***AI-Driven Personalized Pain Management with Real-Time Feedback and Relief Suggestions***

**Team Name:** ReliefXperts

**Team Members:**

| Name | Role in Assignment-1 | Github |
| --- | --- | --- |
| Suryansh Patel | Project leader and manager, responsible for generating technical reports and working on video data for pain detection. | <https://github.com/suryansh-max> |
| Jainil Patel | Involved in processing video data and integrating both the frontend and backend. | <https://github.com/jainilpatel98> |
| Cameron O'Dell | Reviewed research papers, searched for models to replicate, and contributed to deployment tools. | <https://github.com/cam-odell> |
| Jacob Xayaphet | Collected data, worked on integrating large language models (LLMs), and developed scripts for pain detection. | <https://github.com/NoviceOfCode> |

**Project Repositories:**

* <https://github.com/DS-CAPSTONE/exercise_correction> (main repo)
* <https://github.com/suryansh-max/AI-Driven-Personalized-Pain-Management-with-Real-Time-Feedback-and-Relief-Suggestions> (Supporting repo)

**Presentation link:**

<https://docs.google.com/presentation/d/1rZeckRa2NzYs-1Z8ynpQ9dPOPiCFbkwbMqNfy2ZipEk/edit?usp=sharing>

**Demo Video Link:**

**(Jainil paste link here)**

**Problem Statement**

* Incorrect exercise posture can lead to injuries, especially among new gym-goers.
* Managing pain during workouts is crucial to avoid aggravating injuries and to ensure a safe exercise routine.
* There is a need for a tool that provides immediate, personalized feedback on both posture and pain during exercises.
* Also does Sentiment analysis to see when pain hits.

**Progress**:

* TensorFlow-based AI model has been trained on video data to detect squat posture.
* Django-based backend architecture developed to process exercise data and generate feedback.
* Manual data upload and live streaming features implemented for real-time analysis.
* Initial integration of MongoDB to handle unstructured exercise and pain-related data.

**Motivation**

The increasing trend of home workouts, fitness apps, and gym users, along with the dangers of poor form and injury, highlights the need for a smart system to promote safe and efficient exercise. Incorporating pain management supports users in recovering from injuries and preventing additional issues.

**Objective and Significance**:

To create an AI-powered solution that monitors exercise posture and delivers instant feedback to users, while also addressing pain management during workouts. Also we will be doing for top 4 exercises (Squats, Lunges, Bicep Curls & Plank)

* **Posture Correction**: Use video analysis to detect incorrect exercise posture in real time and provide corrective suggestions.
* **Pain Monitoring**: Monitor signs of discomfort or pain during exercises using AI and suggest modifications to avoid injury.
* **Key Features**:
  + Real-time posture correction using camera
  + Real-time sentiment analysis for pain,
  + AI-driven pain monitoring and relief suggestions
  + Personalized pain management plans using LLMs

**Technical Details:**

For our AI-powered exercise posture correction tool, we initially focused on squats, a fundamental exercise that is widely practiced in fitness routines. Squats involve multiple joints and muscles, making them ideal for demonstrating the effectiveness of AI in detecting posture misalignments and providing real-time feedback.

#### **Model Architecture and Training**

We utilized TensorFlow to develop a custom deep learning model specifically trained to monitor squat form. Our training dataset consisted of curated video footage of individuals performing squats, representing both correct and incorrect forms(for incorrect form we manually feed data that posture is incorrect). By focusing on a variety of squat variations, we aimed to ensure the model's versatility in detecting common posture mistakes, such as knees caving in, improper back alignment, or insufficient squat depth.

During the data collection phase, we explored and sourced additional video datasets from public fitness libraries and online repositories, identifying those that showcased diverse body types, angles, and lighting conditions. These additional datasets contributed to the model’s robustness and improved its generalization across different users.

