

Acknowledgment

I extend my heartfelt gratitude to **Prof. Hiten M Sadani** for their invaluable guidance and unwavering support throughout my Capstone Project, "AI Yoga Trainer." Their expertise, mentorship, and encouragement were crucial to the project's success. Prof Hiten Sadani's insightful advice, constructive feedback, and ability to simplify challenges greatly enhanced my learning experience. Their patience and dedication ensured the progress and quality of this work, contributing significantly to my technical and personal growth. Thank you for your inspiration and guidance, which have been instrumental in shaping this project and my future endeavors.

With Sincere Regards From
Mr. Harsh Vaghela
Ms. Kinnari Chaudhari
Mr. Yujit Kumar

Abstract

The AI Yoga Trainer is an innovative project aimed at helping users practice yoga accurately and effectively. Its primary objective is to provide real-time guidance on performing yoga poses correctly and to offer instant feedback on errors, enabling users to refine their techniques. The project consists of two key components. The first component delivers step-by-step instructions for various yoga poses, ensuring accessibility for users of all skill levels, including beginners. It uses clear, concise explanations and visual demonstrations to help users understand and practice poses with proper technique. The second component utilizes live camera input to track the user's movements during yoga sessions. By analyzing the user's body positions and comparing them with the correct poses, the system identifies inaccuracies and provides real-time notifications with suggestions for improvement. This feedback mechanism allows users to adjust their posture promptly, enhancing their practice and reducing the risk of errors over time. The AI Yoga Trainer combines instructional guidance with live performance tracking, creating a comprehensive solution for improving yoga practices and promoting physical well-being.

Table of Contents

1	INTRODUCTION.....	1
1.1	PROJECT OVERVIEW	1
1.2	BACKGROUND	1
1.3	PURPOSE.....	1
1.3.1	PROBLEM STATEMENT.....	1
1.3.2	PROJECT AIM.....	1
1.3.3	PROJECT OBJECTIVES	1
1.4	PROJECT SCOPE	2
1.5	IMPACT, SIGNIFICANCE, AND CONTRIBUTION.....	2
2	LITERATURE REVIEW	3
2.1	COMPUTER VISION AND POSE ESTIMATION	3
2.2	AI IN FITNESS AND HEALTH	3
2.3	YOGA TRAINER TECHNOLOGES.....	3
2.4	CHALLENGES IN REAL-TIME FEESBACK SYSTEMS	3
2.5	KEY STUDIES AND FINDINGS	4
2.6	GAPS IN EXISTING SYSTEMS.....	4
2.7	CONCLUSION OF LITERATURE REVIEW	4
2.8	REAL- TIME POSE DETECTION	4
3	SOFTWARE REQUIREMENTS (SRS)	5
3.1	FUNCTIONAL REQUIREMENTS.....	5
3.1.1	FUNCTIONAL REQUIREMENT FOR MODULE ADMIN	5
3.1.2	FUNCTIONAL REQUIREMENTS FOR MODULE USERS	5
3.2	NON- FUNCTIONAL REQUIREMENTS	6
4	SYSTEM REQUIREMENTS AND ANALYSIS	7
4.1	HARDWARE AND SOFTWARE REQUIREMENTS	7
4.1.1	HARDWARE REQUIREMENTS.....	7
4.1.2	SOFTWARE REQUIRMENTS.....	7
4.2	PROTOTYPE MODEL	8
5	SYSTEM ARCHITECTURE.....	10
5.1	FRONTEND	10
5.2	BACKEND.....	10
5.3	DATA LAYER.....	11
5.4	DATA FLOW AND INTERACTION.....	11
6	FEASIBILITY ANALYSIS	12
6.1	TECHNICAL FEASIBILITY	12

6.2	TIME SCHEDULE FEASIBILITY	12
6.3	OPERATIONAL FEASIBILITY	13
6.4	IMPLEMENTATION FEASIBILITY	13
6.5	ECONOMIC FEASIBILITY	13
7	PROJECT PLAN	15
8	DIAGRAMS	18
8.1	USE CASE DIAGRAM.....	18
8.2	ACTIVITY DIAGRAM	19
8.3	CLASS DIAGRAM	20
8.4	SEQUENCE DIAGRAM	21
8.5	DFD Diagrams	22
8.5.1	DFD Level 0.....	22
8.5.2	DFD Level 1	23
8.5.3	DFD level 2	24
9	DATA DICTIONARY	25
9.1	SIGN – UP PAGE USER.....	25
9.2	SIGN – UP PAGE ADMIN	25
10	IMPLEMENTATION DETAILS.....	26
10.1	ALGORITHM FOR POSE DETECTION	26
11	TESTING.....	32
12	USER MANUAL.....	34
12.1	INSTALLATION STEPS	34
12.2	FEATURES OVERVIEW	34
12.3	EXPLANATION	41
12.4	DEPLOYMENT STEPS	42
13	CONCLUSION AND FUTURE WORK.....	43
13.1	CONCLUSION	43
13.2	FUTURE WORK	43
	ANNEXURE	45
	GLOSSARY OF TERMS AND ABBREVIATION.....	45
	ABOUT TOOL AND TECHNOLOGY	45
	REFERENCES	46

LIST OF FIGURES

Figure 1 Use Case Diagram.....	18
Figure 2 Class Diagram.....	19
Figure 3 Activity Diagram.....	20
Figure 4 Sequence Diagram.....	21
Figure 5 DFD Level 0.....	22
Figure 6 DFD Level 1.....	23
Figure 7 DFD Level 2.....	24
Figure 8 Home page.....	34
Figure 9 Sign up page.....	35
Figure 10 Login Page.....	35
Figure 11 Home page Modules.....	36
Figure 12 Pose Detector Home Page.....	36
Figure 13 Pose Detection Interface.....	37
Figure 14 Pose Detection Modules	37
Figure 15 Pose Detection.....	38
Figure 16 Pose Detection About Page.....	38
Figure 17 Pose Learner options	39
Figure 18 Pose Learner.....	39
Figure 19 Cardio Interface	40
Figure 20 Meditation Interface.....	40
Figure 21 Mental Health Queries.....	41

1 INTRODUCTION

The AI Yoga Trainer project aims to assist individuals in practicing yoga safely and effectively. By providing easy-to-understand instructions and visual examples for various yoga poses, it ensures users can learn and improve their technique. A live camera monitors users as they perform yoga poses, providing real-time feedback to correct errors and enhance posture. Designed for all skill levels, from beginners to experienced practitioners, the system leverages AI to make yoga practice safer, more accessible, and more effective.

1.1 PROJECT OVERVIEW

The AI Yoga Trainer is an innovative solution that combines AI-powered computer vision with user-friendly functionality. It delivers step-by-step guidance, visual demonstrations, and real-time feedback to help users perfect their yoga poses. By catering to a wide range of users, the system promotes engagement, ensures safety, and enhances the overall yoga experience.

1.2 BACKGROUND

Yoga is widely recognized for its physical and mental health benefits. However, incorrect postures and techniques can diminish its effectiveness and increase the likelihood of injuries. Traditional yoga training often requires in-person sessions with instructors, which may not be feasible due to cost or availability. The AI Yoga Trainer bridges this gap by offering an affordable and accessible virtual yoga assistant powered by AI.

1.3 PURPOSE

1.3.1 PROBLEM STATEMENT

Practicing yoga without proper guidance often leads to incorrect postures, reducing its benefits and increasing the risk of injuries. Additionally, the lack of access to qualified instructors due to logistical or financial barriers creates the need for a reliable and affordable alternative.

1.3.2 PROJECT AIM

To develop an AI-driven system that ensures safe and effective yoga practice by providing precise instructions and real-time posture correction through live camera monitoring.

1.3.3 PROJECT OBJECTIVES

Demonstrate yoga poses with clear, step-by-step instructions and visual aids.
Implement real-time body posture tracking using computer vision techniques.
Provide instant feedback for correcting form and improving posture.

Minimize the risk of injuries through accurate guidance.

Create a user-friendly interface suitable for users of all skill levels.

1.4 PROJECT SCOPE

The AI Yoga Trainer includes the following modules:

Yoga Pose Instruction: Offers detailed explanations and visual demonstrations of poses.

Real-Time Posture Monitoring: Tracks user movements to ensure correct alignment using computer vision.

Instant Feedback: Provides live corrections and suggestions for improvement.

Future Scope:

Progress tracking to measure user improvement over time.

Personalized yoga routines based on user needs and goals.

Support for additional yoga poses and advanced training modules..

1.5 IMPACT, SIGNIFICANCE, AND CONTRIBUTION

The AI Yoga Trainer revolutionizes yoga practice by integrating AI and computer vision, making yoga accessible and safe for a wider audience. It reduces the dependency on physical instructors, minimizes injury risks, and enhances the overall effectiveness of yoga sessions. By promoting consistent practice and improving accessibility, the system contributes to physical and mental well-being. Its scalable design allows for continuous innovation, ensuring relevance in the evolving fitness industry.

2 LITERATURE REVIEW

This section explores existing literature and technologies foundational to the development of the AI Yoga Trainer Website, emphasizing advancements in computer vision, pose estimation, and AI's role in fitness and health.

2.1 COMPUTER VISION AND POSE ESTIMATION

Computer vision enables AI systems to process and interpret visual data.

Pose estimation, a subset of computer vision, focuses on identifying key points on the human body to analyze movement and posture.

Frameworks:

OpenPose, MediaPipe, and TensorFlow's PoseNet are widely used for real-time and accurate pose estimation.

Applications:

Commonly used in fields like gaming, healthcare, and sports for enhancing performance and ensuring safety..

2.2 AI IN FITNESS AND HEALTH

Integration of AI:

AI technologies provide real-time feedback, tailored recommendations, and progress tracking in fitness.

Applications:

AI has proven effective in personal training, rehabilitation, and injury prevention.

Benefits:

Enhances user engagement and adherence to fitness programs through interactive and intelligent systems.

2.3 YOGA TRAINER TECHNOLOGES

Traditional Methods:

Yoga instruction traditionally relies on in-person guidance, which may not be accessible or affordable.

Digital Solutions:

Yoga apps and prerecorded videos increase accessibility but lack real-time correction features.

Recent Research:

Studies indicate that combining pose estimation with corrective feedback significantly enhances safety and performance in yoga.

2.4 CHALLENGES IN REAL-TIME FEESBACK SYSTEMS

Pose Estimation Accuracy: Varying lighting, clothing, and background conditions can affect pose detection.

Feedback Design: Feedback needs to be intuitive and actionable without overwhelming the user.

Solutions: Robust algorithms and diverse datasets can mitigate accuracy issues.

2.5 KEY STUDIES AND FINDINGS

PoseNet:

Demonstrated effectiveness in detecting 17 body key points in real-time, making it ideal for fitness applications.

AI for Posture Correction:

Highlights AI's role in detecting posture errors and reducing injury risks.

User Preferences:

Real-time guidance significantly improves user satisfaction compared to prerecorded tutorials.

2.6 GAPS IN EXISTING SYSTEMS

Current Limitations:

Real-time correction accuracy, user customization, and scalability remain challenges.

Dynamic poses requiring precise alignment and flexibility are often inadequately addressed.

Opportunities:

The AI Yoga Trainer Website seeks to address these gaps through innovative solutions and robust AI integrations.

2.7 CONCLUSION OF LITERATURE REVIEW

The literature emphasizes the transformative potential of integrating AI and computer vision into yoga training. However, innovation is needed in pose detection accuracy, user experience, and system adaptability. The AI Yoga Trainer Website builds on these findings to deliver a comprehensive, accessible, and effective yoga training solution.

2.8 REAL- TIME POSE DETECTION

Description: A live camera tracks users' yoga poses, compares them with the correct form, and provides real-time feedback.

Acceptance Criteria:

The system detects body movements during yoga via the camera.

Incorrect poses trigger immediate alerts with improvement suggestions.

Feedback is clear, concise, and user-friendly.

Technology Stack:

Frontend: HTML, CSS, JavaScript.

Backend: OpenCV, TensorFlow.

Hardware: Web Camera.

3 SOFTWARE REQUIREMENTS (SRS)

3.1 FUNCTIONAL REQUIREMENTS

The Admin module ensures backend operations are efficient and scalable. Key functionalities include

3.1.1 FUNCTIONAL REQUIREMENT FOR MODULE ADMIN

The Admin module ensures backend operations are efficient and scalable. Key functionalities include:

Yoga Pose Management:

Add, update, or remove yoga poses with instructions and visual examples.

User Management:

Manage user accounts, including creating, updating, or deleting profiles.

System Monitoring:

Monitor system performance and logs to ensure smooth user experiences.

Feedback Management:

Review user feedback and implement suggestions for feature improvements.

AI Model Updates:

Update or retrain AI models to enhance pose detection accuracy and performance.

3.1.2 FUNCTIONAL REQUIREMENTS FOR MODULE USERS

The User module provides intuitive interaction for end-users. Key functionalities include:

Pose Instructions:

View detailed, step-by-step instructions and visual demonstrations for yoga poses.

Live Pose Tracking:

Perform yoga poses with a live camera feed, receiving real-time posture and alignment feedback.

Error Notifications:

Receive instant alerts with specific corrections if a pose is performed incorrectly.

Pose Selection:

Choose yoga poses based on difficulty, goals (e.g., flexibility, strength), or duration.

Personalized Routines:

Create and save personalized yoga routines tailored to individual preferences.

3.2 NON- FUNCTIONAL REQUIREMENTS

The system's non-functional requirements ensure its performance, usability, and security meet user expectations:

Performance:

Provide real-time feedback with minimal delay (response time <1 second).

Accuracy:

Maintain a pose detection accuracy threshold of at least 90% to ensure effective feedback.

Usability:

Offer an intuitive, user-friendly interface for all skill levels, including beginners.

Scalability:

Support multiple users simultaneously without performance degradation.

