**ABSTRACT**

**DOMAIN: Healthcare and Physical Therapy**

Improper posture detection falls within the domains of Healthcare and Physical therapy, addressing issues like back pain, neck strain, and reduced mobility. Correcting posture is crucial for preventing musculoskeletal problems and promoting overall well-being. Prolonged poor posture can lead to permanent deformation of bones and muscles. In today's professional landscape, long working hours contribute to improper posture, significantly impacting health Identifying and correcting bad postures also play a vital role in healing and preventing additional injuries in physical therapy and rehabilitation contexts.

**PROJECT WORK: Incorrect Posture Detection and Correction**

This project aims to utilize machine learning models to develop an automated posture detection and correction system. By leveraging advanced machine learning algorithms and computer vision techniques, the system aims to detect improper postures in real-time accurately, providing personalized feedback and recommendations to promote better posture habits, mitigate musculoskeletal issues, and enhance overall health and well-being, particularly in professional settings and physical therapy contexts.

**Repository Information:**

The "Silhouettes of Human Posture Dataset" sourced from Kaggle is a meticulously curated collection comprising a total of 4 distinct subdivisions, each meticulously designed to encapsulate a diverse range of human postural configurations. With 1200 high-resolution images meticulously categorized across four discrete classes, this dataset offers a comprehensive and nuanced representation of human body positions in various contexts and environments.

<https://www.kaggle.com/datasets/deepshah16/silhouettes-of-human-posture>

**DEVELOPMENT ENVIRONMENT:**

Programming Language: Python

Libraries Used: OpenCV, Mediapipe, Open Pose

Front-End: HTML, CSS, PHP

Frameworks:

* Python Flask for integration
* Laravel for Web-based design

|  |  |  |
| --- | --- | --- |
| **Chap No.** | **CHAPTER NAME** | **PAGENO** |
| 1 | INTRODUCTION | 4 |
| 2 | LITERATURE SURVEY | 8 |
| 3 | SYSTEM DESIGN | 9 |
|  | |  |  | | --- | --- | | 3.1 | System Architecture design | | 3.1.1 | Architecture diagram | | 3.1.2 | Design Diagrams | |  | |  |  | | --- | --- | | 3.1.2.1 | Data Flow Diagram | | 3.1.2.2 | Use Case Diagram | | 3.1.2.3 | Activity Diagram | | 3.1.2.4 | Sequence Diagram | | 3.1.2.5 | Data Design Diagram | | 3.1.2.6 | Collaboration Daigram | | | 3.2 | Module Description | | 9 |
| 9 |
| 10 |
| 10 |
| 11 |
| 12 |
| 13 |
| 13 |
| 14 |
| 15 |
|  |
| 4 | DETAILED DESIGN | 16 |
|  | |  |  | | --- | --- | | 4.1 | MODULE | |  | |  |  | | --- | --- | | 4.1.1 | Login page | | 4.1.2 | Explore page | | 4.1.3 | Map page | | 4.1.4 | Tracking page | | 4.1.5 | Progress page | | | 16 |
| 16 |
| 16 |
| 16 |
| 17 |
| 17 |
|  |
| 5 | IMPLEMENTATION AND RESULT | 18 |
|  | |  |  | | --- | --- | | 5.1 | Hardware And Software Requirements | |  | |  |  | | --- | --- | | 5.1.1 | Hardware | | 5.1.2 | Software | | 5.1.3 | Software Requirements | | | 5.2 | Classification And Clustering Tool: | |  | |  |  | | --- | --- | | 5.2.1 | Convolutional Neural Networks (CNN) | | 5.2.2 | Mediapipe | | 5.2.3 | Openpose | | 5.2.4 | Posenet: | | 5.2.5 | Comparison Table: | | 5.2.6 | Convolutional Neural Networks (CNN): | | 5.2.7 | K-Means Clustering | | 5.2.8 | Hierarchical Clustering: | | | 5.3 | Results and Discussion | |  | |  |  | | --- | --- | | 5.3.1 | Project Outcome | | 5.3.2 | Web pages Screenshot | | | 18 |
| 18 |
| 18 |
| 18 |
| 19 |
| 19 |
| 20 |
| 21 |
| 22 |
| 23 |
| 24 |
| 24 |
| 24 |
| 25 |
| 25  26 |
| 6 | CONCLUSION | 32 |
|  | |  |  | | --- | --- | | 6.1 | Conclusion | | 6.2 | Future Enhancement | | 6.3 | References | | 32 |
| 32 |
| 32 |

**CHAPTER 1: INTRODUCTION**

* 1. **DOMAIN OF THE PROJECT**

In the domain of healthcare and physical therapy, addressing posture-related issues is crucial due to their significant impact on musculoskeletal health and overall well-being. Here are some detailed aspects of this domain:

**1.1.1. Prevalence of Musculoskeletal Problems**: Musculoskeletal issues such as back pain, neck strain, and reduced mobility are prevalent among individuals, especially those with sedentary lifestyles or occupations that involve prolonged sitting or repetitive movements. These problems can affect people of all ages and can lead to discomfort, pain, and decreased quality of life if left untreated.

**1.1.2. Impact of Poor Posture:** Prolonged poor posture can have detrimental effects on the body, leading to muscle imbalances, joint stiffness, and even permanent deformation of bones and muscles. In addition to physical discomfort, poor posture can also contribute to mental fatigue, reduced productivity, and decreased overall health and vitality.

**1.1.3. Role of Physical Therapy:** Physical therapists play a crucial role in addressing posture-related issues by providing assessment, treatment, and rehabilitation services to individuals with musculoskeletal problems. They employ various techniques such as manual therapy, exercise prescription, and ergonomic education to help patients improve their posture, alleviate pain, and restore functional movement patterns.

**1.1.4. Ergonomics in Healthcare**: Ergonomic principles play a vital role in healthcare settings, particularly in the design of workstations and equipment to promote optimal body alignment and reduce the risk of injury or discomfort among healthcare professionals. Proper ergonomic design can help prevent musculoskeletal disorders and improve workplace safety and efficiency.

**1.1.5. Interdisciplinary Collaboration:** Addressing posture-related issues often requires interdisciplinary collaboration between healthcare professionals, including physical therapists, occupational therapists, chiropractors, and ergonomists. By combining their expertise and leveraging technological innovations, interdisciplinary teams can develop comprehensive approaches to assess, treat, and prevent posture-related problems across various patient populations and settings.

In summary, the domain of healthcare and physical therapy encompasses a broad range of issues related to posture, musculoskeletal health, and ergonomic principles. By addressing these issues through interdisciplinary collaboration and leveraging technological advancements, healthcare professionals can improve patient outcomes, enhance workplace safety, and promote overall well-being.

* 1. **ABOUT THE PROJECT**

The primary objective of this project is to develop an automated system that can detect and correct improper postures in real time, with a focus on promoting musculoskeletal health and well-being in healthcare and physical therapy settings.

