named entity recognition, and stemming, which helps in understanding the natural language content of user inputs.

  B. Scikit-learn is used to implement the item-based collaborative filtering algorithm, providing tools for similarity calculations, including cosine similarity. Additionally, rule-based filtering algorithms are applied to refine the initial recommendations by further screening results based on specific criteria.

  C. OpenAI API for Smart Dialogues: Integrates OpenAI's API to generate intelligent dialogues and suggestions, offering virtual guidance for users.

  D. Voice Input: Integrates voice-to-text input, allowing users to interact hands-free during practice for a more convenient experience.

- Pose Estimation and Comparison

  A. YOLOv11 Keypoint Detection: Detects key body points (e.g., elbows, knees, shoulders) for real-time feedback with low computational requirements.

  B. Pose Comparison Algorithm: Analyzes joint angles, relative positions, and symmetry to detect deviations from standard poses, providing users with specific adjustment recommendations.

  C. Real-Time Feedback:

  Model: GLM-4-0520 model is used to generate clear, personalized feedback, helping users adjust poses in real time.

- Refine YOLO Model & Construct YOLO Yoga Databse

  A. Labelme: Labelme is a versatile, open-source tool for annotating images, especially for keypoint labeling.

B. Google Colab: Google Colab is a cloud-based platform ideal for running Python code, especially useful for machine learning and data annotation tasks like refine a pretrained model.

C. Ultralytics: Ultralytics provides a user-friendly library for YOLO models, particularly valued for its train command, which simplifies training and fine-tuning YOLO models using pretrained weights. This library streamlines the model refinement process, making it ideal for enhancing model performance on custom datasets, like yoga pose estimation.

6.3 System Components

6.3.1 Data Storage Component:

1. User Information Database: Stores comprehensive user details, including age, gender, height, weight, physical condition, and other relevant data. This information serves as a foundation for recommending personalized yoga poses and enables subsequent user analysis and personalized service optimization.
2. Yoga Pose Database: Contains essential details about various yoga poses, such as pose name, descriptions, targeted body parts, difficulty level, difficulty level, and associated images. This database serves as the primary source for selecting and recommending yoga poses, ensuring that the system can match poses to each user’s needs effectively.

6.3.2 Natural Language Processing Component:

1. Text Preprocessing Module: Prepares and refines user-inputted text by removing stop words, applying stemming or lemmatization, and converting text to lowercase. This preprocessing improves the accuracy of intent analysis and key information extraction for better understanding of user needs.
2. Intent Recognition Module: Analyzes pre-processed user text using NLP technology combined with predefined rules or models to identify the user's main intent. for example, whether the user is asking for yoga poses suitable for a specific physical condition, recommendations based on training goals, or simply chatting needs, and so on. This ensures accurate responses aligned with user intent.
3. Entity Extraction Module: Extracts critical entities from user input, such as body parts, training goals, and difficulty levels. These entities are then matched with data in the yoga pose database, enabling the system to provide precise yoga pose recommendations based on specific user needs.

6.3.3 Recommendation Engine Components:

1. Feature Extraction Module: Constructs feature vectors for both users and yoga poses based on information in the user database and yoga pose database. For users, feature vectors may

incorporate exercise experience, physical condition, and training goals. For yoga poses, feature vectors include attributes like difficulty level, involved body parts, and targeted training goals.
2. Similarity Calculation Module: Calculates the compatibility between the user’s and yoga pose feature vectors using cosine similarity. This helps to determine the yoga poses that best align with each user's requirements.
3. Filtering Module: Utilizes the user’s physical condition data, such as areas experiencing pain, to filter out poses that involve those specific body parts. This ensures that the recommended poses do not aggravate any physical discomfort, providing safe and appropriate options for each user.

6.3.4 Image Upload and Preprocessing Module

This module allows users to upload pose images through the front-end interface and pre-process the uploaded images. The processing steps include image resizing, cropping and normalisation to ensure that the image data is suitable for model input. The main function of this module is to ensure that the images uploaded by the user meet the system analysis requirements and provide standardised input for subsequent pose detection and feature extraction.

6.3.5 Pose Estimation Module

The module uses YOLOv11 to extract human key points (e.g. shoulders, elbows, hips, knees and ankles) from preprocessed images. The model identifies the exact location of these key points and generates x, y coordinates for each key point, providing the base data for feature extraction and pose analysis. This module ensures that the posture assessment system is able to accurately capture the joint positions in the user's posture.