Security:

Protect user data, including video feeds and preferences, using secure storage and encrypted communication protocols.

Compatibility:

Ensure accessibility via web browsers and compatibility with major operating systems (Windows, macOS, Android, iOS).

Maintainability:

Facilitate easy updates to AI models and the user interface without disrupting functionality.

Reliability:

Achieve an uptime of at least 99%, ensuring high system availability.

Localization:

Support multiple languages to cater to a diverse user base.

4 SYSTEM REQUIREMENTS AND ANALYSIS

This section details the hardware and software requirements for the AI Yoga Trainer, the development process model, and the prototype model that guides the system's design and implementation.

4.1 HARDWARE AND SOFTWARE REQUIREMENTS

4.1.1 HARDWARE REQUIREMENTS

User Side

Device Requirements:

A device equipped with a camera (Laptop, Smartphone, or Desktop with a webcam).

Minimum Specifications:

Processor: Intel i3 or equivalent.

RAM: 4 GB.

Storage: 2 GB of free space for temporary files.

Internet:

Required for cloud-based operations (optional for local execution).

Server Side (for cloud-based deployment)

Processor: Intel Xeon or equivalent.

RAM: 16 GB or higher.

Storage: 500 GB or more for logs, pose data, and user information.

GPU: NVIDIA GPU with CUDA support for AI model processing.

4.1.2 SOFTWARE REQUIREMENTS

User Side

Web Browser:

Supported browsers include Chrome, Firefox, Safari, or Edge.

Operating Systems:

Compatible with Windows, macOS, Android, or iOS.

Server Side**Operating System:**

Ubuntu 20.04 or later (Linux).

Programming Languages:

Python for AI and backend logic.

JavaScript for frontend development.

Libraries and Frameworks:

AI: TensorFlow or PyTorch for pose detection and model training.

Pose Detection: OpenCV or MediaPipe for real-time keypoint tracking.

Backend: Flask or Django for API development.

Frontend: React or Angular for interactive UI/UX.

Database:

PostgreSQL or MongoDB for storing user and pose data.

4.2 PROTOTYPE MODEL

The prototype model helps visualize and refine system functionality before full-scale development. It focuses on core features and user interactions to ensure alignment with project goals.

Key Features of the Prototype**Pose Demonstration Module:**

Displays step-by-step instructions and animations for yoga poses.

Live Camera Integration:

Captures user movements and detects key body points in real-time.

Basic Feedback System:

Alerts users about misalignments in simple yoga poses with corrective suggestions.

Prototype Development Phases**Initial Design:**

Build a minimal user interface showcasing yoga pose categories and descriptions.

Pose Detection Integration:

Implement pose estimation using MediaPipe or PoseNet to track body key points.

Feedback System:

Develop comparison logic to evaluate user poses against stored templates, providing basic corrections.

User Testing:

Gather feedback on ease of use and pose detection accuracy to refine the prototype.

5 SYSTEM ARCHITECTURE

5.1 FRONTEND

The frontend is the user-facing part of the system, built using web technologies to ensure accessibility and responsiveness.

Technologies:

HTML, CSS, JavaScript.

Key Components:

Sign-Up/Sign-In Pages:

Collect user information and authenticate them.

Yoga Pose Demonstration:

Displays videos, GIFs, or animations to guide users on proper yoga poses.

Live Camera Feed:

Integrates a live webcam feed for real-time pose monitoring.

Error/Feedback Messages:

Provides visual or textual feedback if a user's pose is incorrect, guiding improvements.

5.2 BACKEND

- The backend manages core functionalities like AI model processing, user data handling, and API integrations.
- AI/ML Processing: Python with libraries like TensorFlow, PoseNet, or MediaPipe.
- Server-Side Logic: Node.js/Express for efficient API handling.
- Database: MySQL for storing and managing data.
- **Key Components:**
- **API Endpoints:**
- Facilitate communication between the frontend and backend, such as:
- User Authentication: For sign-up and login.
- Yoga Pose Data: Fetching pose instructions and media.
- User Data Management: Handling user profile and preferences.
- Session Data Management: Storing session metrics like accuracy and feedback.
- AI/ML Processing:
- Analyzes live camera feed data to compare user poses with pre-trained yoga models.

- Generates posture accuracy metrics and feedback.
- Feedback Generation:
- Backend logic generates corrective suggestions or alerts based on AI analysis. Based on this analysis, it will give feedback on whether the user is performing the pose correctly.

5.3 DATA LAYER

The data layer is responsible for securely storing all system information, including user profiles, yoga poses, and session logs.

Technologies:

MySQL (relational database).

Key Components:

User Data:

Stores user information such as username, email, weight, and yoga goals.

Yoga Pose Data:

Includes pose names, difficulty levels, demo video URLs, and alignment templates.

Session Data:

Logs each yoga session's details, including pose accuracy, start/end times, and feedback provided.

5.4 DATA FLOW AND INTERACTION

- **User Interaction:**

The data flow illustrates how different components interact to deliver functionalities to the user:

- **User Interaction:**

Users interact with the frontend to log in, select yoga poses, and start sessions.

- **API Requests:**

The frontend sends API requests to the backend for authentication, pose data, and session initiation.

- **AI Processing:**

The backend processes the live camera feed using AI/ML models to assess the user's posture.

- **Data Storage:**

Session details, including accuracy and feedback logs, are stored in the MySQL database.

- **Feedback Delivery:**

Real-time feedback is sent from the backend to the frontend, where it is displayed to the user.

6 FEASIBILITY ANALYSIS

The feasibility analysis evaluates the practicality and achievability of the AI Yoga Trainer Website project across multiple dimensions, ensuring successful implementation and long-term sustainability.

6.1 TECHNICAL FEASIBILITY

The project is technically feasible due to the following factors:

Pose Detection:

Pre-trained models like MediaPipe, OpenPose, and PoseNet ensure accurate real-time body tracking and pose estimation.

AI Integration:

Open-source libraries such as TensorFlow and PyTorch facilitate efficient AI model customization and training.

System Scalability:

Cloud platforms (e.g., AWS, Azure, Google Cloud) provide scalable infrastructure for real-time analysis and user management.

Hardware Compatibility:

The system requires only basic devices (smartphones or laptops with cameras), ensuring broad accessibility and minimal hardware constraints.

6.2 TIME SCHEDULE FEASIBILITY

The timeline is based on Agile methodology, promoting iterative development and timely delivery:

Requirement Analysis: 2 weeks.

Design Phase: 2 weeks.

Development: 8–10 weeks:

Pose instruction module: 3 weeks.

Real-time tracking and feedback: 5 weeks.

User interface: 2 weeks.

Testing and Debugging: 3–4 weeks.

Deployment and Feedback Iteration: 2 weeks.

Conclusion: The project is feasible within a 4–5-month timeline, given a dedicated team and an efficient development process.

6.3 OPERATIONAL FEASIBILITY

The AI Yoga Trainer is designed to be user-friendly and accessible, ensuring smooth operation:

User Experience:

Intuitive navigation, clear instructions, and engaging visuals cater to users of all skill levels.

Error Handling:

Real-time feedback and pose correction reduce the risk of incorrect usage, fostering trust and reliability.

Maintenance:

Regular updates to AI models and routine system checks ensure consistent performance and reliability.

6.4 IMPLEMENTATION FEASIBILITY

The modular design and widely available tools make implementation feasible:

Phase-Wise Deployment:

Begin with basic pose instruction and feedback features, expanding functionality incrementally.

Flexible Hosting:

Deploy on cloud infrastructure (AWS, Azure) or local servers based on requirements.

Team Expertise:

A skilled team in AI, computer vision, and web development ensures challenges are effectively managed.

6.5 ECONOMIC FEASIBILITY

- The project is economically feasible due to the low-cost development and potential for revenue generation
- **Development Costs:**
 - Software Tools and Frameworks:** Largely open-source, minimizing initial expenses.
 - Hardware:** Basic infrastructure with optional GPU for AI training.
- **Personnel:** Salaries for developers, designers, and testers.
- **Operational Costs:**
 - Cloud hosting and server maintenance.

Periodic updates and support services.

- **Revenue Opportunities:**

Subscription-based premium features (e.g., personalized routines, advanced analytics).

Partnerships with yoga studios or fitness brands for marketing and sponsorship..

7 PROJECT PLAN

Requirement Analysis:

Description: Gather and analyze project requirements, including functional and non-functional requirements.

Duration: 2 weeks

Design Phase:

Description: Develop wireframes, UI/UX designs, and system architecture.

Duration: 2 weeks

Development:

Description: Build the system's core features and functionalities in iterative sprints.

Duration: 8–10 weeks

Pose Instruction Module: Develop the yoga pose demonstration and selection interface. (3 weeks)

Real-Time Tracking & Feedback: Implement pose estimation and feedback systems using AI models. (5 weeks)

User Interface: Integrate all modules into a seamless and intuitive frontend interface. (2 weeks)

Testing and Debugging

Description: Conduct rigorous testing to identify and fix bugs, ensuring system reliability.

Duration: 3–4 weeks

Deployment:

Description: Deploy the system on cloud/local servers, conduct final testing, and go live.

Duration: 1–2 weeks

Feedback Iteration:

Description: Collect user feedback post-deployment and incorporate improvements.

Duration: 1–2 weeks.

7.2 Agile Methodology Breakdown

Sprints:

Each phase (design, development, testing, deployment) is divided into 2-week sprints.

Daily Standups:

Regular team meetings ensure progress tracking and quick resolution of roadblocks.

Backlog Management:

Prioritize tasks in the product backlog to focus on high-impact features first.

Iterative Feedback:

Gather feedback from stakeholders after each sprint to refine features and ensure alignment with goals.

7.3 Milestones

Requirement Finalization:

Deliverables: Project requirements document, user stories.

Design Approval:

Deliverables: Wireframes, UI/UX designs, system architecture diagrams.

Module Completion:

Deliverables: Fully functional pose instruction, tracking, and feedback modules.

System Integration:

Deliverables: Integrated and tested application ready for deployment.

Go-Live:

Deliverables: Deployed system with initial user feedback collected.

7.4 Resource Allocation

Team Roles:

Project Manager: Oversees progress and ensures timely delivery.

Frontend Developer: Designs and implements the user interface.

Backend Developer: Develops API endpoints and integrates AI/ML models.

AI Specialist: Customizes and trains pose estimation models.

QA Engineer: Conducts testing and ensures quality.

DevOps Engineer: Handles deployment and server management.

Tools and Technologies:

Collaboration: Jira, Trello, Slack for task management and communication.

Development: TensorFlow, React, Flask/Django, MySQL.

Testing: Selenium, Postman for automated testing and API validation.

7.5 Risk Management

Delays in Development:

Mitigation: Use Agile sprints to deliver incremental progress and adjust timelines if necessary.

Pose Detection Accuracy Issues:

Mitigation: Train AI models on diverse datasets and perform extensive testing.

User Adoption Challenges:

Mitigation: Design an intuitive UI/UX and provide clear instructions for beginners.

Scaling and Performance:

Mitigation: Leverage cloud infrastructure to scale resources as needed.

8 DIAGRAMS

8.1 USE CASE DIAGRAM

The Use Case Diagram for the “Yoga AI Trainer” outlines the interactions between two primary actors: the User and the Admin. Users can log in, watch yoga videos, participate in live sessions for real-time guidance, and log out. Admins have additional responsibilities, such as managing user accounts, application data, and website content. This diagram provides a clear overview of the system's functionality, highlighting the distinct roles and interactions within the Yoga AI Trainer application.

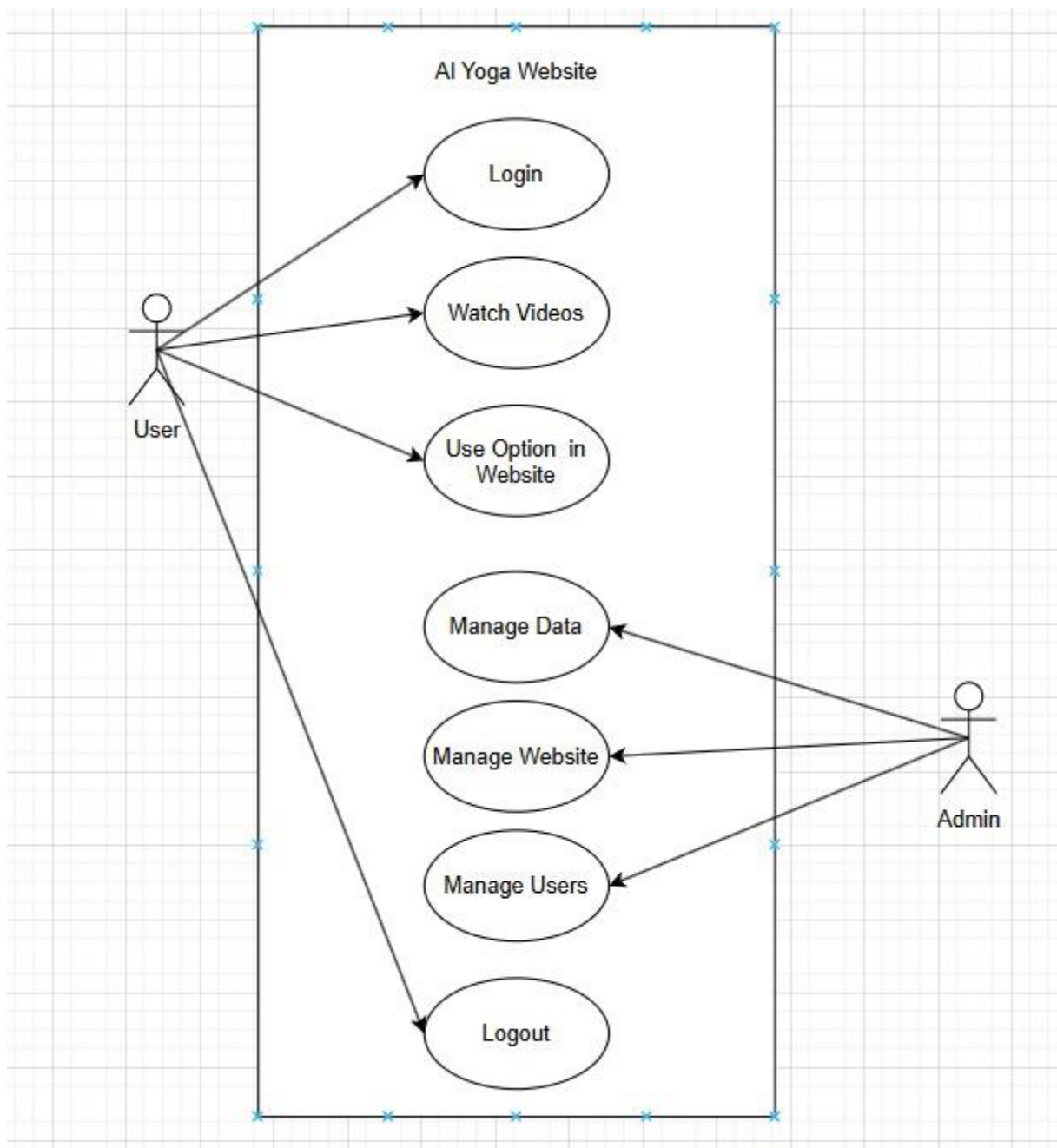


Figure 1 Use Case diagram

8.2 ACTIVITY DIAGRAM

The Activity Diagram for the “**Yoga AI Trainer**” illustrates the interaction between the User and the System. The process starts with user registration or login, where credentials are verified. After selecting a yoga pose, the system provides instructions, captures the user's pose, and analyzes it for accuracy. Feedback is given for corrections, session data is saved, progress is tracked, and users can either view their progress or log out.