#### **Preprocessing Layer for Enhanced Data Quality**

A key challenge in video-based AI models is the variability in data quality. Factors such as lighting, camera angle, and background noise can introduce inconsistencies in the input data. To address this, we implemented a preprocessing layer designed to cleanse and normalize the input data before it reaches the model.

This preprocessing layer performs several tasks:

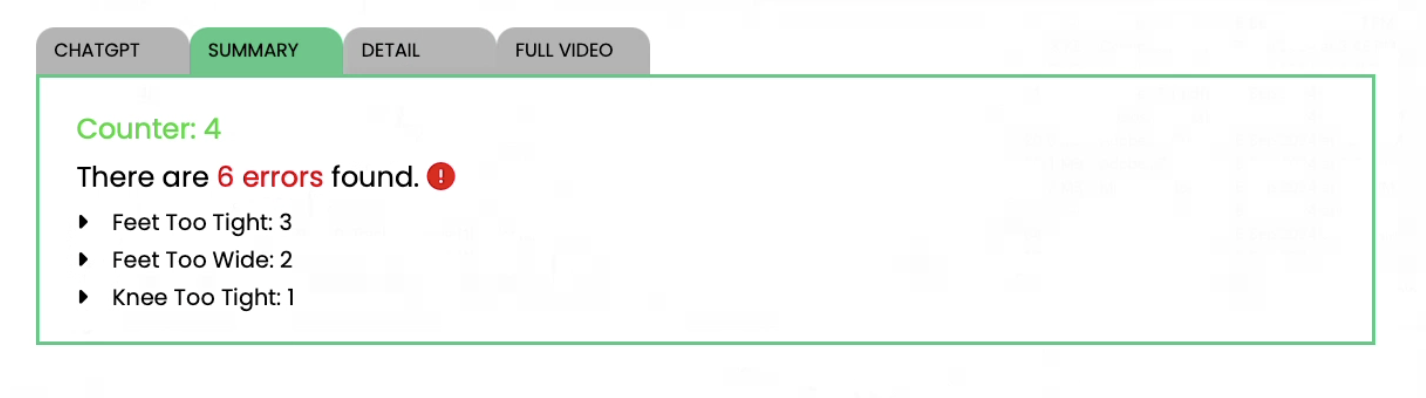
* **Data Cleansing:** It filters out noisy frames and corrects inconsistencies in video input, such as frames with obstructions or poor visibility.
* **Data Structuring:** It converts raw video input into structured data, including key pose landmarks, which are extracted and fed into the model. This helps the model focus on relevant features, such as joint angles and body alignment.
* **Frame Consistency:** The layer ensures uniformity in video frame rates and resolutions, helping to prevent fluctuations in model performance due to input irregularities.

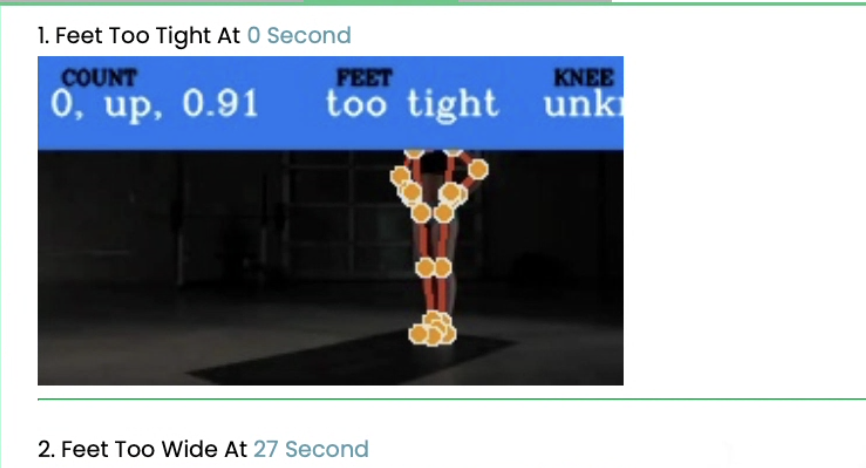
This preprocessing not only enhances the quality of the training data but also ensures smoother real-time inference by providing the model with clean, consistent input during exercise sessions. This improvement has led to more accurate posture detection and better identification of common squatting errors such as knee valgus, back rounding, or leaning forward excessively.