6.3.6 Feature Extraction and Analysis Module

In this module, the system calculates features of the user's posture from the detected key points, including joint angles, relative positions between joints, and symmetry of the left and right sides of the body. These features are used to fully characterise the accuracy of the user's posture. For example, the module determines the relative positions of the shoulders and hips through geometric calculations and analyses the symmetry of the joints to ensure a stable and balanced posture. The output of this module provides data support for subsequent threshold comparisons.

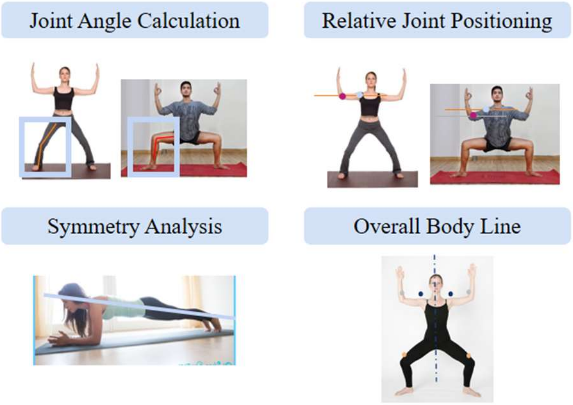


Figure7. Four different types of comparison

6.3.7 Threshold Comparison Module

This module is responsible for comparing the user's postural features with preset standard thresholds. The standard thresholds are set based on the analysis of standard yoga poses, and each threshold represents an acceptable range on a specific feature (e.g., joint angle, relative position, or symmetry). This module performs deviation checking for each feature and flags the parts of the pose that are not within the tolerance range, providing specific deviation data for generating feedback.

6.3.8 Feedback Generation Module

The Feedback Generation Module uses the deviation detection results to generate personalised feedback via the GLM-4-0520 language model. If certain joints or positions in the user's posture do not match the standard posture, the module will provide specific suggestions to help the user adjust their posture. For example, if the angle of the left knee is too far out of alignment, the system will indicate that ‘the left knee is bent too far, try straightening it for better alignment’. This module ensures that the user receives clear, specific feedback so that real-time adjustments can be made during the exercise.

6.4 System Workflow

Overall, the yoga master is a linear process system. Firstly, the system recommends yoga poses. Users upload yoga images, which are recognized and compared by the system. Finally, the system provides recommendations. But in practical applications, users can choose whether the system recommends yoga poses, and can ask other questions about yoga to the system. The system's NLP API can also help users provide relevant information.

6.4.1. Yoga Pose Recommendation:

Users first provide the system with key information such as their preferences for yoga poses and the body parts they wish to exercise, helping the system determine the information needed to recommend yoga poses. The information provided by the user will be processed by the NLP API integrated in the backend, and the user's requirements will be processed and returned to the backend for corresponding recommendations. The integrated NLP API also supports users to inquire and communicate about yoga content. In this section, a simple chatbot and backend recommendation algorithm are mainly combined to meet the needs of users.

6.4.2. Pose estimation and comparison:

In this section, users only need to provide their own yoga movement pictures, which clearly display their movements. The system provides users with a ten second countdown photo button, allowing them enough time to perform yoga movements after pressing the button. After uploading the user's image to the system, the system will first perform keypoint detection on the user's image and determine which type of yoga pose the user is performing. After determining the key points and yoga pose categories in the user's image, the system calculates the angles and symmetries formed by the key points in the image, and compares these values with preset thresholds to obtain an evaluation of whether the user's actions are standard.

6.4.3. Feedback Generation

After obtaining the posture comparison results, the system sends the results to the NLP API to beautify and output them in the chat window. The comparison result is obtained by comparing the key point information of the uploaded images by the user with the preset threshold information. This result needs to be converted into yoga movement correction suggestions that are convenient for users to read and understand. Our system provides a preliminary adjustment suggestion, such as how much to raise the arm, and NLP API helps the system process these suggestions into more user-friendly content.

6.4.4. Extra Features

The system also includes some additional features, such as users being able to start a new conversation at any time and adjust current account information at any time. These will be showcased in our demonstration video.

7.System Development & Implementation

7.1 Development Process

7.1.1 Development Methodology

The Yoga Master AI project followed an agile development methodology, focusing on iterative improvements and user feedback integration to refine system features. By utilizing agile methods such as sprints and continuous evaluation, the team was able to address technical challenges and adapt to new requirements quickly.

7.1.2 Requirement Analysis

The project began with an in-depth requirement analysis aimed at identifying the core functionalities needed for an intelligent yoga recommendation and evaluation system. Through consultations with stakeholders and potential users, critical components such as real-time posture assessment, personalized recommendations, and feedback mechanisms were identified as priorities. The user requirements were broken down into actionable development tasks to guide the project's initial stages.