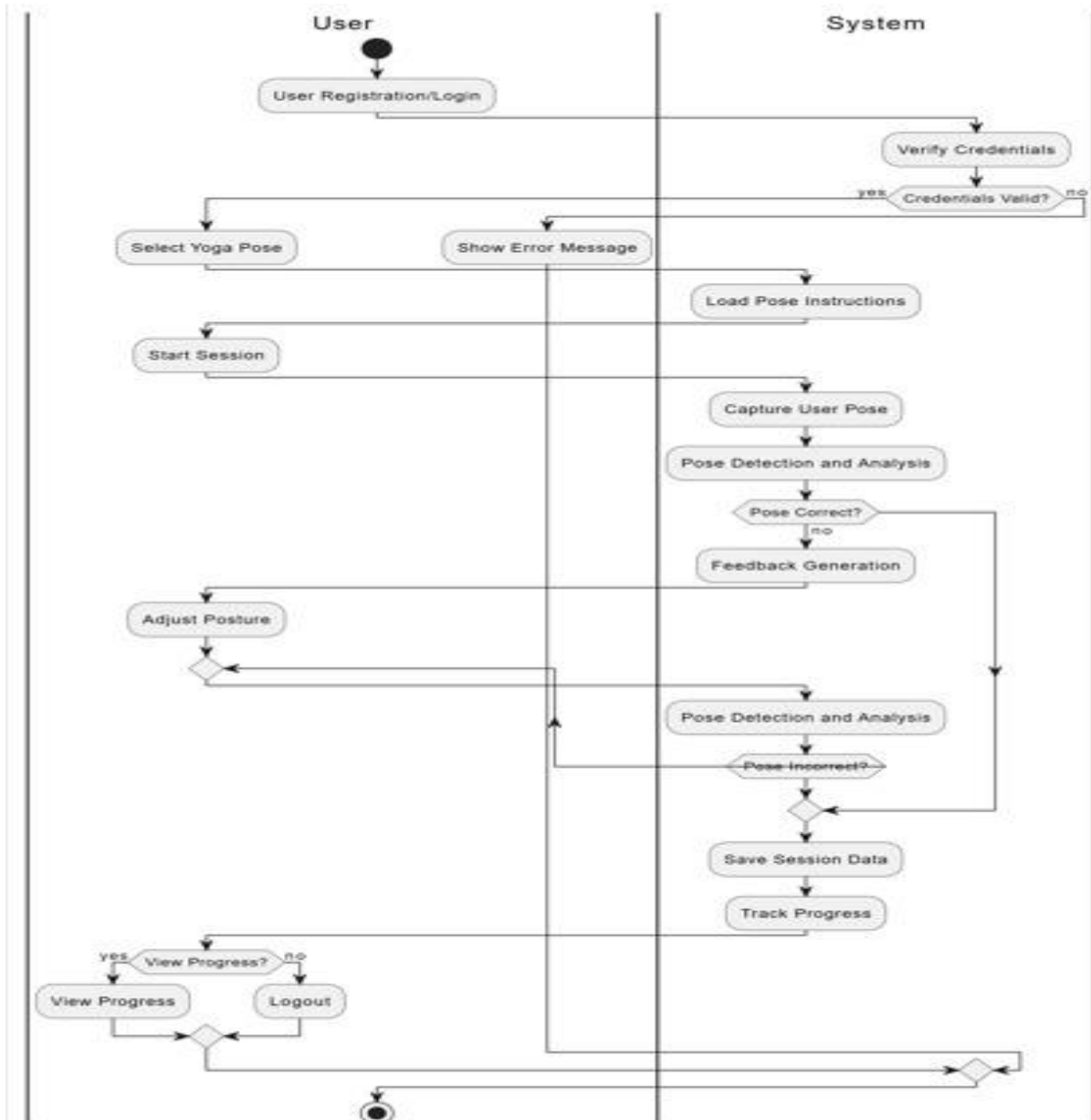


Figure 2: Activity Diagram

8.3 CLASS DIAGRAM

This Class Diagram for the “Yoga AI Trainer” depicts the structure and relationships between system components. The Admin manages users and yoga poses, while the User can register, log in, and interact with poses. The Pose class stores information about each yoga pose, and the Feedback class generates correction messages for users based on real-time analysis. The System handles operations like initialization, real-time tracking, and pose correction, ensuring smooth functionality.

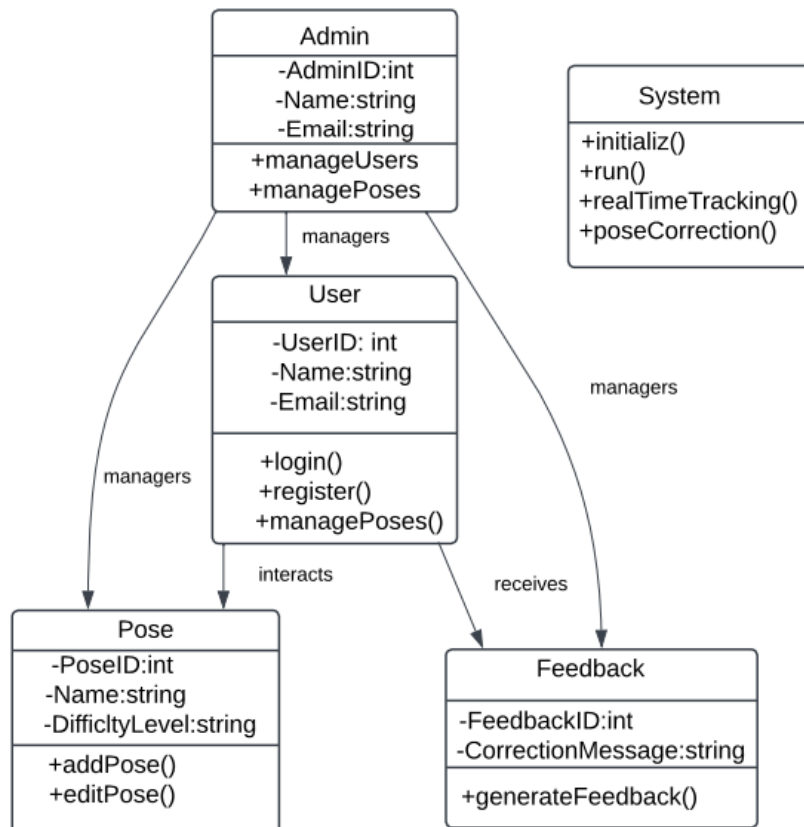


Figure 3: Class Diagram

8.4 SEQUENCE DIAGRAM

The sequence diagram illustrates the flow of an AI-powered yoga trainer system, encompassing both user and admin interactions. Users can sign up or log in, with the system validating their credentials against the database. Once logged in successfully, users can access a list of yoga videos retrieved from the database, select a video, and stream it seamlessly. Admins have a dedicated panel where they can log in to manage the platform's content and users. They can add, update, or delete yoga videos and modify user details, with all actions validated by the system and synchronized with the database. The system ensures smooth interactions, secure validation, and real-time updates for both users and administrators.

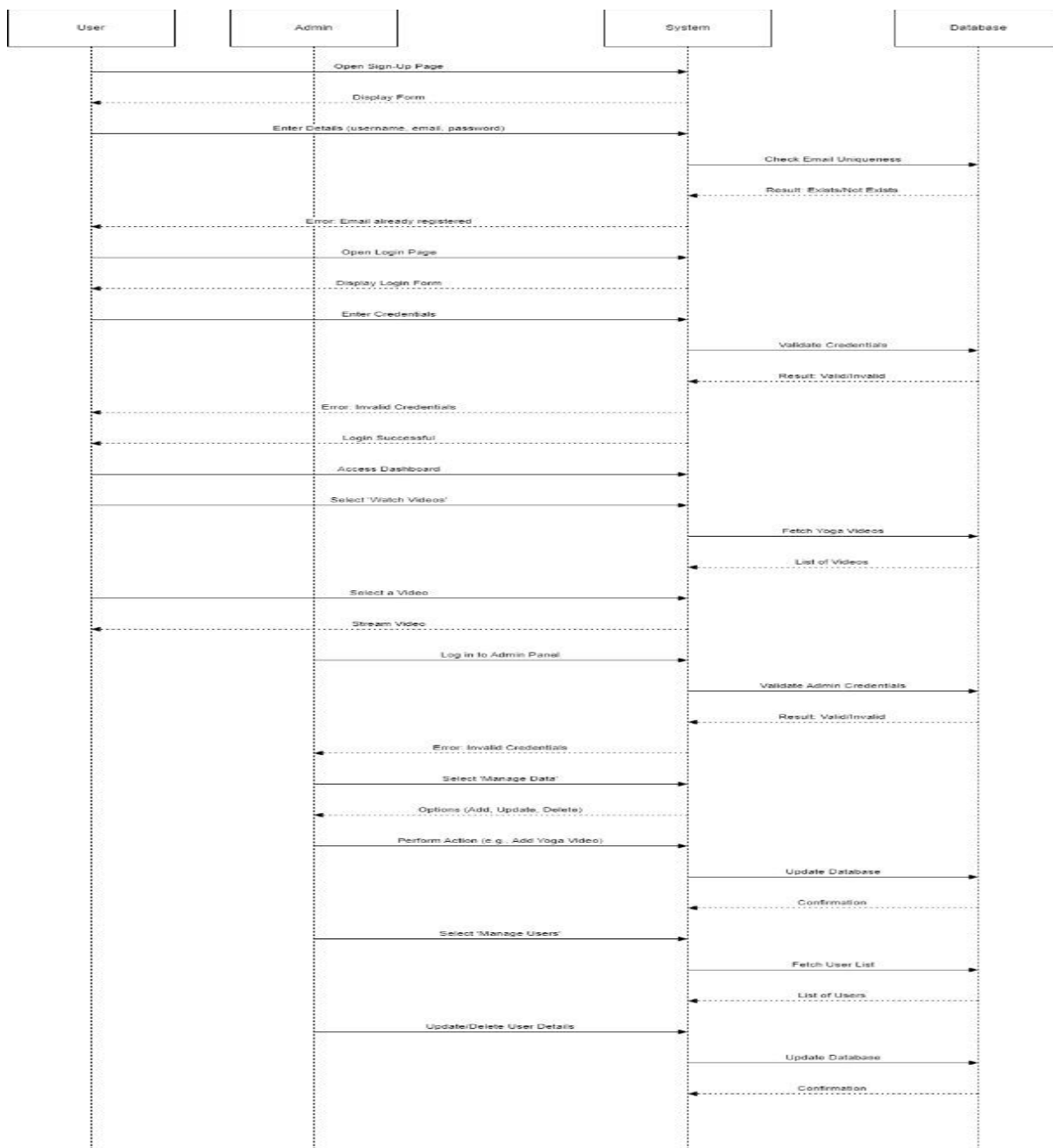


Figure 4: Sequence Diagram

8.5 DFD Diagrams

8.5.1 DFD Level 0

The diagram represents the core functionalities of the Zen Yoga system. Users can interact with the platform to watch pre-recorded yoga videos or join live sessions through the "Go Live" feature. Additionally, administrators or authorized users can manage the system, including content and user data. The Zen Yoga system serves as a centralized hub, seamlessly connecting these functionalities for an enhanced yoga experience.