**1.2.1. Real-Time Detection:** The system operates in real-time, continuously monitoring the user's posture during activities such as sitting, standing, or performing specific exercises. Using cameras or wearable sensors, the system captures images or data streams and processes them through machine-learning models to identify any instances of poor posture.

**1.2.2. Machine Learning Models**: The project leverages machine learning algorithms, particularly in the field of computer vision, to analyze images or video feeds and identify specific postural deviations. These models are trained on datasets containing annotated examples of proper and improper postures to accurately classify and detect deviations from optimal alignment.

**1.2.3. Real-Time Detection:** The system operates in real-time, continuously monitoring the user's posture during activities such as sitting, standing, or performing specific exercises. Using cameras or wearable sensors, the system captures images or data streams and processes them through machine-learning models to identify any instances of poor posture.

**1.2.4. User Interface:** The system features a user-friendly interface that allows individuals to interact with the feedback mechanisms, access personalized recommendations, and view their posture data and progress reports. The interface may be available through desktop applications, mobile apps, or web-based platforms for convenient access moreover the system offers

* Feedback mechanism
* Progress Tracking
* Personalized recommendations
* Long-term monitoring and analytics

**1.2.5 Dataset Information:**

In the dataset, there are four subdivisions. They are:

1. **Sitting:**

This dataset includes a diverse range of images capturing individuals seated in various positions, from chairs to the ground. It covers differences in posture, upper body orientation, and lower limb positioning across different ages, genders, and cultural backgrounds.

1. **Standing:**

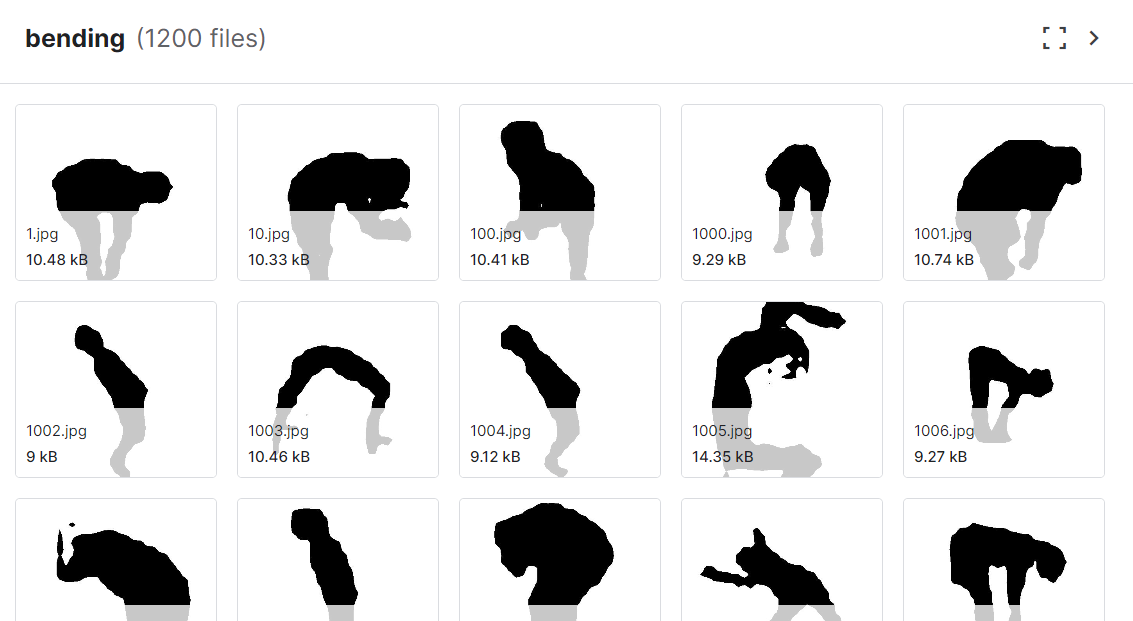
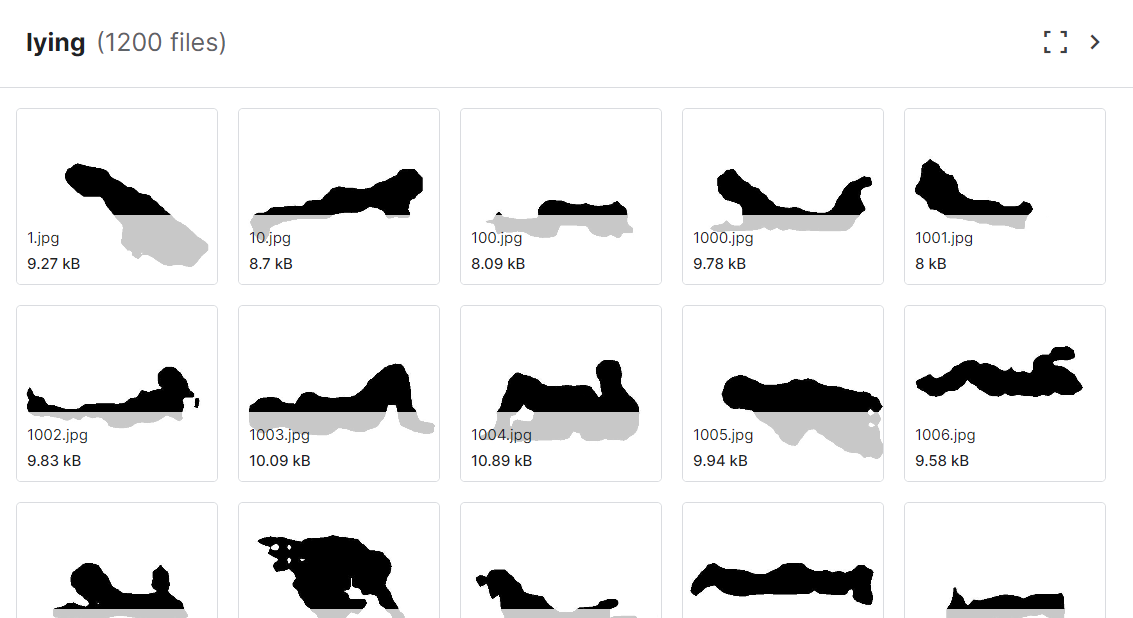
The dataset showcases individuals in upright postures, capturing variations in body alignment and limb positioning in different settings and environments.