7.1.3 Technology Selection

Key technologies and tools were selected based on their suitability for the specific demands of the project:

- Python was chosen as the backend programming language for its extensive libraries in machine learning and computer vision, crucial for implementing yoga pose recognition.
- YOLOv11 was utilized for keypoint detection, which allows the system to recognize critical body points, such as shoulders and knees, for accurate posture evaluation.
- Vue.js served as the frontend framework due to its reactive data-binding capabilities, enhancing user interaction by providing real-time feedback.
- The K-means clustering algorithm was integrated for clustering similar yoga poses, enabling a structured recommendation system that groups poses based on difficulty and target body parts.

These technologies supported both the frontend and backend functionalities, ensuring seamless integration between user input, pose analysis, and feedback delivery.

Version Control and Collaboration

7.2 Implementation Details

7.2.1 Data Collection and Preprocessing

User Information Collection: When users first access the system, they are guided to enter basic information such as age, gender, height, weight, and physical condition. This data is stored in the user info section.

Text Preprocessing: For the natural language text input by users, the process begins with tokenization, breaking the text into individual words or phrases. Next, stop words that do not significantly contribute to understanding user intent are removed. Finally, the remaining words undergo stemming or lemmatization to unify different word forms to their base form for further analysis and processing.

7.2.2 Front-end implementation

The front-end implementation of the Yoga Master project is centered around Vue.js, chosen for its reactive and efficient data-binding capabilities that create a dynamic and interactive user experience. Vue.js components are structured modularly, which ensures easy maintenance and scalability. Key

UI components include a message input and chat window for real-time interactions, a camera module for direct pose photo uploads, and a recommendation display that showcases tailored yoga poses in an intuitive layout.

Navigation and data synchronization are handled seamlessly with Vue Router, creating a single-page application (SPA) experience. As users navigate across different views, such as the chat interface, recommendation page, and feedback display, Vue’s reactivity ensures that the UI automatically updates whenever the data model changes. This smooth interaction flow is particularly essential in offering users an uninterrupted and responsive practice session.

Axios is employed for HTTP communication between the front and back end, allowing the system to fetch pose recommendations, upload pose images, and retrieve analysis feedback in real time. This API integration enables a responsive feedback loop, where users can view suggested adjustments to their poses immediately after analysis, enhancing the adaptability and efficiency of each yoga session.

To further enhance usability, Yoga Master incorporates the Web Speech API, enabling users to interact with the application via voice commands. This hands-free functionality is particularly valuable during yoga practice, allowing users to request recommendations or feedback adjustments without manual input. Together, these front-end implementations ensure that Yoga Master delivers a cohesive, interactive, and personalized yoga experience.

7.2.3 Content-Based & Similarity based Recommendation

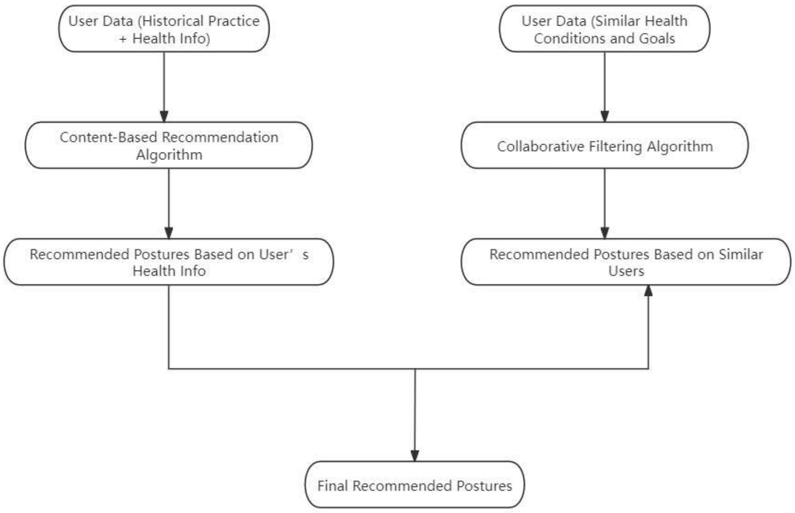

User Data (Historical Practice + Health Info)
User Data (Similar Health Conditions and Goals
Content-Based Recommendation Algorithm
Collaborative Filtering Algorithm
Recommended Postures Based on User’s Health Info
Recommended Postures Based on Similar Users
Final Recommended Postures

Figure8. Recommendation Algorithm

Feature Extraction: When extracting the user feature vector, corresponding processing is carried out for the physical condition based on the data stored in the user information database. Similarly, for the extraction of the yoga pose feature vector, targeted processing is also performed on factors such as the involved body parts, suitable populations, and training goals according to the data content in the yoga pose database.