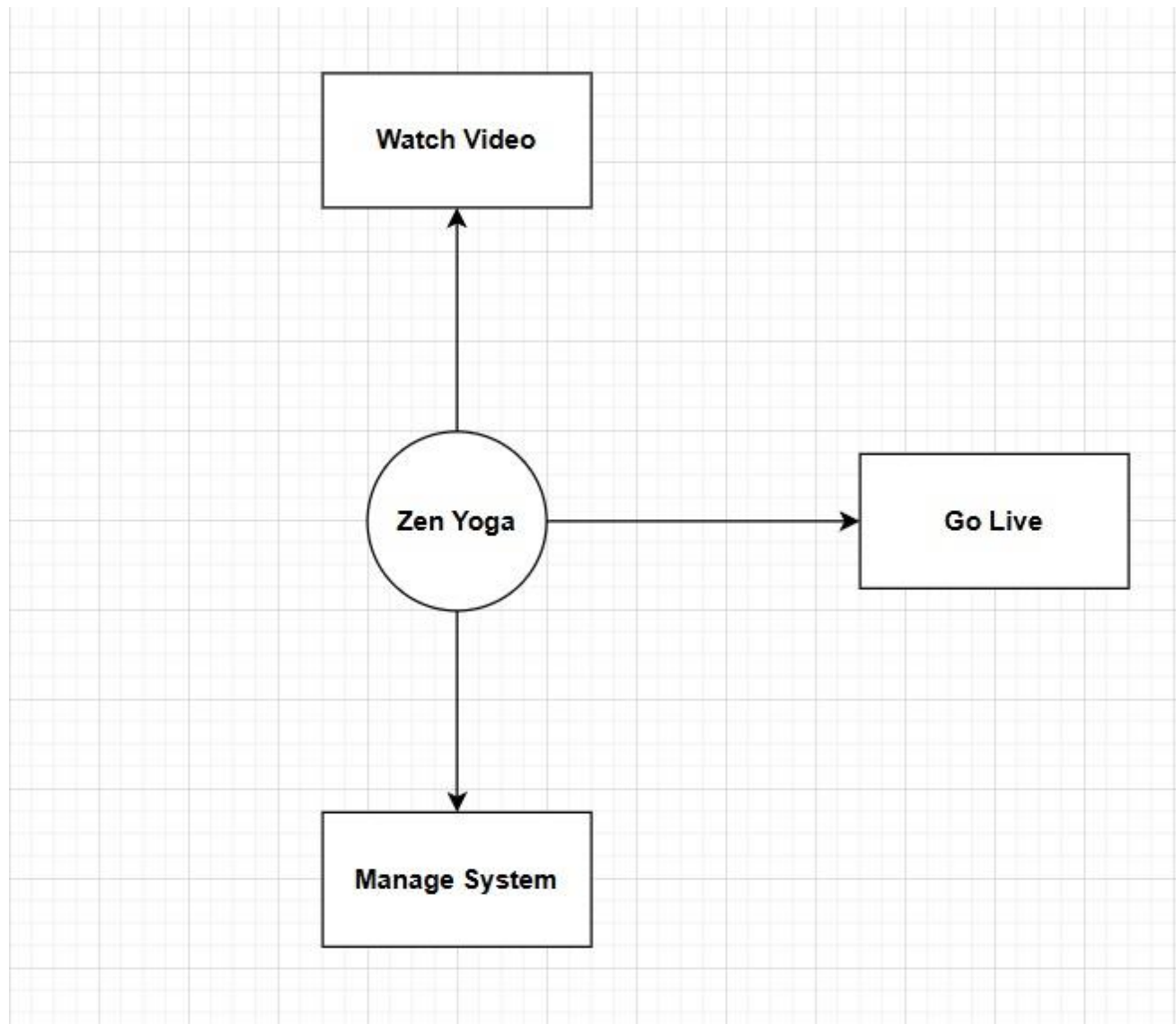


Figure 5 DFD Level 0

8.5.2 DFD Level 1

The diagram showcases the functional architecture of the Zen Yoga system. Users can access features like watching yoga videos or participating in live sessions, both facilitated by the core Zen Yoga platform. Administrators manage the system through functionalities such as user management, website maintenance, and data handling, ensuring seamless operations. The system connects to a user database to store and retrieve user-related data, creating an integrated platform for both users and administrators.

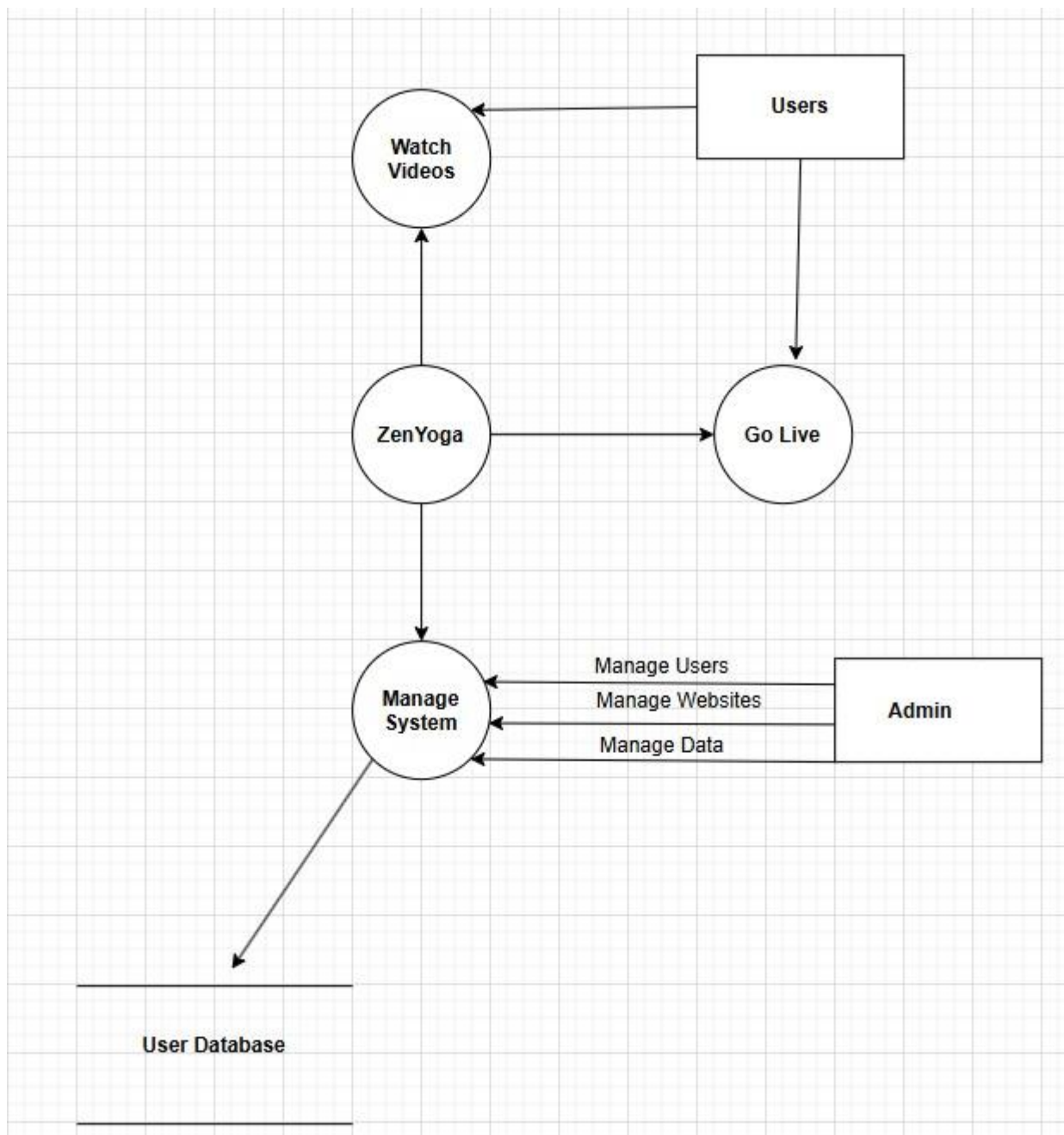


Figure 6 DFD Level 1

8.5.3 DFD level 2

The diagram outlines a yoga platform's architecture. Users can sign up, log in, and access various modules, including watching videos, live yoga sessions, pose correction, and a questionnaire, with a "Forget Password" feature for account recovery. Admins oversee user management, data handling, and website maintenance to ensure smooth platform functionality. The system enables a seamless user experience with robust administrative control.

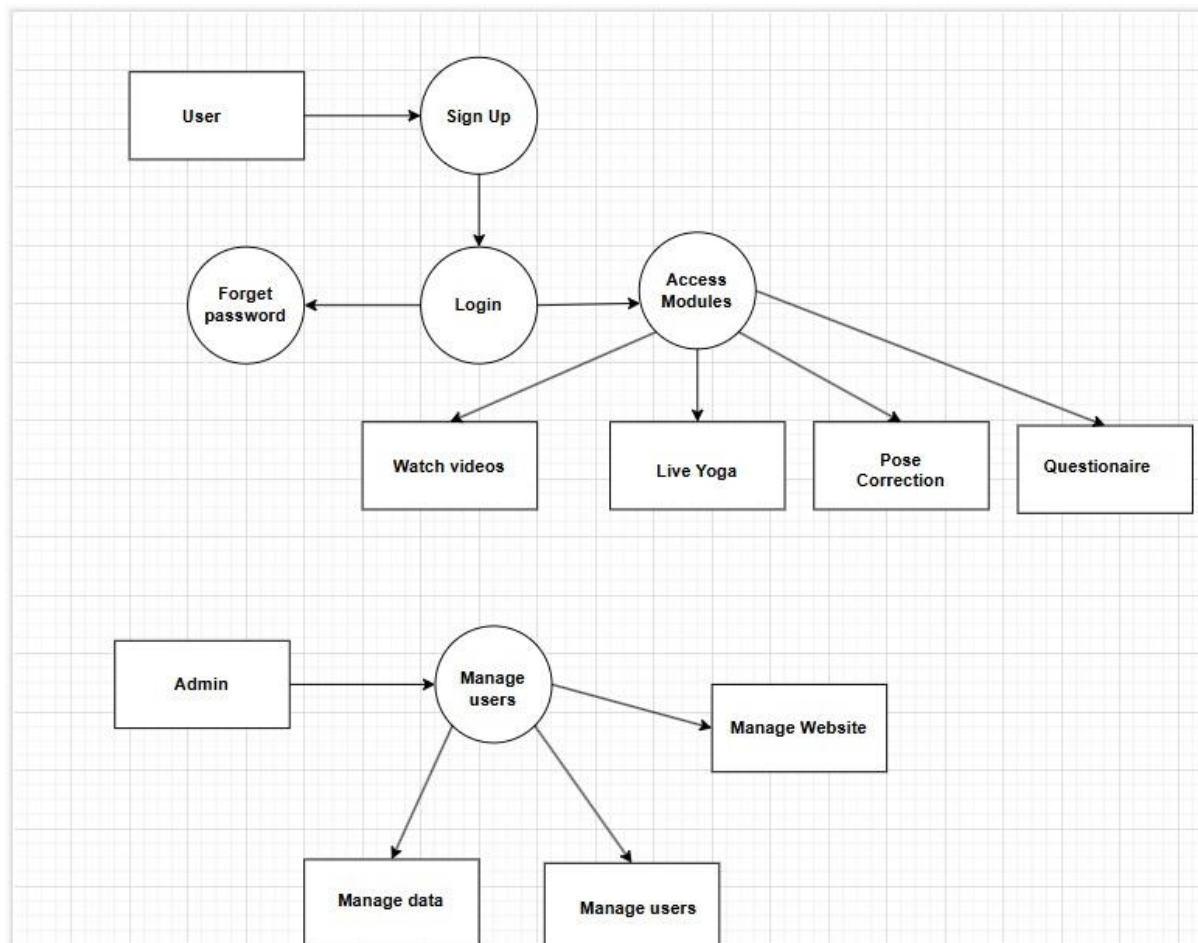


Figure 7 DFD Level 2

9 DATA DICTIONARY

9.1 SIGN – UP PAGE USER

Attribute	Data Type	Description	Constraints
FullName	Integer	Unique identifier for each user.	Primary key, Auto increment
Email	String (100)	The name of the user.	Not Null
Username	String (100)	User's email address. Used for login and identification.	Unique, Not Null
password	String (200)	Hashed password of the user.	Not Null
Confirm Password	String (VARCHAR) 200	Field to confirm the entered password. Must match Password	Not Null

9.2 SIGN – UP PAGE ADMIN

Field Name	Data Type	Description	Constraints
Username	String (VARCHAR) 50	The unique username entered for logging in.	Cannot be NULL
Password	String (VARCHAR) 200	The password entered by the user for authentication (hashed)	Cannot be NULL

Purpose of Each Table

1. **User Table:** stores user account details such as username.email and password
2. **PoseTable:** Contains information about yoga poses. including difficulty level, instruction, related images
3. **Session Table:** Tracks individual yoga sessions, linking user to specific poses and session timing
4. **Feedback Table:** Records feedback generated during yoga sessions, helping users improve their posture and accuracy.
5. **Progress Table:** Monitors a users consistency and overall progress over time including practice history and frequency.

10 IMPLEMENTATION DETAILS

The implementation of the **AI Yoga Trainer** Website involves key algorithms and processes for real-time yoga pose monitoring, user interaction, and feedback generation. Below are the detailed implementation steps.

10.1 ALGORITHM FOR POSE DETECTION

The yoga pose detection algorithm uses pre-trained models like OpenPose or MediaPipe to monitor and guide users' yoga sessions in real-time.