#### **Model Details:**

* **Data Source:** The model was trained on video datasets specifically curated to represent various squatting techniques and potential posture issues. We included videos from multiple camera angles to ensure that the model could detect errors regardless of the user’s positioning.
* **Real-Time Optimization:** One of the core goals of the model was to ensure real-time performance, which required optimizing the model architecture for low latency. We focused on reducing the number of parameters without sacrificing accuracy, allowing the model to run efficiently even on resource-constrained devices.
* **Error Detection:** The model is trained to identify specific posture errors during squats, including improper foot placement, excessive forward lean, knee misalignment, and lack of hip engagement. Based on these detections, the system provides real-time feedback on how to correct the posture.

**Results for squats:**





#### **Triangle Model**

**Frontend**:

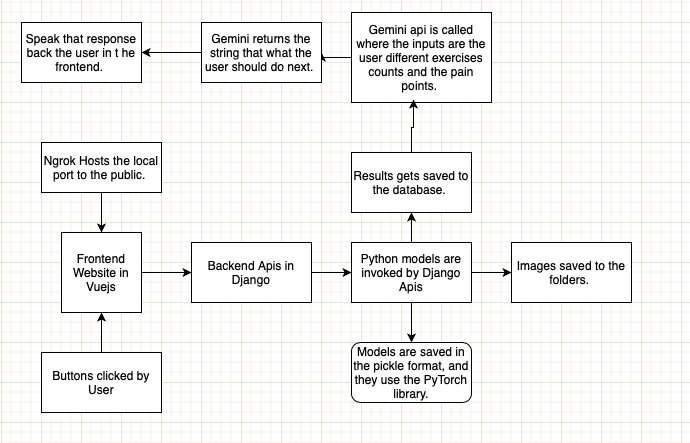
* Initially developed using React.js for user interface components, the team is transitioning to Vue.js for better maintainability. Currently, both React and Vue.js are being utilized as we shift to Vue for long-term support.
* Users can upload exercise videos or stream live workout sessions, and receive real-time feedback with visual outputs such as images, text annotations, and error counts.

**Backend**:

* Implemented using Django, the backend supports three separate "Apps" within the framework, each responsible for processing specific exercises. These apps handle real-time data input (both manual uploads and streaming) and generate feedback in the form of error reports, images, and suggestions for improvement.
* We plan to integrate ChatGPT API to provide tailored suggestions for users based on their performance.

**Database**:

* MongoDB was chosen as the primary database due to its ability to handle unstructured data and its efficiency in processing real-time streaming data. This enables us to efficiently store, retrieve, and analyze data related to both exercise performance and pain management.



Note - figure shows implementation of Triangle Model

#### **Challenges & Solutions**

* **Data Quality**: One of the main challenges was ensuring clean and accurate data for model training. The solution involved adding a preprocessing layer to our pipeline that filters and organizes the incoming data, ensuring that the model receives only well-structured data for training and analysis.
* **Frontend Transition**: As our team was more experienced in React.js but the project required us to adopt Vue.js, we faced challenges adapting to the new framework. Our solution was to maintain both frontends during the transition period, allowing team members to gradually shift their efforts to Vue.js while still supporting existing features in React.js.
* **Real-Time Data Processing**: Managing real-time posture feedback and error reporting in live streaming environments posed technical challenges. Leveraging MongoDB’s capability to handle streaming data helped solve this problem, ensuring smooth, real-time data handling.