Similarity Calculation: The cosine similarity function provided by the Scikit-learn library is utilized to complete the calculation of the similarity between the user feature vector and the yoga pose feature vector. During the specific implementation process, first, both the user feature vector and each of the yoga pose feature vectors are all converted into an appropriate vector form, and then they are substituted into the cosine similarity formula for calculation. The obtained similarity value will serve as a crucial reference for determining whether a yoga pose meets the user's needs. The higher the similarity, the higher the degree of matching between the yoga pose and the user's requirements.

Filtering: After initially screening out a series of potentially suitable yoga poses by means of cosine similarity, combined with the user's input physical condition information, especially the relevant content regarding the painful body parts, these already screened yoga poses are checked one by one. If a certain yoga pose involves the body part where the user has pain, then it will be removed from the recommendation list, so as to ensure that the final yoga poses recommended to the user will not further aggravate their physical discomfort.

7.2.4 Pose Detection and Key Point Extraction

When a user uploads a photo, the system first passes the image through YOLOv11 to identify key body points, such as the shoulders, elbows, hips, knees, and ankles.
These detected key points are then converted into coordinates for further analysis. OpenCV and NumPy are used to process and store these coordinates for subsequent computations.

7.2.5 Feature Calculation and Threshold Setting

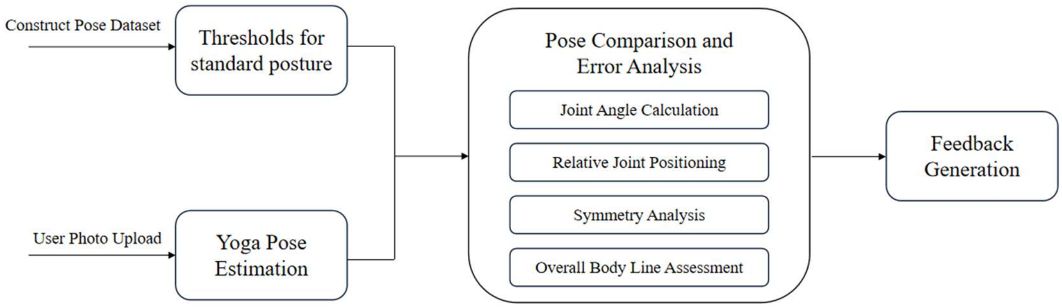

Construct Pose Dataset
Thresholds for standard posture
User Photo Upload
Yoga Pose Estimation
Pose Comparison and Error Analysis
Joint Angle Calculation
Relative Joint Positioning
Symmetry Analysis
Overall Body Line Assessment
Feedback Generation

Figure9. Threshold Calculation Workflow

Using the extracted coordinates, the system calculates relevant posture features, such as joint angles, relative joint positions, and symmetry between the left and right sides of the body.
Each feature is compared against a pre-defined threshold range, determined by analyzing standard yoga poses. Statistical analysis is performed on the data to establish a mean and standard deviation for each feature, allowing for tolerance ranges that account for individual differences.

7.2.6 Deviation Detection and Feedback Generation

If the user's posture deviates from the standard thresholds, the system uses the GLM-4-0520

language model to generate targeted feedback.
For example, if a significant angle deviation is detected in the left elbow, the model might produce feedback such as "Straighten your left elbow for better alignment with your shoulder." These instructions are designed to be actionable and specific, enabling users to correct their posture effectively.

7.2.7 Real-Time Feedback Display on Front-End

The feedback, including specific suggestions and visual cues, is displayed on a Vue.js-based interface. The user can see the feedback in real-time, allowing them to adjust their pose immediately.
The front-end also provides an option for users to view their progress over time, allowing for a more personalized and interactive experience.

7.2.8 Refine YOLO V11 Pose Estimation Model

We use a refined YOLO V11 model in our system. During the training process, we constructed a dataset containing yoga images and yoga pose keypoints. We used 200 different yoga pose images and manually annotated each yoga image using label tool - labelme, including labeling 17 key points of the human body and dividing the human body. We will divide the obtained dataset into a training set and a testing set, where the training set contains 120 images and the testing set contains 80 images. And we will convert the JSON files obtained through labelme into COCO and YOLO formats respectively to meet YOLO's training needs. After obtaining the dataset, we trained the yolo11m-pose. pt model in Colab, and the training process lasted for 100 epochs. After training, we obtained a refined YOLO model. Here are some of our training results.