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>AI Yoga Trainer - Login</title>

  <style>

    @import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700;800;900&di
splay=swap');

    * {

      margin: 0;

      padding: 0;

      box-sizing: border-box;

      font-family: 'Poppins', sans-serif;

    }

    body {

      display: flex;

      justify-content: center;

      align-items: center;

      min-height: 100vh;
```

```
background: radial-gradient(circle, rgba(0, 0, 0, 1) 0%, rgba(18,18,18,1) 100%);
}

.wrapper {
  position: relative;
  width: 400px;
  height: 500px;
  background: #121212;
  box-shadow: 0 0 20px #0ef;
  border-radius: 20px;
  padding: 40px;
  overflow: hidden;
  display: flex;
  flex-direction: column;
  justify-content: center;
}

h2 {
  font-size: 30px;
  color: #fff;
  text-align: center;
  margin-bottom: 20px;
}

.input-group {
  position: relative;
  margin: 20px 0;
  background: #000;
  border-bottom: 2px solid #0ef;
}

.input-group input {
  width: calc(100% - 20px);
  height: 40px;
```



```
font-size: 16px;
color: #fff;
padding: 0 10px;
background: transparent;
border: none;
outline: none;
color: #fff;
}

button {
width: 100%;
height: 40px;
background: #0ef;
color: #000;
border: none;
border-radius: 20px;
font-size: 16px;
font-weight: 500;
cursor: pointer;
box-shadow: 0 0 15px #0ef;
margin-top: 20px;
}

.remember {
color: #fff;
font-size: 14px;
display: flex;
align-items: center;
margin-bottom: 20px;
}

.remember input {
margin-right: 10px;
```

```
    accent-color: #0ef;
}

.link {
    color: #0ef;
    text-align: center;
    text-decoration: none;
    font-weight: 500;
    cursor: pointer;
}

.link:hover {
    text-decoration: underline;
}
</style>
</head>
<body>
    <div class="wrapper">
        <h2>Login</h2>
        <div class="input-group">
            <input type="text" id="loginUsername" required placeholder="Username">
        </div>
        <div class="input-group">
            <input type="password" id="loginPassword" required placeholder="Password">
        </div>
        <div class="remember">
            <input type="checkbox" id="rememberMe"> <label for="rememberMe">Remember
me</label>
        </div>
        <button onclick="checkLogin()">Login</button>
        <p class="link" onclick="toggleForms()">Don't have an account? Sign Up</p>
    </div>
```

```
<div class="wrapper" style="display: none;">
  <h2>Sign Up</h2>
  <div class="input-group">
    <input type="text" id="signupUsername" required placeholder="Username">
  </div>
  <div class="input-group">
    <input type="email" id="signupEmail" required placeholder="Email">
  </div>
  <div class="input-group">
    <input type="password" id="signupPassword" required placeholder="Password">
  </div>
  <button onclick="registerUser()">Sign Up</button>
  <p class="link" onclick="toggleForms()">Already have an account? Sign In</p>
</div>
<script>
function toggleForms() {
  const forms = document.querySelectorAll('.wrapper');
  forms.forEach(form => {
    form.style.display = (form.style.display === 'none' ? 'flex' : 'none');
  });
}

function registerUser() {
  event.preventDefault();
  const username = document.getElementById('signupUsername').value;
  const password = document.getElementById('signupPassword').value;
  const email = document.getElementById('signupEmail').value;
  localStorage.setItem(username, JSON.stringify({email: email, password: password}));
  alert('Registration successful. Please log in.');
```