#### **Next Steps**

* **Model Refinement**: We plan to continue enhancing our AI model by adding more data and refining the posture detection algorithms to improve accuracy.
* **ChatGPT Integration**: The next phase involves linking our system to the ChatGPT API, allowing users to receive personalized suggestions for exercise improvement and pain management.
* **Frontend Migration**: We aim to fully transition from React.js to Vue.js for frontend development while ensuring seamless user experience.
* **Expansion of Exercises**: Once the squat posture detection model is perfected, we plan to expand our system to support additional exercises, incorporating more personalized feedback and injury prevention techniques.
* **Add 3 exercises to the model:** We are planning to add 3 more exercises into our current model which needs to be code and test.

### **Individual Contributions:**

#### ***Suryansh Patel – Leader (Manager)***

GitHub:<https://github.com/suryansh-max>

Suryansh Patel serves as the project leader and manager, overseeing all phases of the development process. He is responsible for generating technical reports that summarize the progress, objectives, and challenges faced by the team. His leadership role involves coordinating team activities, ensuring deadlines are met, and maintaining a clear vision for the project.

Key Contributions:

* Coordinated and led the team in identifying the project scope and objectives.
* Managed video-related data collection and worked closely with the team on pain detection through AI models.
* Played a key role in setting up the TensorFlow-based AI model and overseeing its training on video data.
* Ensured that all technical challenges were addressed promptly, proposing solutions for data preprocessing and model refinement.
* Drafted and presented detailed technical reports to track project progress and highlight significant milestones.

#### ***Jainil Patel – Video Data Processing & Frontend/Backend Integration***

GitHub:<https://github.com/jainilpatel98>

Jainil Patel has focused extensively on video data processing, working to integrate the AI model into both the frontend and backend systems. His work involved managing the flow of data from the user interface to the AI model and ensuring that results were properly displayed to users.

Key Contributions:

* Led the effort in handling video-related data, contributing to preprocessing the data before it is passed into the AI model.
* Developed the backend system using Django, where he helped set up multiple "Apps" responsible for handling different exercises, starting with squats.
* Integrated the frontend with React.js and Vue.js for smooth data input, ensuring that users could upload videos or stream live exercises for real-time feedback.
* Contributed to the manual and live data streaming options for exercise monitoring.
* Worked on generating real-time results, such as images, text feedback, and error counts, which are displayed on the frontend.

#### ***Cameron O'Dell – Research Paper Reviewer & Deployment Specialist***

GitHub:<https://github.com/cam-odell>

Cameron O'Dell plays a critical role in reviewing existing research related to pain detection models and identifying cutting-edge AI technologies that can be applied to the project. In addition, he has been instrumental in researching and suggesting deployment tools for the project.

Key Contributions:

* Acted as the research paper reviewer, responsible for finding relevant research on AI models related to posture detection and pain management.
* Collaborated with the team to search for models that could be reproduced or modified to enhance the project’s efficiency.
* Worked on identifying and integrating deployment tools and environments to ensure the project can be easily deployed in both development and production environments.
* Provided valuable insights into optimizing model performance during deployment, helping the team make informed decisions regarding hosting and scaling.

#### 

#### ***Jacob Xayaphet – Data Collection & LLM Integration***

GitHub:<https://github.com/NoviceOfCode>

Jacob Xayaphet has been responsible for collecting and curating data, particularly video data used in training the AI model. He also took on the task of integrating large language models (LLMs) into the system to assist with pain detection.

Key Contributions:

* Led the effort in gathering and organizing data necessary for training the AI model, focusing on ensuring the video data was relevant, accurate, and varied.
* Developed scripts for processing the data and preparing it for integration into the pain detection model.
* Worked closely on incorporating LLM technology, which would later be used to enhance the pain detection feature by providing intelligent feedback to users.
* Contributed to developing the code responsible for using the collected data to detect pain during exercises, helping ensure that users receive real-time feedback on how to prevent injury or alleviate discomfort.

NOTE : Each team member has made significant contributions to the project's success, from building the AI models and integrating them into the system to refining data collection and implementing the frontend/backend infrastructure. This collaboration has driven steady progress toward achieving the project’s objectives.