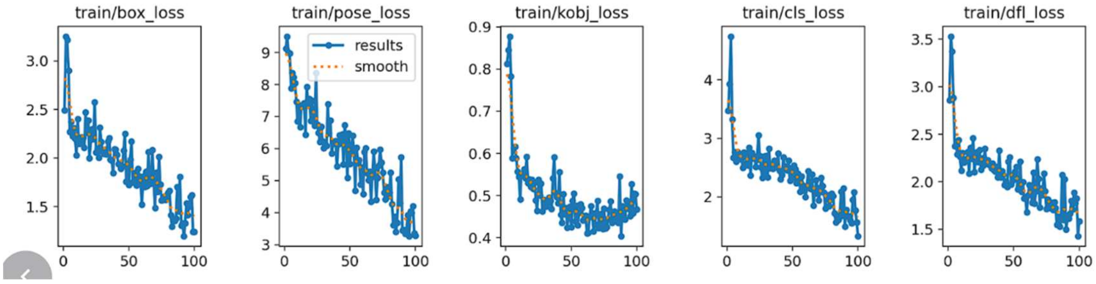


Figure10. YOLO Train Result

8. Findings and Discussion

1. System Performance and User Experience

Findings: The Yoga Master system effectively combines advanced AI technologies, including pose estimation, collaborative filtering, and real-time feedback, to create an engaging and personalized yoga experience. Users report a high level of satisfaction due to the accuracy of pose recommendations and the interactive nature of the feedback loop.

Discussion: By implementing keypoint detection and pose analysis with YOLOv11 and GLM-4-0520, the system achieves precise posture evaluation, helping users understand and improve their alignment. The integration of collaborative filtering ensures that the recommendations are not only personalized based on individual data but are also enhanced by insights from similar users, making them both relevant and adaptive.

2. Effectiveness of Real-Time Feedback

Findings: Real-time feedback has proven crucial for helping users make immediate adjustments, which prevents repetitive mistakes and reduces the risk of injury. The GLM-4-0520 model generates specific, actionable feedback, which users find highly valuable for improving posture accuracy.

Discussion: Real-time adjustments contribute to a deeper learning experience, as users can correct their poses without delay. This feedback loop reinforces proper technique, especially in an unsupervised home practice environment. In future iterations, expanding the feedback to include specific muscle engagement tips or breathing guidance may further enhance user outcomes.

3. User Interaction and Hands-Free Operation

Findings: The voice command functionality allows users to engage with the system without disrupting their practice, a feature that received positive feedback. Users found it convenient to request new poses, ask for feedback, and navigate between sessions through simple voice commands.

Discussion: The hands-free functionality enabled by the Web Speech API significantly improves the accessibility and usability of the app, especially for users who may have mobility constraints or prefer uninterrupted practice. As voice recognition technology continues to improve, expanding the system’s vocabulary for better accuracy and adding multi-language support could make the system even more versatile.

4. Limitations and Future Improvements

Findings: While the Yoga Master system performs well overall, certain limitations were identified. For instance, pose estimation accuracy may decrease in low-light environments, and voice recognition may sometimes struggle with background noise or varied accents.

Discussion: Addressing these limitations by enhancing image preprocessing for varying light conditions and optimizing voice recognition algorithms for different accents and noise conditions will improve robustness. Additionally, incorporating more diverse pose datasets and fine-tuning the model could support a broader range of yoga styles and levels. Future development could also explore integrating wearable devices to monitor additional metrics like heart rate or balance, providing users with even more comprehensive feedback on their yoga practice.

This section highlights the project’s strengths in real-time feedback, user engagement, and personalized recommendations while discussing areas for enhancement, aligning with future improvements to further elevate the user experience and accuracy of the Yoga Master system.

Appendix 1: Mapped System Functionalities Against Knowledge

Deep learning techniques:

In our system, we used a YOLO based pose estimation model. In pose estimation, YOLO detects multiple key-points within a single pass through the network. Each key-point represents a specific body part (like the shoulder or knee), and YOLO predicts the (x, y) coordinates for each key-point relative to an image grid. This approach allows YOLO to localize multiple key-points with high efficiency. For pose estimation, YOLO leverages a deep convolutional neural network backbone, such as Darknet, to learn multi-scale features essential for identifying body parts at different scales. By capturing features at multiple levels of detail, YOLO can handle poses in complex scenes, where body parts may vary in size or be partially occluded. YOLO's adaptability for pose estimation often involves customizing the loss function to consider both localization accuracy for key-points and classifying the presence of each keypoint correctly. Furthermore, YOLO models for pose estimation are sometimes trained on pose-specific datasets, such as COCO Key-points or MPII, to learn human body structures better. In our system we constructed a yoga pose dataset which combine both yoga pose pictures and labels to refine the YOLO model.