```
toggleForms();
}
```

```
}

function checkLogin() {
  event.preventDefault();
  const username = document.getElementById('loginUsername').value;
  const password = document.getElementById('loginPassword').value;
  const user = JSON.parse(localStorage.getItem(username));
  if (user && user.password === password) {
    alert('Login successful');
    window.location.href = 'home.html';
  } else {
    alert('Invalid username or password');
  }
}
</script>
</body>
</html>
```

Key Technologies

Frontend:

- HTML, CSS, JavaScript for the user interface.
- React or Angular for modular design.
- Backend:
- Flask or Django for API and server logic.
- TensorFlow or PyTorch for AI model integration.
- AI Models:
- MediaPipe or PoseNet for pose estimation.
- Database:
- MySQL or MongoDB for storing user details, pose data, and session logs

11 TESTING

Testing is a critical phase of the AI Yoga Trainer Website development to ensure the application functions as expected. It involves validating core functionalities, usability, performance, and error handling.

12.1 Testing Module Used

The following testing approaches were employed:

Manual Testing:

Tested core functionalities like user registration, login, and pose feedback accuracy manually to identify any visible issues.

Unit Testing:

Focused on testing individual components such as database queries and pose detection modules to ensure each function works independently.

Integration Testing:

Validated seamless communication between the front-end, back-end, and AI-based pose detection system.

Usability Testing:

Assessed the user interface for ease of use and accessibility to ensure a user-friendly experience. Key Features Tested.

12.2 Key Features Tested

User Authentication:

Verified both valid and invalid scenarios for:

User registration.

User login.

Ensured email uniqueness checks during registration to prevent duplicates.

Pose Detection:

Tested real-time tracking of body key points using live camera feed.

Evaluated the accuracy and relevance of corrective feedback for incorrect postures.

Error Handling:

Validated system responses to:

Empty fields during user registration or login.

Incorrect inputs like invalid email formats or passwords.

Checked for appropriate and user-friendly error messages.

Database Operations:

Conducted CRUD (Create, Read, Update, Delete) operations on the user database to ensure data integrity.

Verified the security and encryption of stored user data.

12.3 Test Tools**Postman:**

Used to test and validate API endpoints and back-end functionality.

Mediapipe Visualization:

Used to visualize and validate pose detection accuracy by analyzing key point tracking.

SQLite Viewer:

Used to inspect and ensure the integrity of data stored in the SQLite database.

12 USER MANUAL

This user manual provides detailed, step-by-step instructions to help users install, navigate, and effectively use the AI Yoga Trainer Website.

12.1 INSTALLATION STEPS

- **Software Requirements**
- **Operating System:** Windows, macOS, or Linux
- **Python Version:** 3.7 or higher
- **Required Libraries:**
- **Flask:** For backend development and API creation.
- **SQLAlchemy:** For database operations.
- **OpenCV:** For camera feed processing and pose detection.
- **Mediapipe:** For pose estimation and body key point tracking.
- **Werkzeug:** For secure password hashing.

12.2 FEATURES OVERVIEW

Home Page:

Access the main interface to navigate between features like yoga sessions, instructions, and profile management.

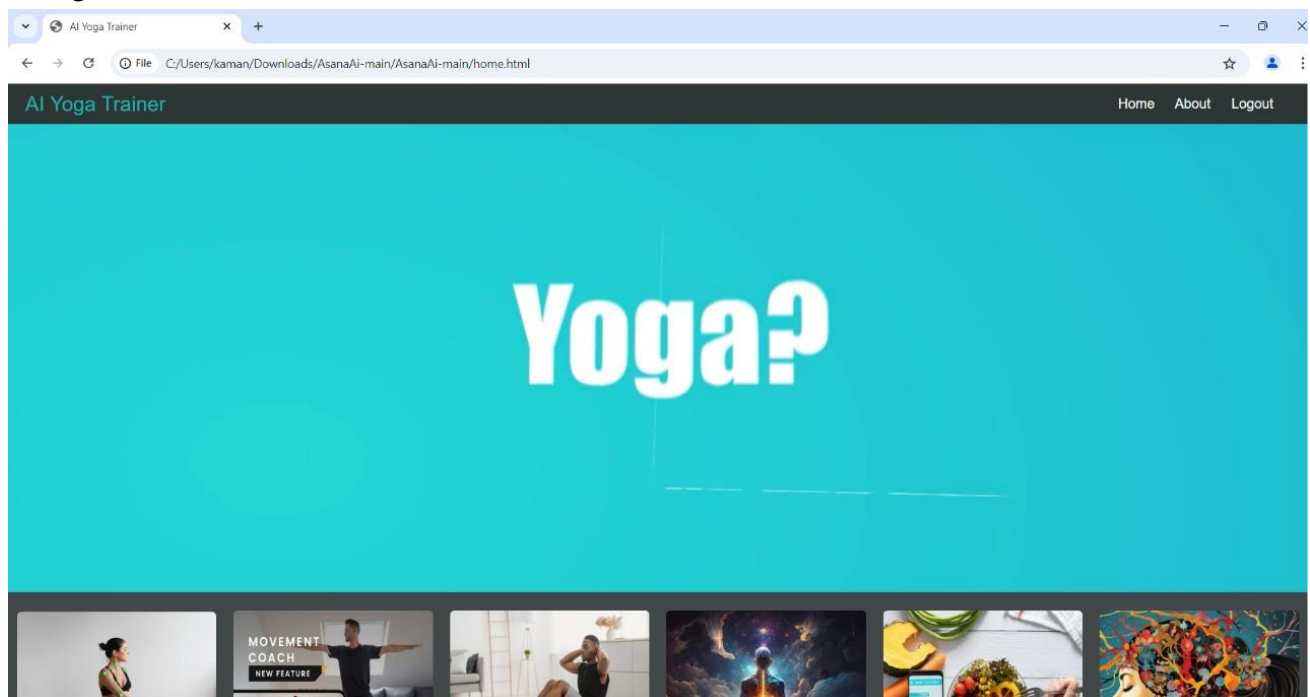
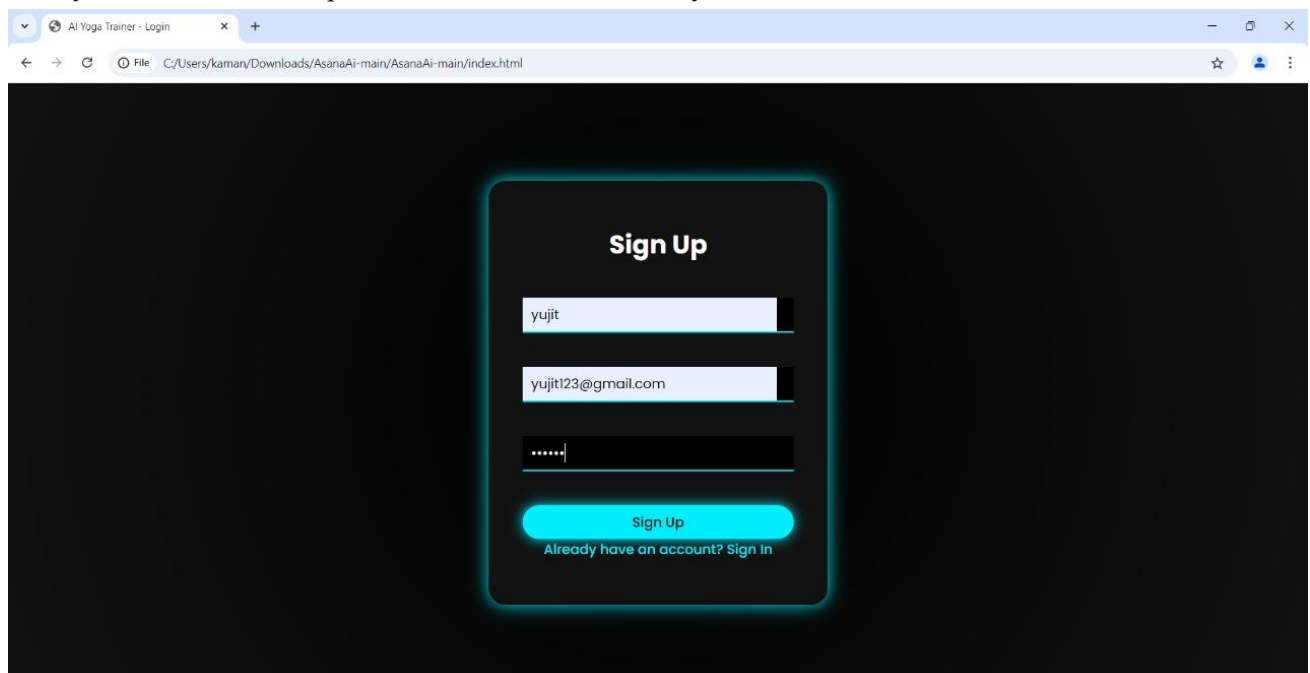


Figure 8: Home page

Sign-Up Page:

Register as a new user by providing your username, email, and password.

The system validates the inputs and stores the data securely.



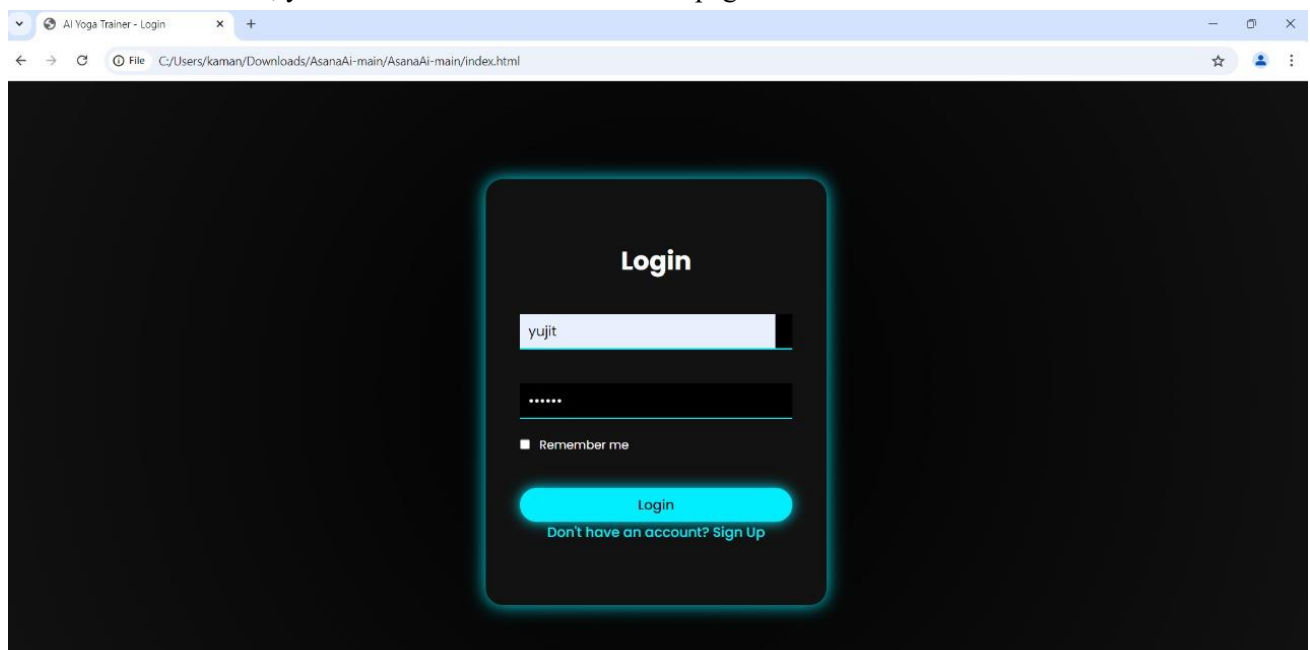
A screenshot of a web browser displaying the 'Sign Up' page. The browser's address bar shows the file path 'C:/Users/kaman/Downloads/AsanaAi-main/AsanaAi-main/index.html'. The page has a dark background with a central white box containing the 'Sign Up' form. The form includes three input fields: 'username' with the value 'yujit', 'email' with the value 'yujit123@gmail.com', and 'password' with masked characters '*****'. Below the inputs is a red 'Sign Up' button and a link that says 'Already have an account? Sign In'.

Figure 9: Sign-up page

Sign-In Page:

Log in using your registered email and password.

If credentials are valid, you will be redirected to the home page.



A screenshot of a web browser displaying the 'Login' page. The browser's address bar shows the file path 'C:/Users/kaman/Downloads/AsanaAi-main/AsanaAi-main/index.html'. The page has a dark background with a central white box containing the 'Login' form. The form includes two input fields: 'email' with the value 'yujit' and 'password' with masked characters '*****'. Below the inputs is a checkbox labeled 'Remember me' and a red 'Login' button. At the bottom of the form is a link that says 'Don't have an account? Sign Up'.

Figure 10: Login page