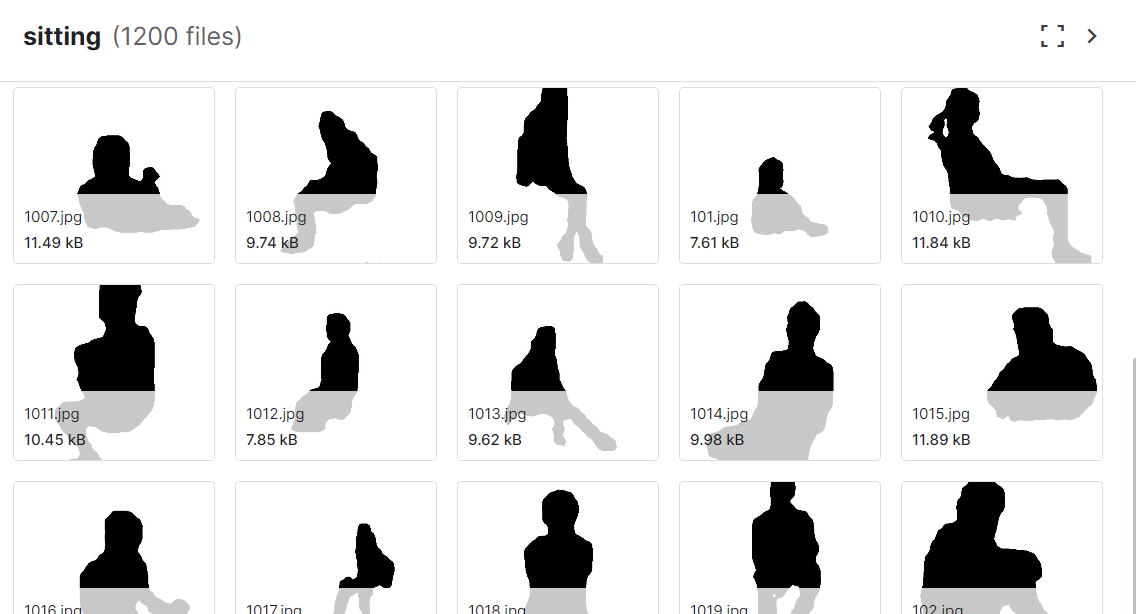
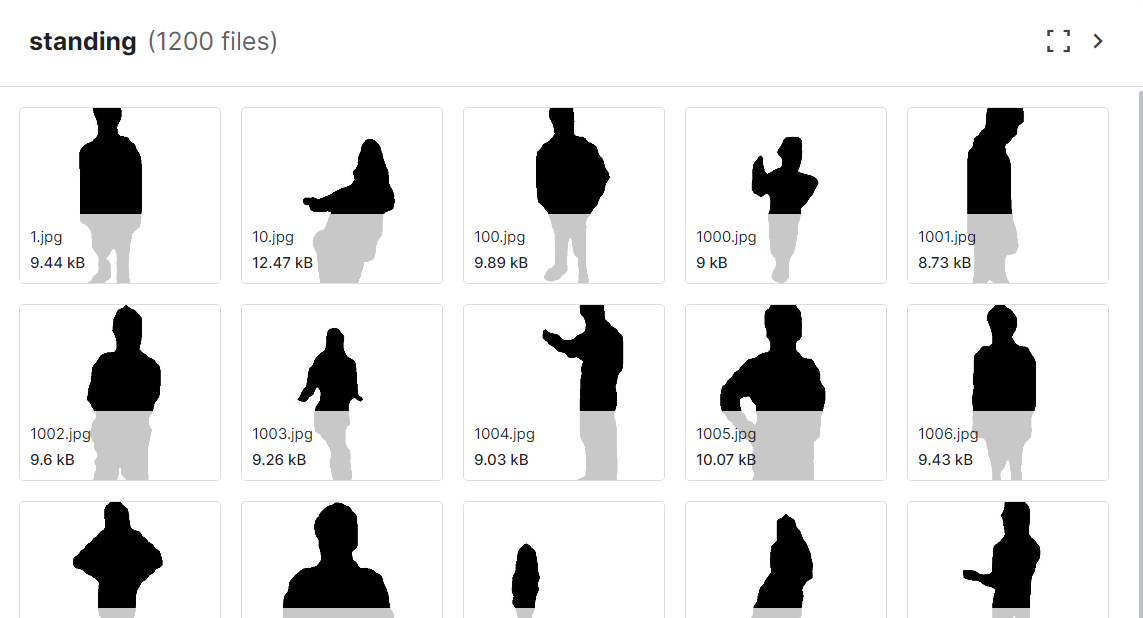
1. **Lying:**

This subdivision presents images of individuals in reclined positions, including lying on the back, stomach, or side. It captures variations in lying angles, body contours, and limb orientations for analysis of postural dynamics during rest or sleep.

1. **Bending:**

Images in this category depict individuals engaged in activities involving significant body flexion, such as picking up objects or performing exercises. It documents variations in bending angles, spinal curvature, and limb positioning to understand biomechanical principles during dynamic movements.

**** ****

** **

In summary, the Automated Posture Detection and Correction System integrates machine learning technology with principles of healthcare and physical therapy to provide real-time feedback, personalized recommendations, and long-term monitoring of posture habits. By addressing posture-related issues proactively, the system aims to improve musculoskeletal health, prevent injuries, and promote overall well-being in diverse settings.

* 1. **ORGANISATION OF THE REPORT**

**Literature Survey:**

In the literature survey section, provide a concise overview of existing research, methodologies, and technologies related to improper posture detection systems. Highlight key studies, algorithms, and approaches used in posture detection, including their strengths and limitations. Discuss the importance of posture detection in various fields such as healthcare, ergonomics, and fitness.

**System Design:**

In the system design section, outline the architecture and components of your improper posture detection system. Describe the overall system flow, including data acquisition, preprocessing, feature extraction, classification, and feedback generation. Detail the hardware and software components required for system operation and discuss their roles in achieving accurate posture detection.

**Detailed Design:**

In the detailed design section, delve deeper into the implementation details of your system. Describe the algorithms, techniques, and methodologies used for posture detection, such as image processing, machine learning models, or sensor fusion techniques. Provide diagrams, flowcharts, or pseudocode to illustrate the functioning of different system modules and their interactions.

**Implementation and Results:**

We utilize CNNs, OpenPose, and MediaPipe models for posture detection. CNNs extract features for accurate posture recognition, while OpenPose estimates key body landmarks. MediaPipe offers efficient pre-trained models for real-time posture analysis, enhancing system versatility and performance.

**CHAPTER 2: LITERATURE SURVEY**

**1. Real-time Human Posture Recognition Based on Deep Learning with LSTM Network (Chen et al., 2018)**

This study explores a deep learning approach for real-time human posture recognition. The authors propose a method that combines convolutional neural networks (CNNs) for feature extraction and long short-term memory (LSTM) networks for capturing temporal dependencies in posture sequences. Their findings suggest that the LSTM network effectively improves recognition accuracy compared to CNNs alone.

**2. Research on Posture Recognition Using Convolutional Neural Networks (Liu et al., 2020)**

This research investigates the use of CNNs for posture recognition. The authors evaluate different CNN architectures and training strategies to achieve optimal performance. Their work highlights the effectiveness of CNNs in posture recognition tasks.

**3. Posture Recognition Using a Deep Learning Approach: Application to Elderly Fall Detection (Rahmani et al., 2017)**

This article explores the application of deep learning for posture recognition in the context of elderly fall detection. The authors develop a deep learning model to identify postures indicative of potential falls. Their research demonstrates the potential of deep learning for fall prevention systems.