Unsupervised scenarios:

In your system, clustering yoga poses using an unsupervised approach serves two main purposes:

1. Recommendation of Yoga Poses: By grouping similar poses, the system can recommend yoga poses that align well with user preferences or skill levels. This clustering enables the system to find a variety of similar poses within each group, making recommendations more personalized and accurate.
2. Pose Comparison and Organization: Clustering allows for easy comparison between different types of yoga poses by categorizing them based on their similarity. This helps in analyzing pose variations, identifying key pose differences, and even developing structured sequences for training purposes.

Using unsupervised learning for clustering here is effective, as it enables the system to discover natural pose groupings based on key features (such as angles, joint positions, and body alignments) without predefined labels, giving flexibility for pose analysis and recommendation refinement.

Sense making techniques:

Sense-making involves collecting and analyzing data in a way that helps users understand and improve their actions—in this case, yoga poses. Here's how your system's approach achieves this:

1. Data Collection: By capturing a photo in 10 seconds, the system gathers regular data

points on the user's pose, offering a temporal view of their progress or deviations. This allows the system to capture subtle pose adjustments, helping users understand and refine their movements.

2. Automated Key-point Extraction and Classification: The system’s extraction of key-points and classification of the pose type allows it to generate meaningful insights into the user’s posture and alignment. This helps users quickly identify correct versus incorrect poses, aiding in real-time adjustment.

This automated approach not only captures relevant information but also interprets it, enhancing user experience by providing practical guidance for yoga practice.

Appendix 2: Installation & User Guide

Installation

1. Environment Requirement

pip install requirements

2. Run the system on local machine

Download the full zip file and Open terminal in its local fold file

```
$ cd yogamasterai-chat
$ npm run serve
$ cd yogamasterai-chat/backend
$ python app.py
```

Go to URL using web browser http://0.0.0.0:8080 or http://127.0.0.1:8080

User Guide

1. Login Page

The login interface provides a simple page for users to enter the system. Users can enter the system main interface by entering their username and password.

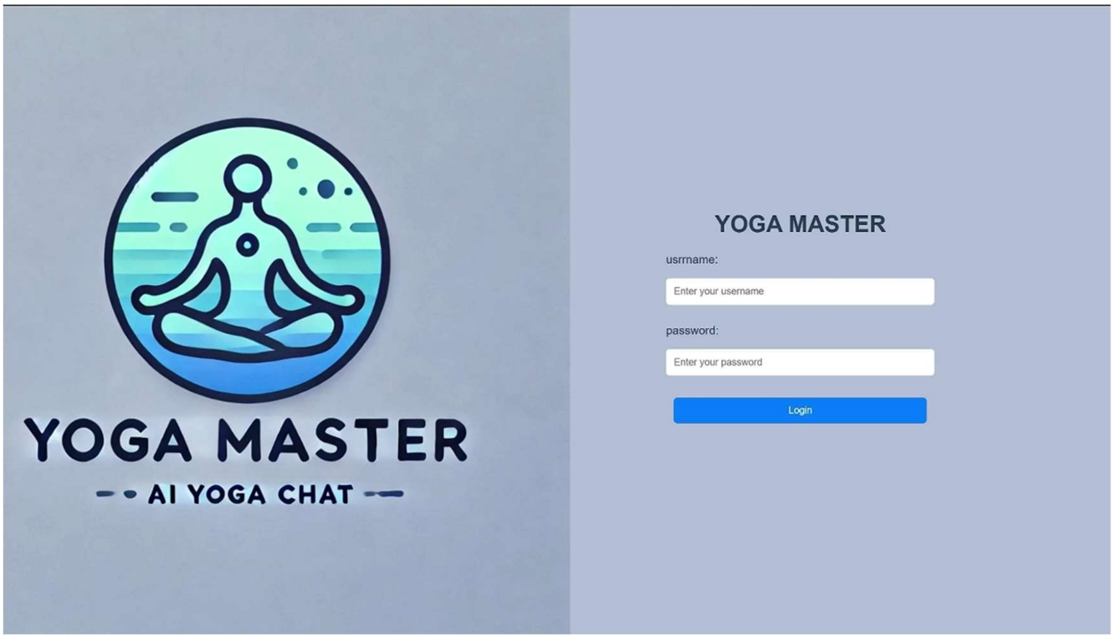
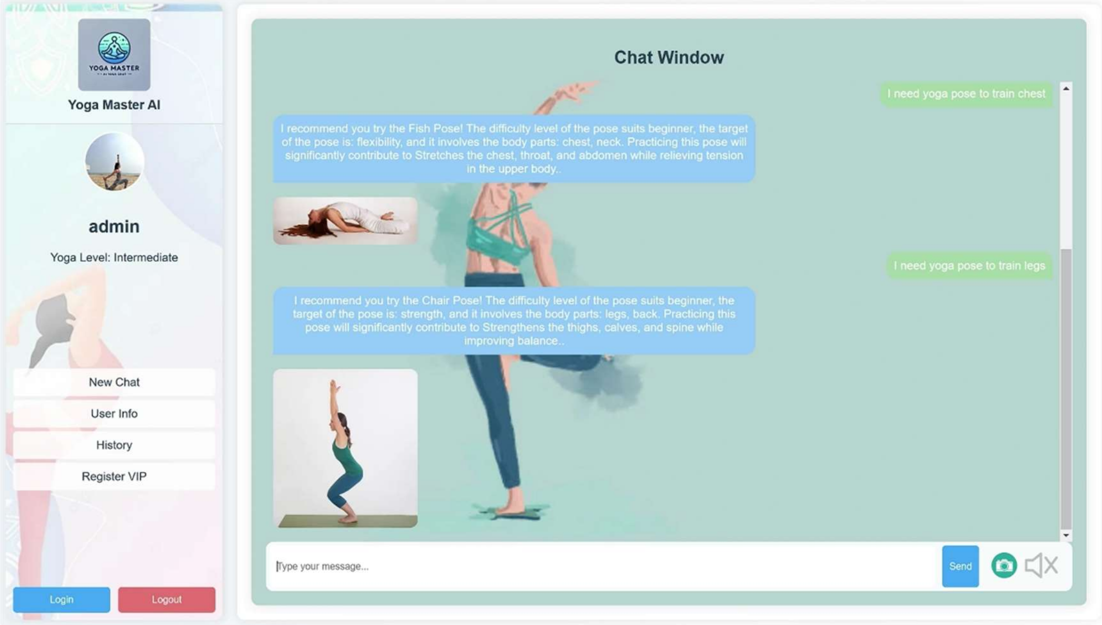