After Sign in Homepage:

This is the page shown after logging in showing the options the user can choose from.

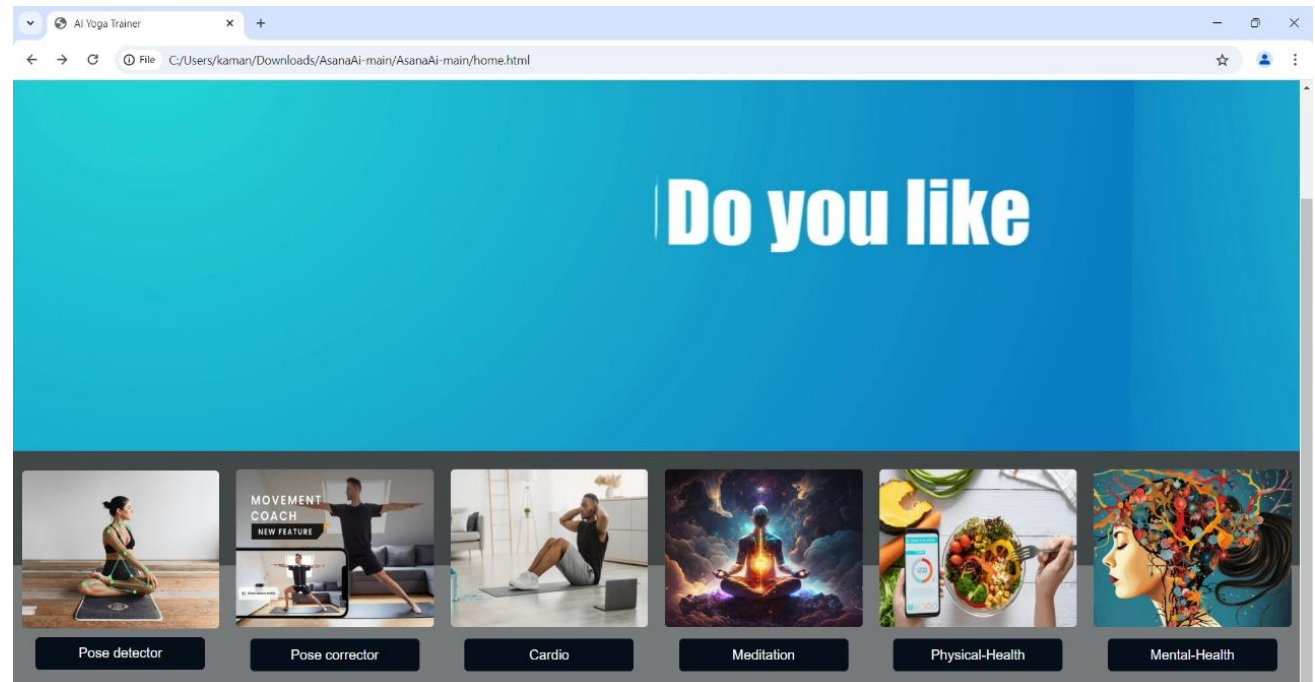


Figure 11: Home Page Modules

Yoga Practice Interface:

View detailed instructions and visual demonstrations for selected yoga poses.

Perform poses in front of your webcam and receive real-time feedback on accuracy and alignment.

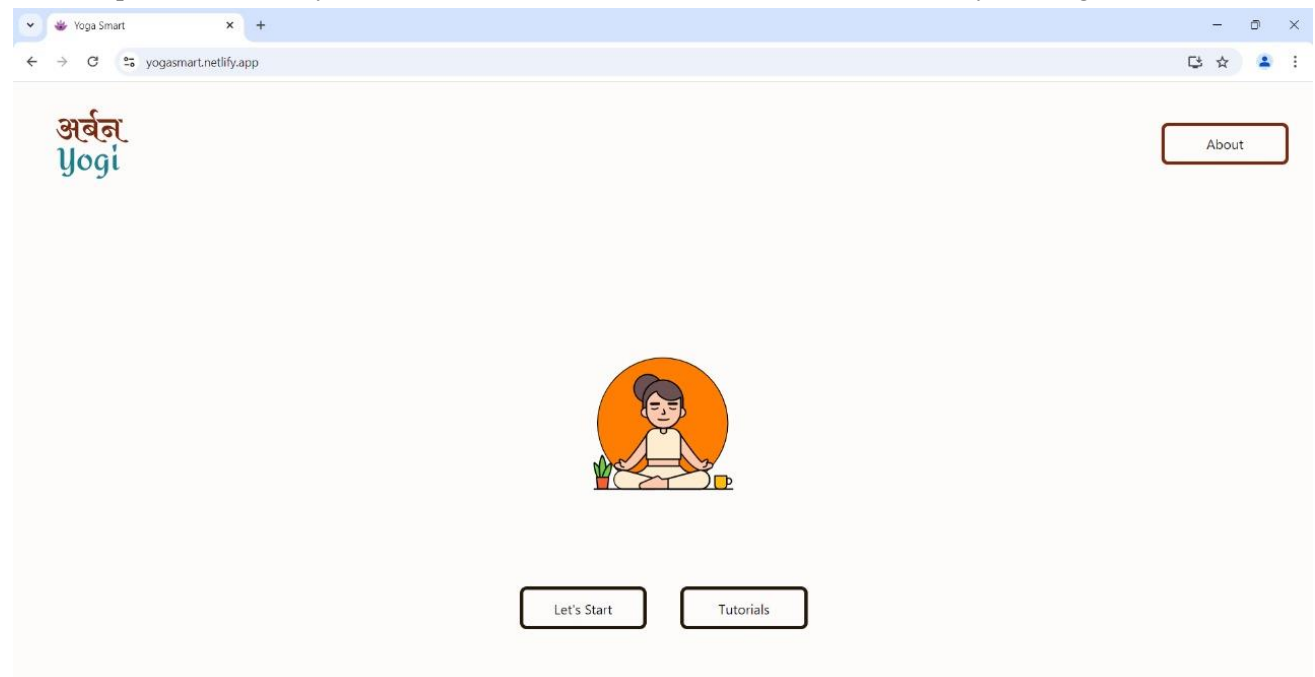


Figure 12: Yoga Pose Home page

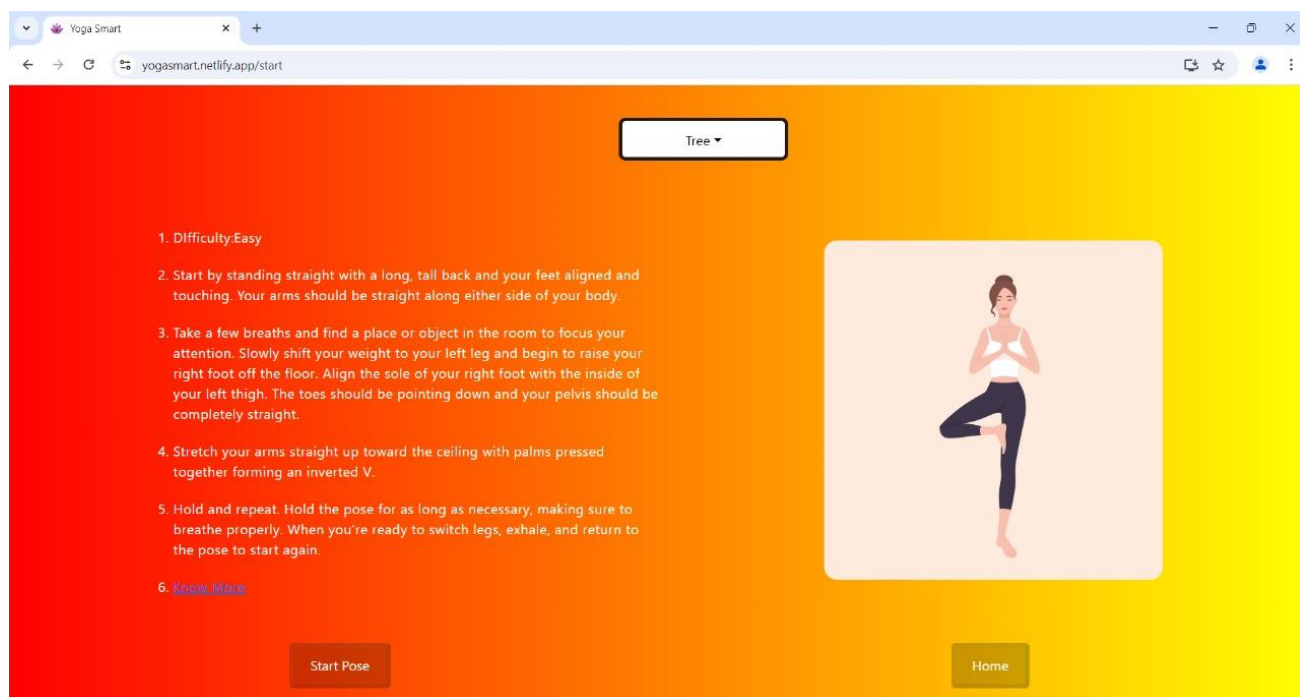


Figure 13: Pose detection interface

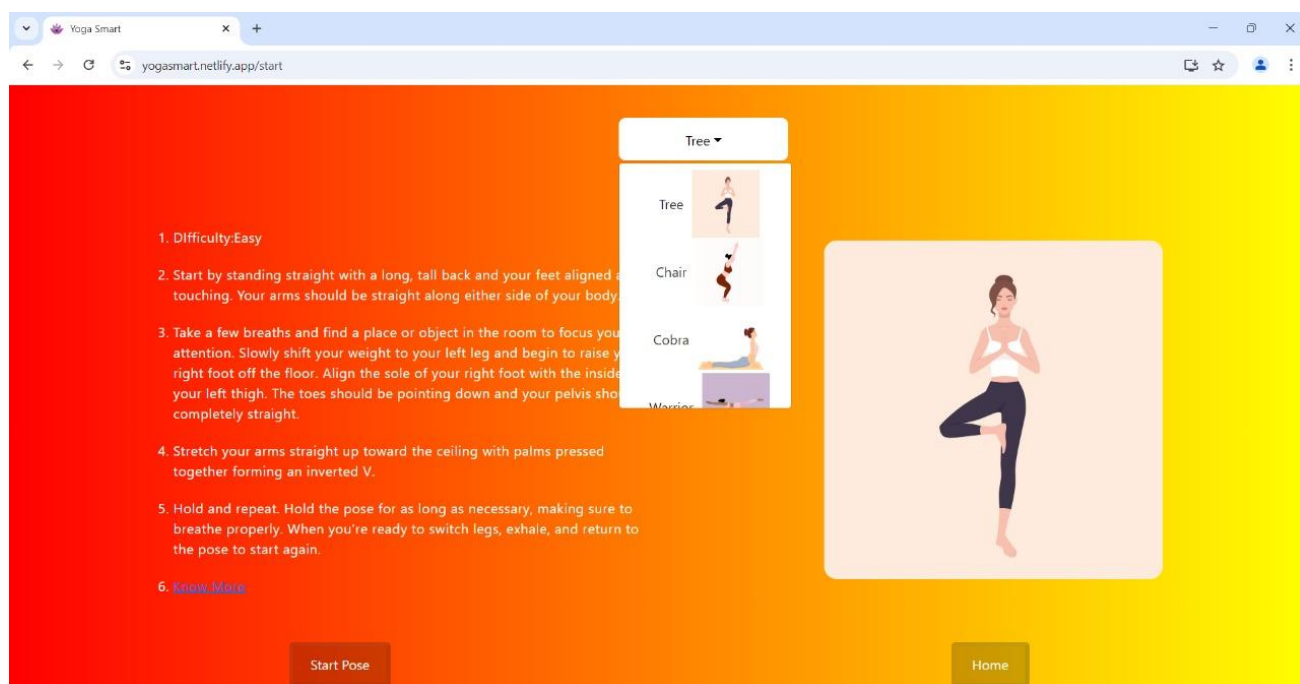


Figure 14 Pose detection Modules

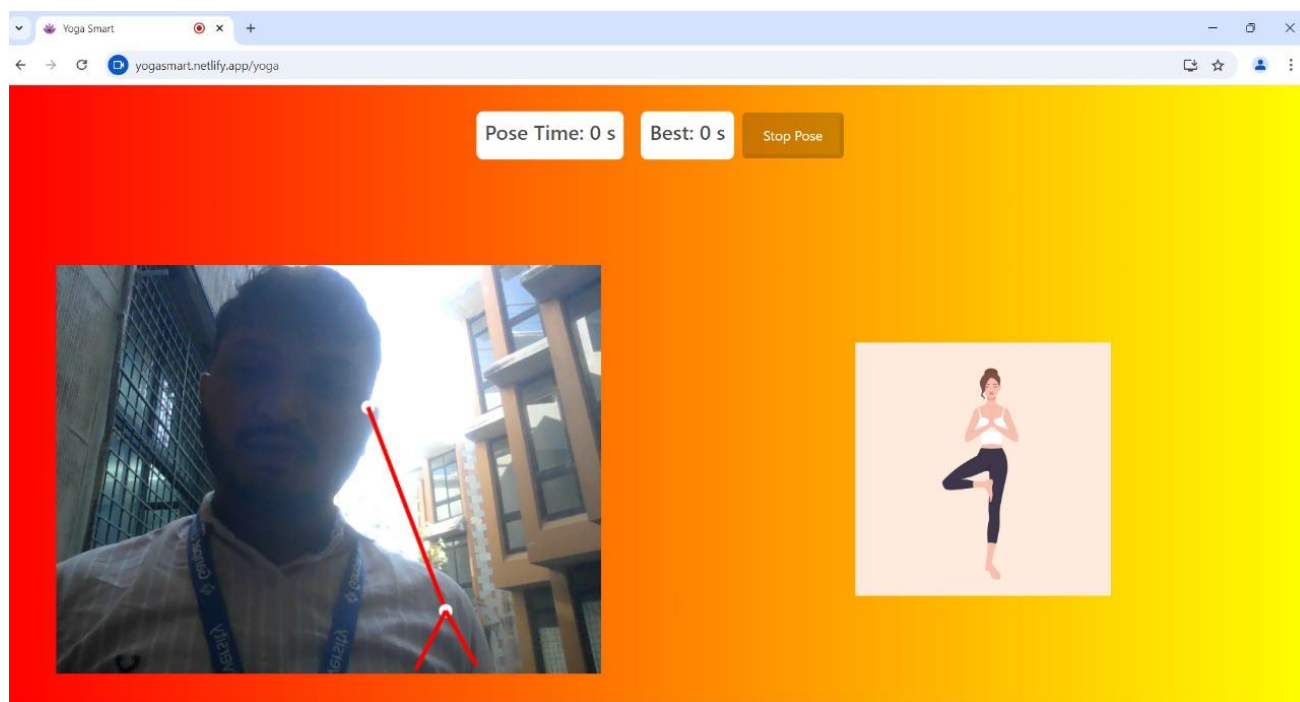


Figure 15 pose detection

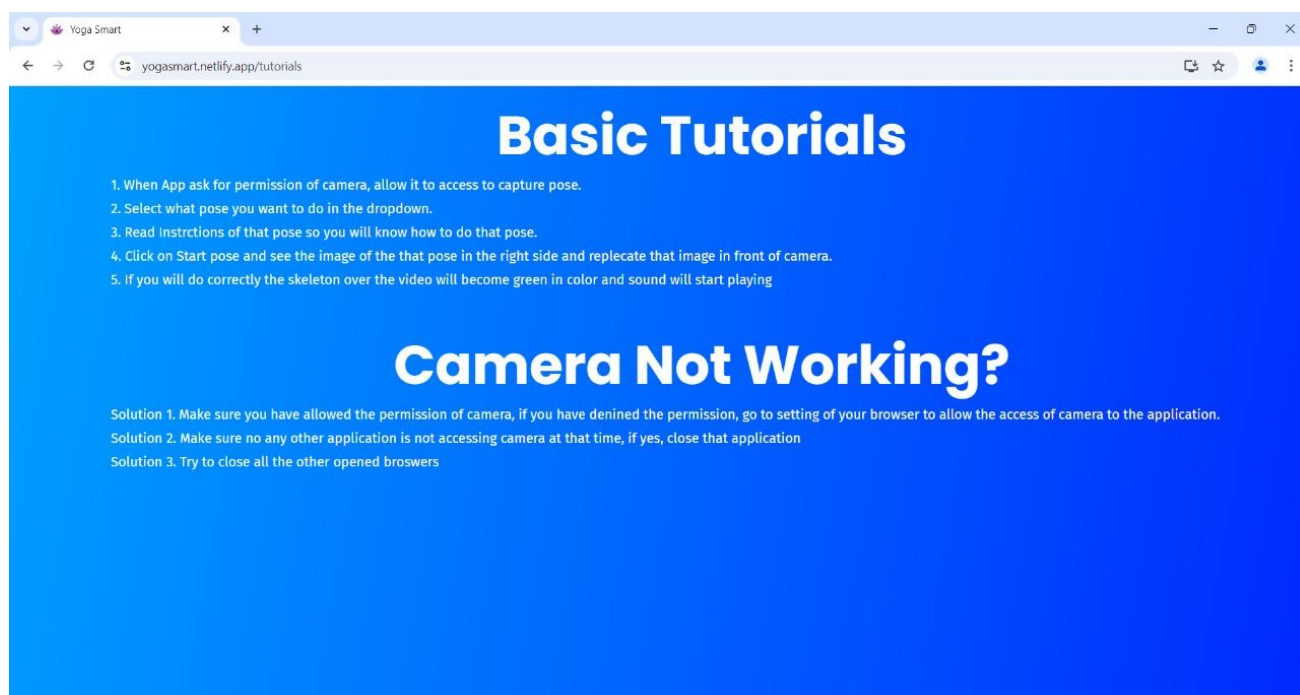


Figure 16 Pose Detection about page

Yoga Pose Learner:

This page provides the feedback to the user so that he can learn on hand on how o improve his pose.

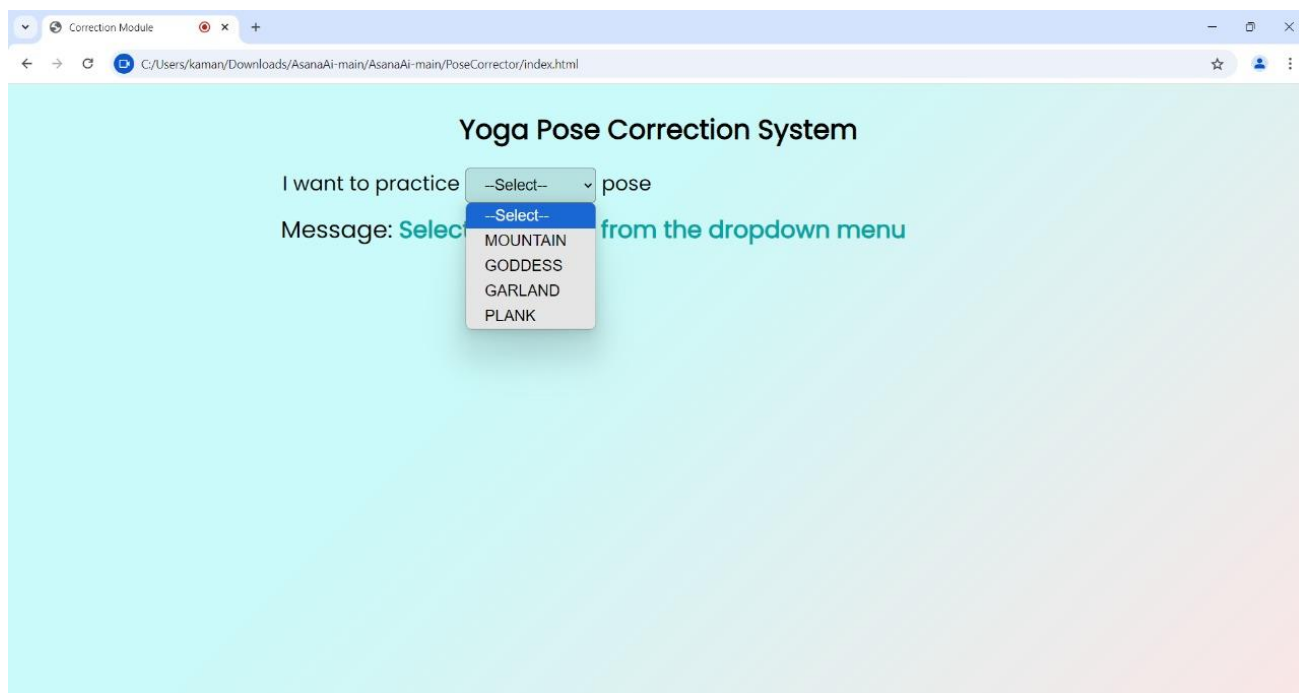


Figure 17 Pose Learner options

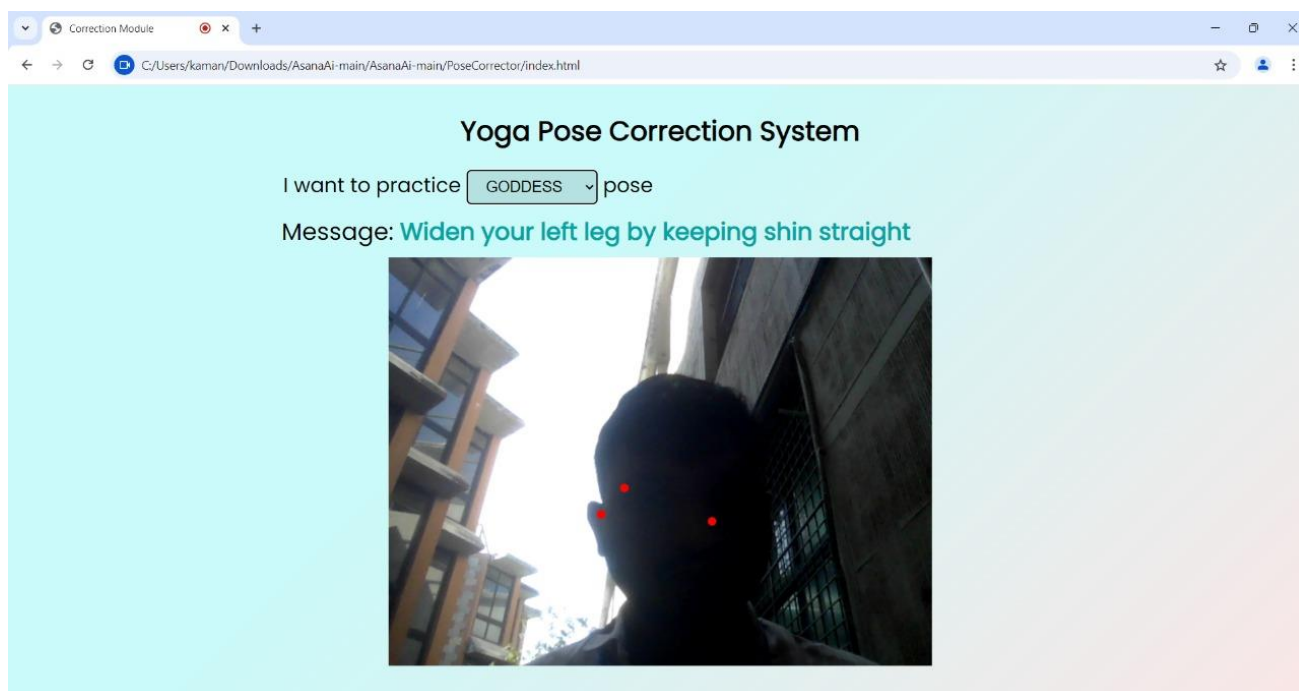


Figure 18 Pose Learner page

Cardio Option:

This page provides guidance specifically for cardio workout options present on the page

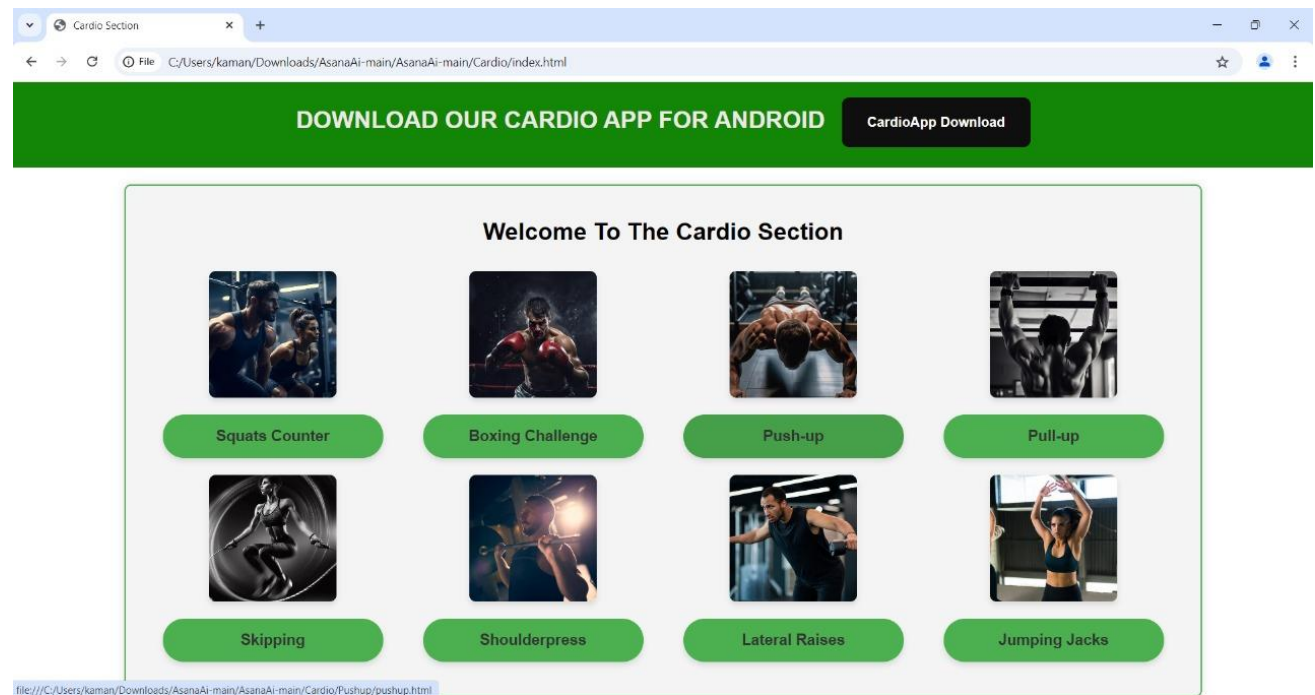


Figure 19 Cardio interface

Meditation:

This page helps you with your meditation while tracking your progress.

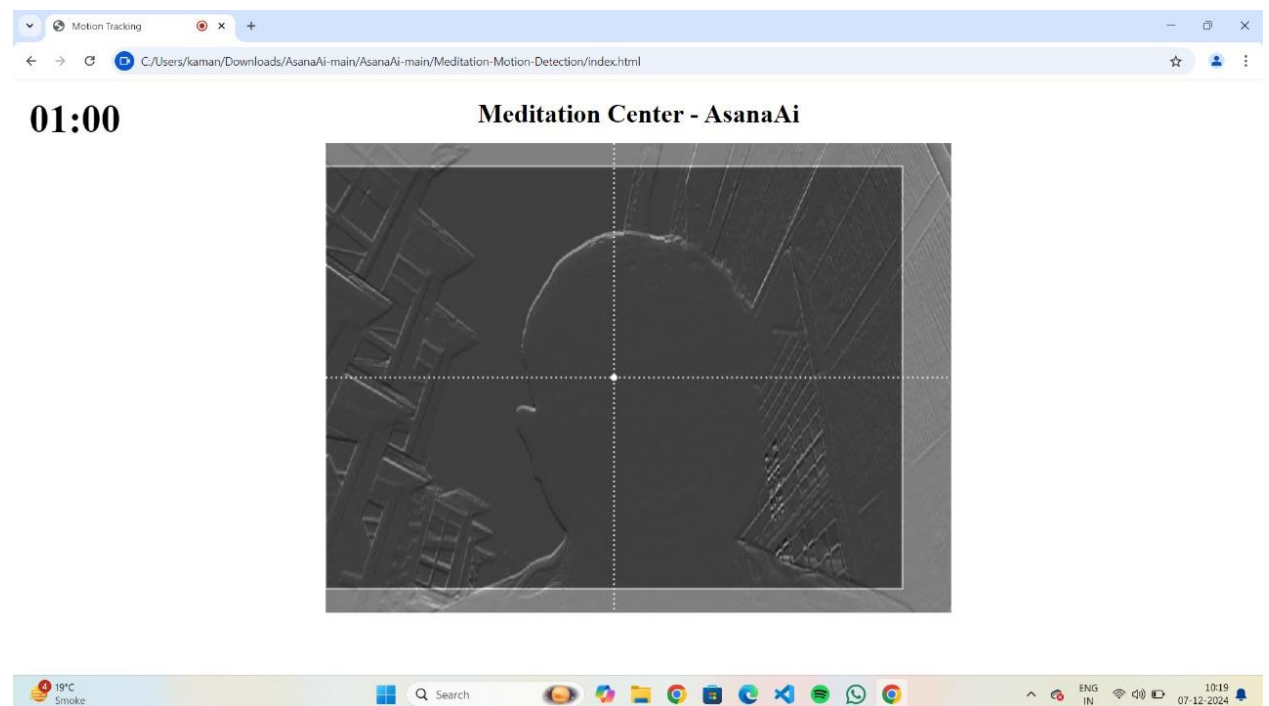


Figure 20 Meditation interface

Mental Health Screening:

This page takes a quiz with some questions and evaluates your mental health.

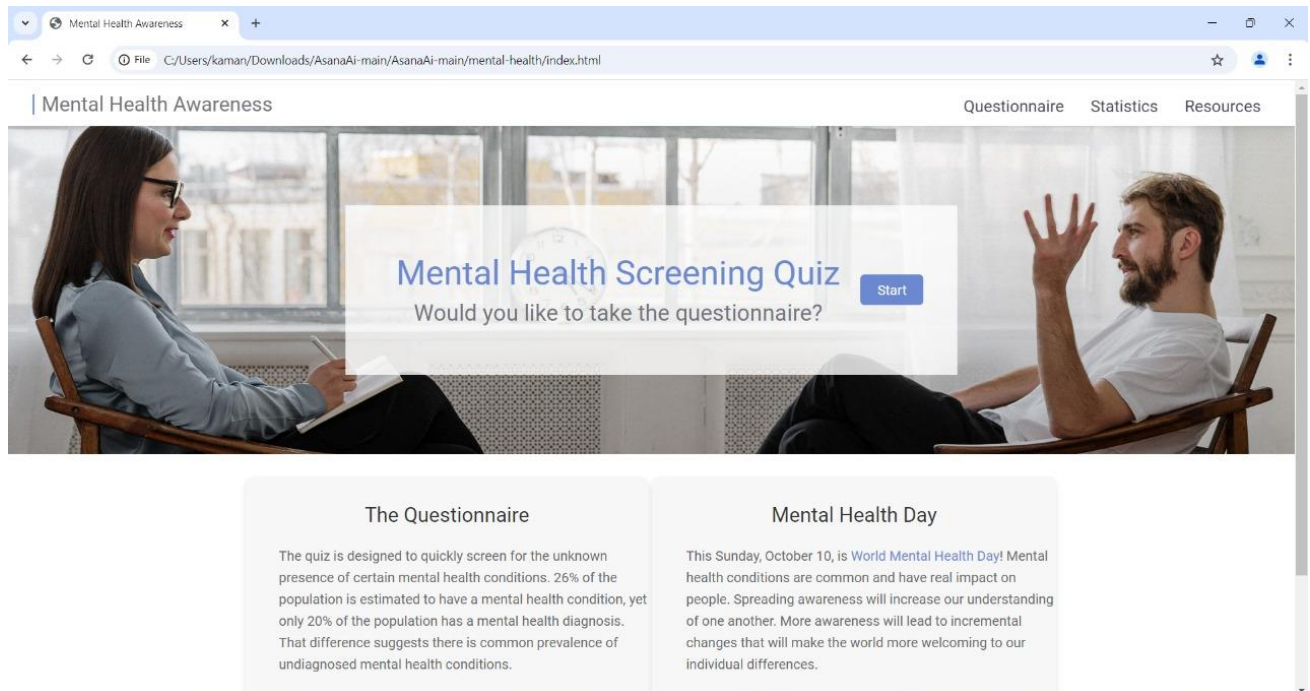


Figure 21 Mental Health queries

12.3 EXPLANATION

- **Sign-Up Page:**

Description: Allows new users to create an account

Instructions :

Enter your name, email, and password.

Click the Sign-Up button to register.

- **Feedback:**

A success message is displayed if the registration is successful.

If the email is already registered, an error message is shown.

- **Sign-In Page:**

Description: Enables registered users to log into their accounts.

- **Instructions:**

Enter your registered email and password.
Click the Sign In button to log in.

- **Feedback:**

Successful login redirects the user to the Home Page.
In correct credentials display an error message.

- **Sign-Up Page**

Users can register by entering their name, email, and password.

- **Sign-In Page**

Returning users can log in to their account.

- **Yoga Session**

The system uses the webcam to detect poses and provides feedback in real time.

12.4 DEPLOYMENT STEPS

Local Deployment

Follow the installation steps outlined above.

Production Deployment

- Use a production-ready WSGI server like **Gunicorn** or **uWSGI**.
- Set up a reverse proxy with **Nginx** or **Apache** for secure hosting.
- Deploy on cloud platforms like **AWS**, **Google Cloud**, or **Heroku**.

13 CONCLUSION AND FUTURE WORK

13.1 CONCLUSION

The AI Yoga Trainer Website successfully combines artificial intelligence, computer vision, and web technologies to deliver an innovative yoga training platform. By offering real-time feedback on posture and movement, it ensures users practice yoga safely and accurately, minimizing the risk of injuries. The platform is inclusive, catering to beginners with step-by-step guidance while providing advanced feedback for experienced practitioners.

The project demonstrates the immense potential of AI-driven solutions in the fitness domain, showcasing their ability to enhance personalization, improve user engagement, and contribute to physical and mental well-being. Its user-friendly interface, real-time pose correction, and scalability position it as a valuable tool for yoga enthusiasts seeking convenience and effectiveness in their practice.

13.2 FUTURE WORK

To enhance the AI Yoga Trainer Website, the following features and developments are proposed:

1. Custom Routines

Allow users to design and save personalized yoga routines tailored to their goals, preferences, and fitness levels.

2. Subscription Plans

Introduce monetization through premium features such as advanced pose analytics, personalized coaching, and exclusive content.

3. Mobile Integration

Develop a mobile application for iOS and Android platforms, making the platform portable and accessible anytime, anywhere.

4. Progress Tracking and Analytics

Enable users to monitor their performance over time with detailed insights into improvement trends.

Implement a scoring system for poses to gamify the experience and boost user engagement.

5. AI Model Enhancement

Integrate advanced pose estimation models for improved accuracy in challenging conditions, such as poor lighting or partial visibility.

Expand the AI system to recognize a wider variety of yoga poses and sequences.

6. Voice Assistance

Add voice-guided instructions for hands-free practice, enabling users to focus on their movements without needing to look at the screen.

7. Multi-User Support

Introduce group session capabilities, where the system can monitor multiple users simultaneously for family yoga or fitness classes.

ANNEXURE

GLOSSARY OF TERMS AND ABBREVIATION