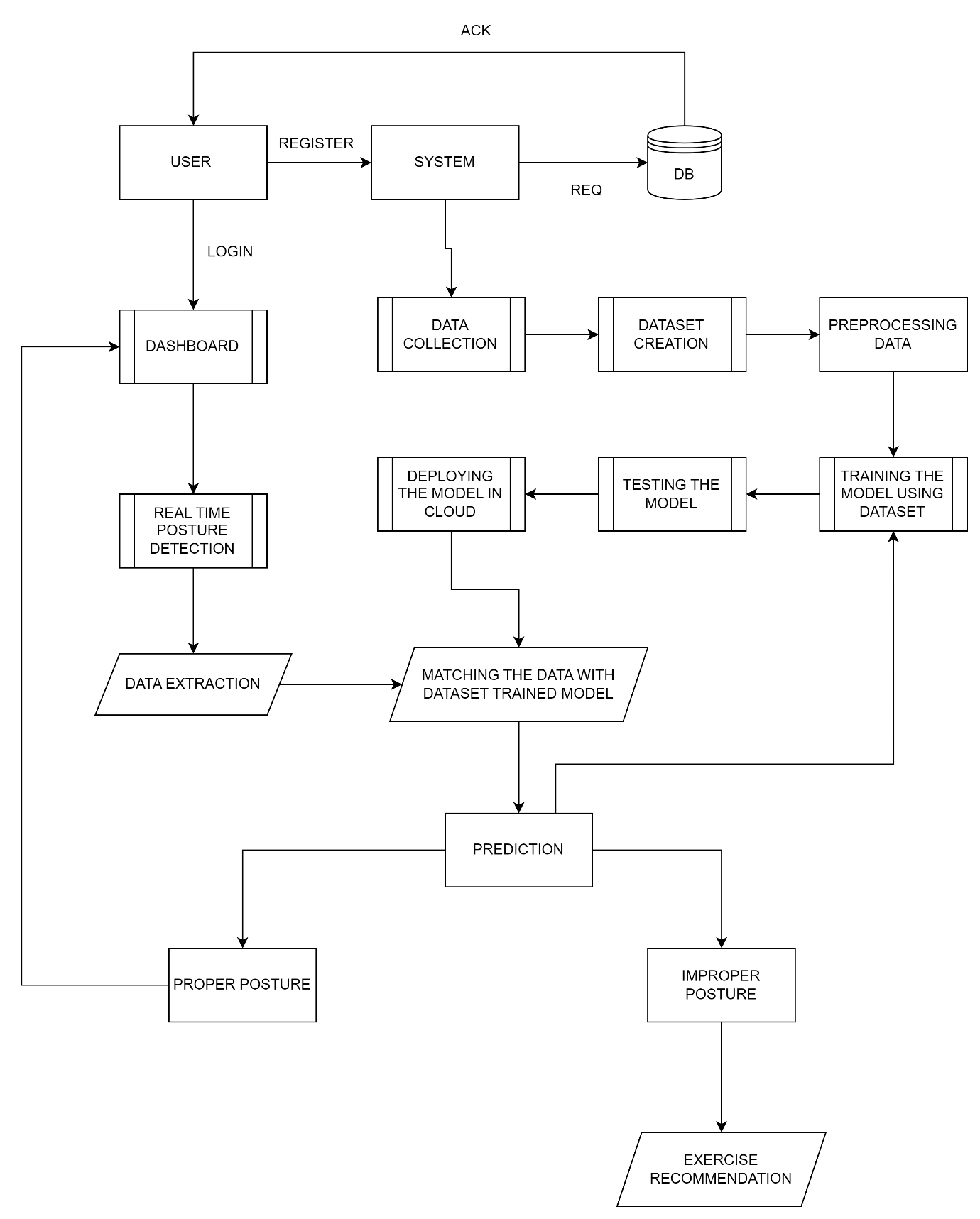
**4. An Adaptive and Efficient Human Posture Recognition System Based on Deep Residual Learning (Xia et al., 2020)**

This study proposes an adaptive and efficient human posture recognition system using deep residual learning. The authors introduce a deep residual network architecture that achieves high accuracy while maintaining computational efficiency. Their work contributes to developing real-time posture recognition systems with limited resources.