Figure 1: Login Page

2. Chat Window Page

The chat window provides users with a port to interact with the system, where they can input their requirements and the system will display the corresponding information in the chat window.



I need yoga pose to train chest

I recommend you try the Fish Pose! The difficulty level of the pose suits beginner, the target of the pose is: flexibility, and it involves the body parts: chest, neck. Practicing this pose will significantly contribute to Stretches the chest, throat, and abdomen while relieving tension in the upper body..

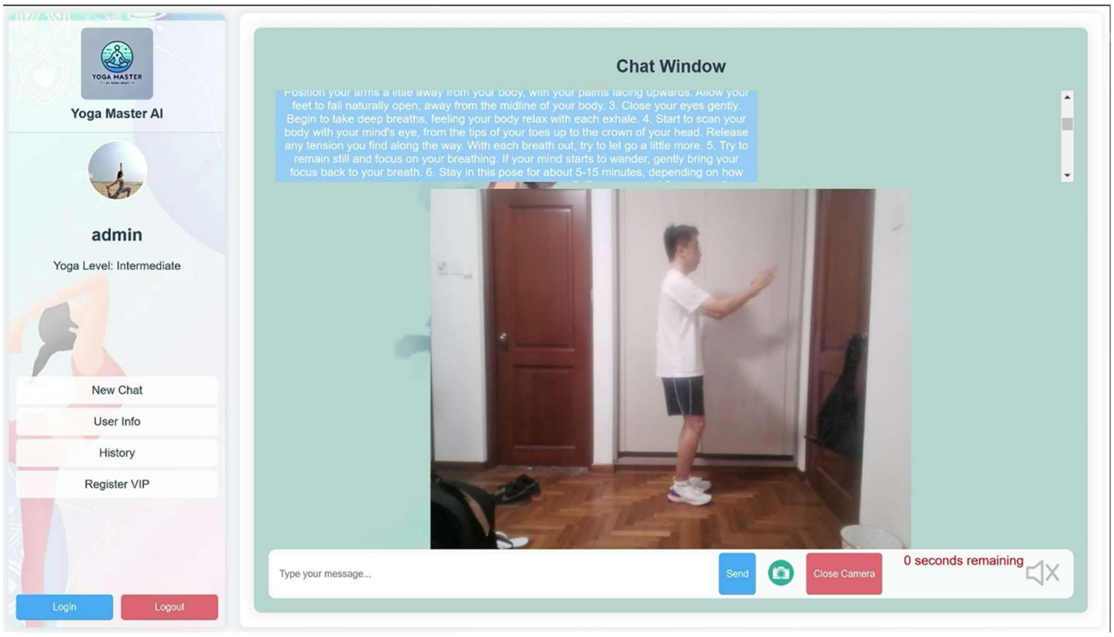
I need yoga pose to train legs

I recommend you try the Chair Pose! The difficulty level of the pose suits beginner, the target of the pose is: strength, and it involves the body parts: legs, back. Practicing this pose will significantly contribute to Strengthens the thighs, calves, and spine while improving balance..