AI (Artificial Intelligence): A branch of computer science focused on creating intelligent systems capable of performing tasks that typically require human intelligence.

SQLite: A lightweight database engine used for local data storage within the application.

Flask: A Python-based web framework used for server-side logic and routing.

UI (User Interface): The graphical and interactive components that users interact with in the application.

ABOUT TOOL AND TECHNOLOGY

Flask Framework

Purpose:

Manages server-side logic, routing, and user session handling.

Facilitates API integration for pose detection and feedback delivery.

Key Features:

Lightweight and easy to use for building scalable web applications.

Handles backend operations efficiently.

SQLite Database

Purpose:

Stores user information such as credentials, preferences, and session logs.

Advantages:

Simple, lightweight, and requires minimal configuration, making it ideal for this application.

HTML (HyperText Markup Language)

Purpose:

Structures the web pages and lays out content for the user interface.

Role in the Project:

Provides the skeleton of web pages for features like the Sign-Up, Sign-In, and Yoga Practice interfaces.

CSS (Cascading Style Sheets)

Purpose:

Styles the web pages, ensuring they are visually appealing and responsive across devices.

Role in the Project:

Adds design elements like layout, color schemes, and animations to enhance user experience.

JavaScript

Purpose:

Adds interactivity to web pages, enabling dynamic content updates.

Role in the Project:

Implements features like form validation, animations, and live camera feed processing.

Python**Purpose:**

Core programming language for developing AI logic and server-side scripts.

Role in the Project:

Integrates components like pose detection, user authentication, and data management.

OpenCV (Open Source Computer Vision Library)

Purpose:

Handles the pose detection module by tracking body key points and comparing them to predefined poses.

Key Features:

Efficient processing of real-time video input for pose analysis and feedback.

Technology Stack Summary

The combination of these technologies ensures:

Efficiency: Lightweight frameworks and libraries optimize performance.

Reliability: Robust tools like Flask and SQLite maintain system stability.

Scalability: The modular design supports future enhancements and increased user loads.

REFERENCES**TensorFlow Documentation:**

<https://www.tensorflow.org/>

MediaPipe Pose Estimation:

<https://mediapipe.dev/>

Flask Framework Documentation:

<https://flask.palletsprojects.com/>

OpenCV Documentation:

<https://opencv.org/>

SQLite Documentation:

<https://sqlite.org/>

W3Schools HTML, CSS, and JavaScript Tutorials:

<https://www.w3schools.com/>

Python Official Documentation:

<https://www.python.org/doc/>

About College (UVPCE)

U.V. Patel College of Engineering
Ganpat University



Ganpat University-U. V. Patel College of Engineering (GUNI-UVPCE) is situated in Ganpat Vidyanagar campus. It was established in September 1997 with the aim of providing educational opportunities to students from It is one of the constituent colleges of Ganpat University various strata of society. It was armed with the vision of educating and training young talented students of Gujarat in the field of Engineering and Technology so that they could meet the demands of Industries in Gujarat and across the globe.

The College is named after Shri Ugarchandbhai Varanasibhai Patel, a leading industrialist of Gujarat, for his generous support. It is a self-financed institute approved by All India Council for Technical Education (AICTE), New Delhi and the Commissionerate of Technical Education, Government of Gujarat.

The College is spread over 25 acres of land and is a part of Ganpat Vidyanagar Campus. It has six ultra-modern buildings of architectural splendor, class rooms, tutorial rooms, seminar halls, offices, drawing hall, workshop, library, well equipped departmental laboratories, and several computer laboratories with internet connectivity through 1 Gbps Fiber link, satellite link education center with two-way audio and one-way video link. The superior infrastructure of the Institute is conducive for learning, research, and training.

The Institute offers various undergraduate programs, postgraduate programs, and Ph.D. programs.

Our dedicated efforts are directed towards leading our student community to the acme of technical excellence so that they can meet the requirements of the industry, the nation and the world at large. We aim to create a generation of students that possess technical expertise and are adept at utilizing the technical 'know-hows' in the service of mankind.

We strive towards these Aims and Objectives:

- To offer guidance, motivation, and inspiration to the students for well-rounded development of their personality.
- To impart technical and need-based education by conducting elaborated training programs.
- To shape and mold the personality of the future generation.
- To construct fertile ground for adapting to dire challenges.