**CHAPTER 3: SYSTEM DESIGN**

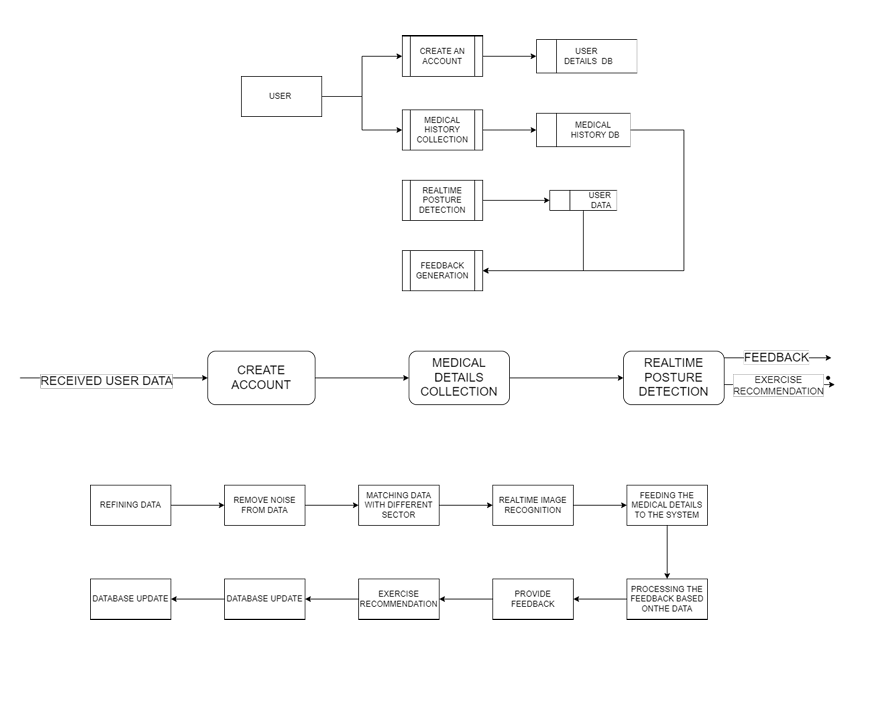
**3.1 ARCHITECTURE**

**3.1.1 Architecture Diagram**

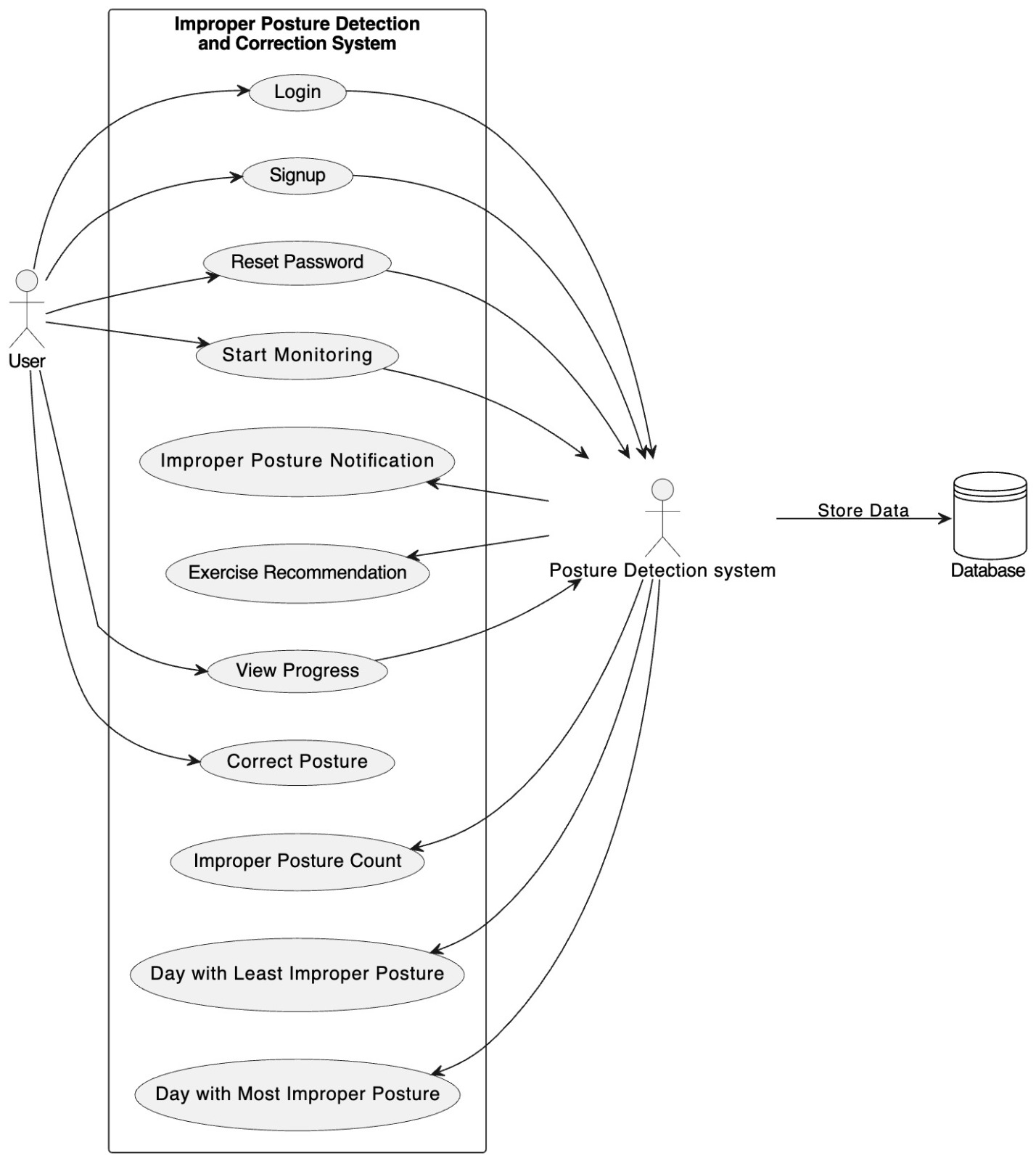


**3.1.2 Design Diagrams:**

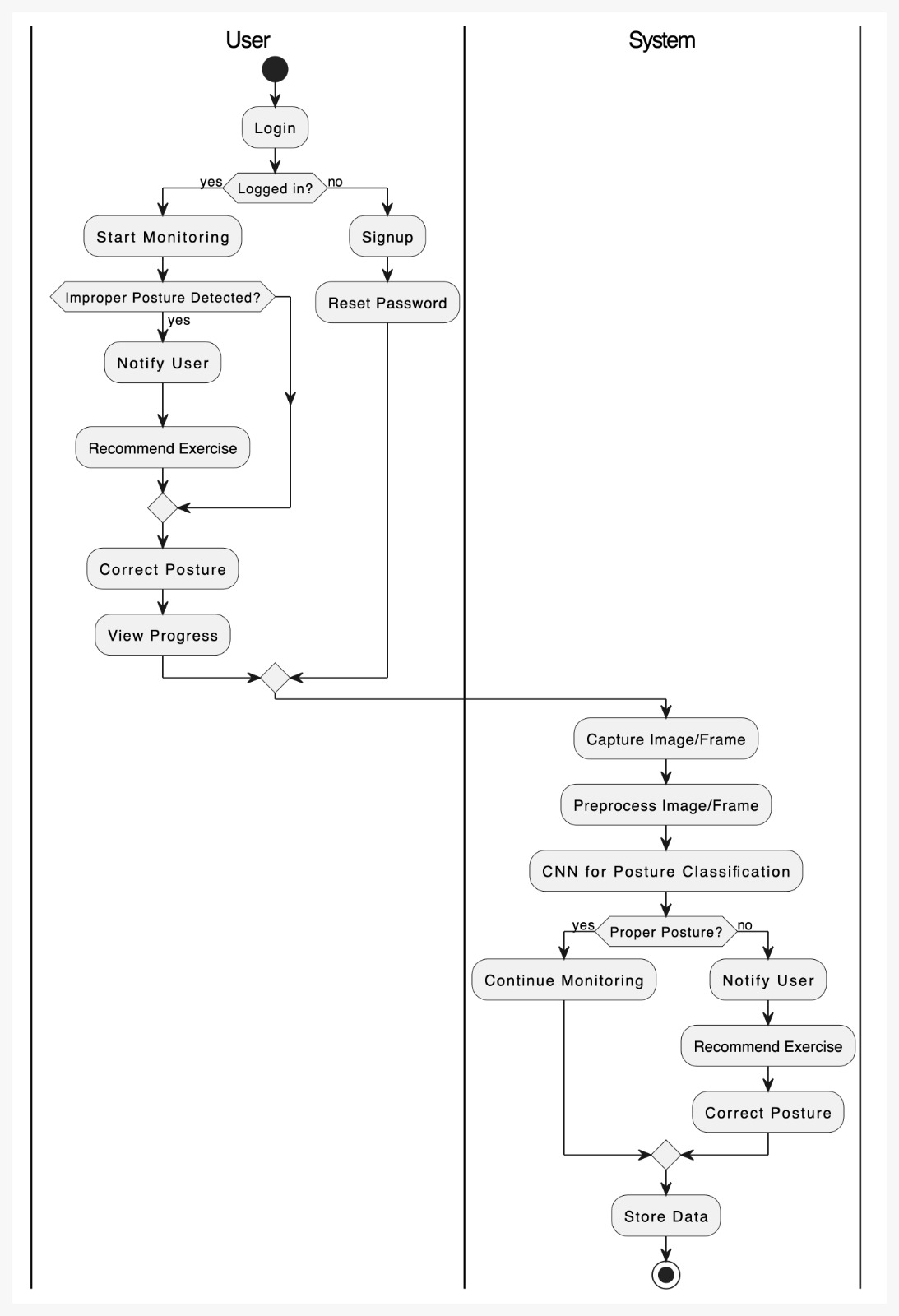
**3.1.2.1 DATA FLOW DIAGRAM:**

****

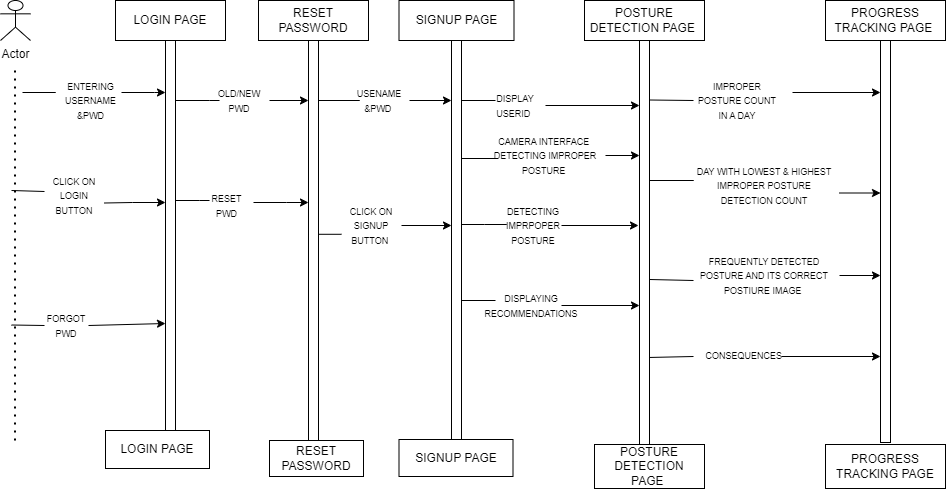
**3.1.2.2 USE CASE DIAGRAM:**



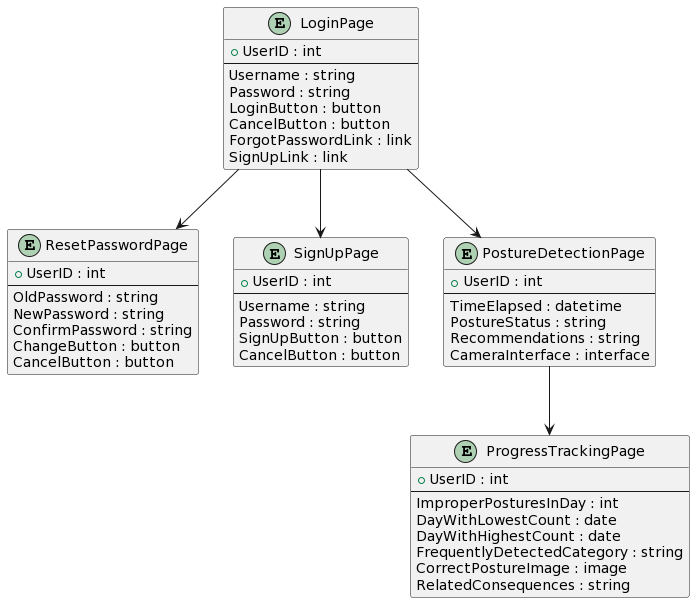
# 3.1.2.3 ACTIVITY DIAGRAM:



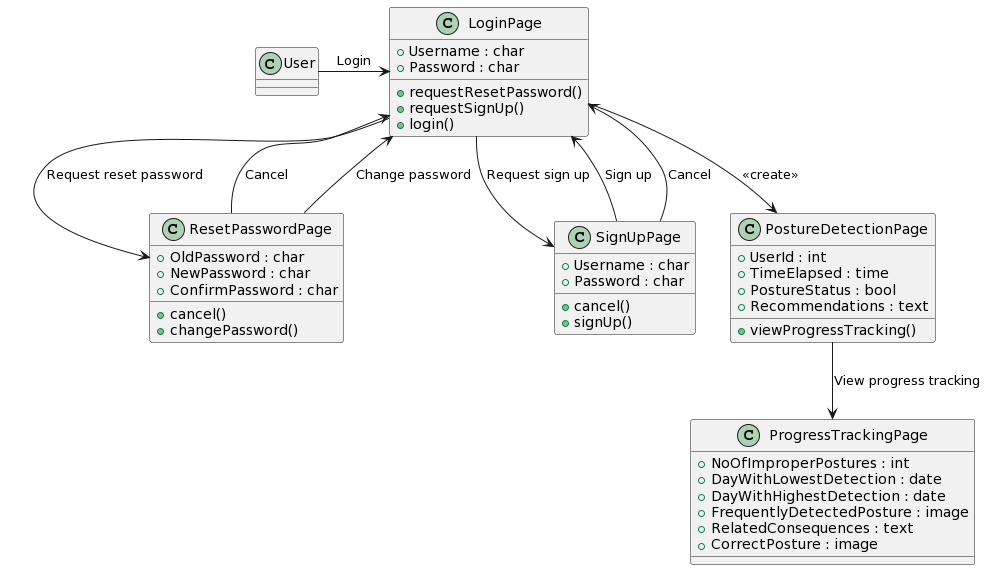
**3.1.2.4 SEQUENCE DIAGRAM:**



**3.1.2.5 DATA DESIGN DIAGRAM:**

****

**3.1.2.6 COLLABORATION DIAGRAM:**

****

**3.2 MODULES DESCRIPTION**

**3.2.1 Login Page:**

The login page provides secure access to the system, requiring users to input their credentials such as username and password. Upon successful authentication, users gain entry to their personalized accounts, enabling them to access system features and functionalities.

**3.2.2 Explore Page:**

The Explore page offers a comprehensive overview of the project, outlining its objectives and significance in promoting musculoskeletal health. Additionally, it showcases an image gallery highlighting key features and aspects of the application, providing visual context to visitors.

**3.2.3. Map Page:**

The map page displays a geographical map where the distribution of professionals across various fields is visualized through plotted data points. Each point represents the concentration of workers in a specific area.

**3.2.4 Tracking Page:**

The tracking page utilizes a real-time camera feed to detect and classify the user's posture (sitting, standing, lying, or bending) dynamically. It provides instant feedback on posture correctness, indicating the user's current position.

**3.2.5 Progress Tracking:**

The progress tracking page allows users to monitor their posture improvement journey over time. Users can track achievements, and receive personalized insights and recommendations based on their progress data, empowering them to take proactive steps towards better posture and well-being.

**CHAPTER 4: DETAILED DESIGN**

**4.1 MODULE:**

**4.1.1 Login Page (Security, User Experience)**

* **Input:** Username, Password
* **Process:**
  + Securely validates credentials against a user database or authentication service using hashing and salting techniques.
  + Implements two-factor authentication (2FA) for an extra layer of security (optional but highly recommended).
  + Enforces strong password policies, including minimum length, character complexity, and regular password changes.
  + Provides a user-friendly interface that facilitates easy login (e.g., remember username or password functionality).
  + Offers password reset options in case of forgotten credentials.
* **Output:**
  + Grants access to the user's personalized account upon successful authentication.
  + Delivers informative error messages in case of login failure, guiding the user towards a successful attempt.