Figure 2: ChatWindow Page

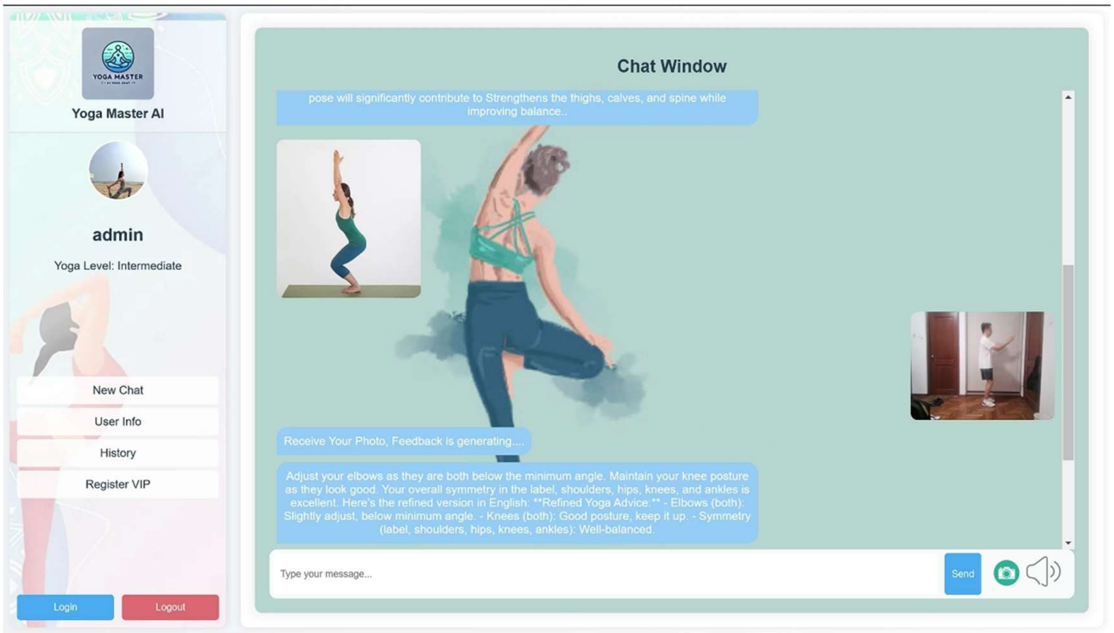
3. Photo Capturing and Feedback Generation

After clicking the camera button and selecting 'take photo', the system will automatically take a photo of the user's yoga pose and upload it to the backend. After comparing the user's posture with the standard posture, the system will output the correction suggestions in the chat window.



Position your arms a little away from your body, with your palms facing upwards. Allow your feet to fall naturally open, away from the midline of your body. 3. Close your eyes gently. Begin to take deep breaths, feeling your body relax with each exhale. 4. Start to scan your body with your mind's eye, from the tips of your toes up to the crown of your head. Release any tension you find along the way. With each breath out, try to let go a little more. 5. Try to remain still and focus on your breathing. If your mind starts to wander, gently bring your focus back to your breath. 6. Stay in this pose for about 5-15 minutes, depending on how

Figure 3: Photo Capturing



pose will significantly contribute to Strengthens the thighs, calves, and spine while improving balance..

Receive Your Photo, Feedback is generating....

Adjust your elbows as they are both below the minimum angle. Maintain your knee posture as they look good. Your overall symmetry in the label, shoulders, hips, knees, and ankles is excellent. Here's the refined version in English: **Refined Yoga Advice:** - Elbows (both): Slightly adjust, below minimum angle. - Knees (both): Good posture, keep it up. - Symmetry (label, shoulders, hips, knees, ankles): Well-balanced.

Figure 4: Feedback Output

4. Voice Input

Users can input voice information by clicking the speaker button, and the system will automatically convert the user's voice information into text information and send it to the chat window. The specific demonstration can be watched in the demonstration video.

5. Chat Side Bar

The chat side bar integrates three different functions. By clicking the New Chat button, you can start a new chat at any time. Click the User Info button to display the information of the currently logged in user. Click the Register VIP button to choose to register as a premium user and allow users to submit changes to their current user information in the form.