**4.1.2 Explore Page (Engaging Content, Accessibility)**

* **Input:** None (directly from Login or landing page)
* **Process:**
  + Presents a clear and concise overview of the project's objectives and its importance in promoting musculoskeletal health.
  + Utilizes captivating visuals (e.g., infographics, animations) to engage users and enhance understanding.
  + Employs clear and concise language, catering to a broad audience.
  + Offers content in multiple languages (optional but recommended for wider accessibility).
  + Makes the page accessible to users with disabilities using appropriate coding techniques and assistive technologies.
* **Output:**
  + Educates and motivates users about the benefits of good posture and the app's role in achieving it.
  + Encourages users to explore the app's functionalities.

**4.1.3 Map Page (Visualization, Filtering)**

* **Input:** User's location (optional, for personalized view)
* **Process:**
  + Displays a geographical map using a user-friendly interface (e.g., interactive map libraries).
  + Visualizes the distribution of musculoskeletal health professionals across various fields using color-coded data points or markers.
  + Allows users to filter data by profession, location, or other relevant criteria (optional but improves usability).
  + Provides clear legends and tooltips to enhance understanding of the map's elements.
  + Integrates with real-time data sources (optional, for dynamic updates) to reflect current professional distributions.
* **Output:**
  + Empower users to discover nearby musculoskeletal health professionals relevant to their needs.
  + Provides insights into the availability of specialists in different regions.

**4.1.4 Tracking Page (Real-Time Feedback, Customization)**

* **Input:** Real-time camera feed capturing the user's posture
* **Process:**
  + Employs machine learning algorithms to accurately detect and classify the user's posture (sitting, standing, lying, bending) in real time.
  + Provide immediate feedback on posture correctness in a clear and actionable way (e.g., visual cues, audio notifications).
  + Offers personalized feedback and guidance based on the user's posture history, goals, and preferences (optional).
  + Allows users to customize feedback settings (e.g., frequency, type) for a more tailored experience (optional).
  + Implements privacy measures to ensure user data is anonymized and securely stored.
* **Output:**
  + Helps users maintain healthy posture by providing real-time corrective feedback.
  + Motivates users to improve their posture by offering personalized support and encouragement.

**4.1.5 Progress Tracking Page (Gamification, Long-Term Engagement)**

* **Input:** User's posture data from the Tracking Page
* **Process:**
  + Stores and analyzes user posture data over time.
  + Provides visual representations of progress, such as graphs or charts, to track improvements and celebrate achievements.
  + Integrates gamification elements (e.g., badges, points, leaderboards) to enhance user motivation and engagement (optional but effective).
  + Offers personalized recommendations based on progress data, suggesting exercises or adjustments tailored to individual needs.
  + Connects with wearable devices or health apps (optional) to collect additional data points and provide a more holistic view of user well-being.
* **Output:**
  + Empower users to monitor their posture improvement journey and witness the value of their efforts.
  + Offers actionable insights and recommendations to guide further progress.
  + Encourages sustained engagement with the app by fostering a sense of accomplishment and community (through gamification if implemented).

**CHAPTER 5: IMPLEMENTATION AND RESULT**

* 1. **HARDWARE AND SOFTWARE REQUIREMENTS:**

**5.1.1 Hardware:** A device with a camera (webcam or built-in camera on a smartphone/laptop/tablet)

**5.1.2 Software:**

* Programming language: Python (3. x recommended)
* Libraries:
  + OpenCV (for computer vision tasks)
  + NumPy (for numerical operations)
  + Scikit-learn (for machine learning tasks - optional for CNN approach)
  + TensorFlow or Keras (for deep learning models - CNN and PoseNet)
  + MediaPipe (for real-time pose estimation)
* Additional libraries (depending on the chosen approach):
  + Matplotlib (for visualization - optional)
  + Pillow (for image processing - optional)

**5.1.3 Software Requirements:**

* **OpenCV (Open Source Computer Vision Library):** A versatile library for real-time computer vision tasks. It provides functions for image processing, video capture, and object detection. OpenCV can be used for tasks like image preprocessing and visualization in this project.
* **NumPy (Numerical Python):** A fundamental library for scientific computing in Python. It offers efficient arrays and mathematical operations. NumPy is essential for working with image data and performing calculations in the project.
* **Scikit-learn (Machine Learning library):** Provides tools and algorithms for various machine learning tasks, including classification. While the document mentions using sci-kit-learn for the CNN approach, other deep-learning frameworks might not require it.
* **TensorFlow or Keras:** Both are popular deep learning frameworks. TensorFlow offers a powerful low-level API for building and training complex models. Keras provides a higher-level interface on top of TensorFlow, making it easier to build and experiment with deep learning models. This project can leverage either framework for building the CNN or PoseNet model.
* **MediaPipe:** An open-source framework from Google, specifically designed for building real-time applications that process audio, video, and sensor data. It offers pre-trained models for various tasks, including pose estimation. MediaPipe can be a good choice for real-time posture detection due to its pre-trained model and efficient processing capabilities.
* **Matplotlib:** A library for creating static, animated, and interactive visualizations in Python. While not essential for core functionality, Matplotlib can help visualize results during development and debugging.

**5.2CLASSIFICATION AND CLUSTERING TOOL:**

**CLASSIFICATION :**

**5.2.1 Convolutional Neural Networks (CNNs)** have emerged as a powerful tool for posture detection tasks due to their exceptional ability to learn intricate patterns from image data. This in-depth exploration delves into their inner workings, highlighting how they excel in extracting features and performing posture classification.A single neuron for binary classification (proper vs. improper posture)Multiple neurons for multi-class classification (identifying specific postures, like standing, sitting, or bending)

Data Preparation: The journey begins with the raw materials – images depicting various postures. These images undergo preprocessing, ensuring consistency for analysis. This might involve resizing to a standard dimension and normalizing pixel values (like adjusting brightness levels).

Model Architecture: The CNN architecture is meticulously designed to automatically extract hierarchical features from the images. It often comprises multiple convolutional layers followed by pooling layers for spatial feature extraction, culminating in fully connected layers for classification. Think of it as a meticulously designed factory pipeline, where each stage transforms the raw data into a refined posture classification.

Training: The Learning Phase: During training, the CNN undergoes a rigorous learning process. It progressively refines its internal parameters (weights and biases) through a technique called backpropagation. This iterative process involves adjusting these parameters to minimize errors and enhance its ability to map input images to their corresponding posture labels. Imagine the detective meticulously studying past cases and refining their detection skills.

Feature Extraction: Unveiling the Secrets Once trained, the CNN becomes adept at extracting informative features from new, unseen images. These features, at varying levels of abstraction, hold the key to posture classification. Lower layers capture fundamental features like edges and textures, while higher layers delve into more abstract posture-related details, such as body orientation and joint positioning.

Classification: The Verdict The extracted features are fed into the fully connected layers for the final act – classification. The output layer generates probability scores or class predictions, indicating the likelihood of each posture class. In binary classification, the model distinguishes between proper and improper posture, while multi-class classification pinpoints the specific posture category.

Inference: Putting Knowledge to Use The trained CNN is now equipped to classify unseen images. The input image traverses the same preprocessing steps and is then fed into the CNN, which delivers the predicted posture label. Imagine the detective confidently analyzing a new crime scene, drawing upon their honed skills.

**5.2.2 Mediapipe:**

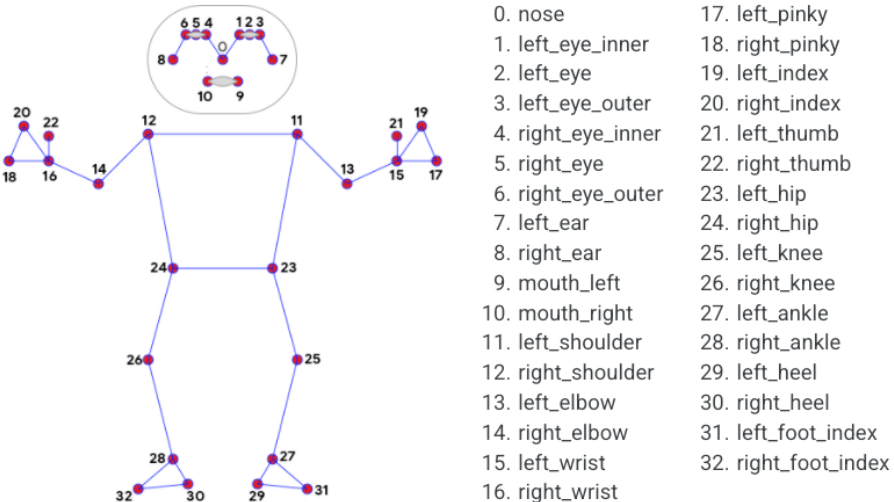
MediaPipe, a brainchild of Google, is an open-source framework designed to streamline the development of real-time applications that process various data streams, including audio, video, and sensor data. Its strength lies in providing pre-trained models and efficient pipelines, empowering developers to create sophisticated real-time applications with ease.

MediaPipe in Action: Unveiling Posture

In the realm of posture detection, MediaPipe shines as a powerful tool. It excels at a computer vision task called pose estimation, where it predicts human poses in images or videos. This is achieved by pinpointing key body joints (landmarks) like elbows, shoulders, and knees.

MediaPipe offers :

* Real-time Processing: It predicts 33 3D landmarks on a human body in real time, even on devices with CPUs. This is ideal for applications requiring immediate feedback on posture.
* High Accuracy: Despite its real-time capabilities, MediaPipe delivers high accuracy in its pose estimations.
* Lightweight and Efficient: The two-step machine learning pipeline keeps the processing demands lightweight, making it suitable for various platforms.



Key Code Components Involved In Posture Classification Using Mediapipe:

1. Essential Libraries: The code starts by importing necessary libraries like OpenCV for image processing and MediaPipe for pose estimation.
2. MediaPipe Pose Setup: The MediaPipe Pose class is initialized to handle pose detection tasks. Parameters like static\_image\_mode are set to indicate working with static images and min\_detection\_confidence to ensure reliable detections.
3. Drawing Class Initialization: The MediaPipe drawing class is used to visualize the detected pose landmarks on the image.
4. Calculate Angle Function: This function plays a crucial role in posture classification. It calculates the angle between three landmark points, providing valuable insights into body posture.
5. Classify Pose Function: This function analyzes the angles formed by shoulders, hips, and ankles (calculated using the calculateAngle function) to classify the posture. It defines specific angle conditions indicative of "Correct Posture" and classifies poses based on these conditions. If the angles deviate significantly from these ranges, the posture is deemed "Incorrect." The classified label is then displayed on the image.
6. Detect Pose Function: This function is the workhorse for pose detection. It takes an image as input and performs the following:
   * Detects human poses using the MediaPipe pose setup.
   * Classifies the detected posture using the classifyPose function.
   * Optionally displays the original image, the annotated image with landmarks, and a 3D plot of the detected landmarks.
7. Input Images for Classification: This part of the code demonstrates reading an image, detecting poses, and classifying them using the functions defined earlier. If successful, it displays the image with classified posture labels.

MediaPipe's Advantages for Posture Classification:

* Real-time Performance: Delivers immediate feedback on posture, making it suitable for applications like posture correction exercises.
* Ease of Use: Pre-trained models and efficient pipelines simplify development.
* Lightweight and Efficient: Runs well on various platforms, including devices with limited processing power.
* High Accuracy: Provides reliable pose estimations despite real-time processing.

**5.2.3 Openpose:**

OpenPose, while not exclusively designed for posture classification, can be effectively used for this purpose due to its strengths in keypoint detection and pose estimation. Here's how it approaches posture classification:

Keypoint Analysis:

* Rich Keypoint Data: OpenPose excels at detecting a large number of key points (typically 135) across the body. This comprehensive data provides a detailed picture of the body's posture.
* Posture Definition Through Angles: By analyzing the relative positions and angles formed between key points (e.g., shoulder, elbow, wrist angles), OpenPose can infer aspects of posture like slouching or imbalanced weight distribution.

Classification Workflow:

1. Thresholding: Specific thresholds are set for angles or keypoint positions. These thresholds define what constitutes a "good" or "bad" posture for specific body parts or the overall pose.
2. Comparison: The detected keypoint positions and angles are compared to the defined thresholds.
3. Classification: Based on the comparison, OpenPose classifies the posture as "good," "bad," or a specific category depending on the defined thresholds (e.g., "rounded shoulders," "swayback").

Code Implementation (Conceptual):

* Posture Classification Function: This function would take the detected key points as input and compare them to pre-defined thresholds for various postures.
* Threshold Definition: A separate section of code would define thresholds for angles or keypoint positions relevant to different posture classifications.

Advantages of OpenPose for Posture Classification:

* Accuracy: With a large number of key points, OpenPose offers a more nuanced understanding of posture compared to models detecting fewer points.
* Customizable Thresholds: You can tailor the classification criteria to your specific needs by adjusting the posture thresholds.
* Multi-Person Capability: OpenPose can handle multiple people in a scene, making it suitable for group posture analysis.

Limitations of OpenPose as a Classification Tool:

* Complexity: Implementing posture classification from scratch using OpenPose requires more development effort compared to pre-trained posture classification models.
* Computational Cost: Processing a large number of key points can be computationally expensive, especially for real-time applications.
* Threshold Tuning: Defining appropriate thresholds for accurate classification can be an iterative process and may require domain expertise related to posture analysis.

**5.2.4 Posenet:**

* Core Functionality: Pose Estimation - PoseNet excels at this task. It analyzes images or videos, detects key body joints (like shoulders, and elbows), and creates a skeletal structure representing the person's pose.
* Indirect Role in Classification - While PoseNet doesn't directly classify postures itself, it provides the foundation for classification. By identifying key points and their locations, it lays the groundwork for analyzing posture.

The Classification Process:

1. PoseNet Detects Key Points: It identifies key body joints like shoulders, elbows, and hips.
2. Feature Extraction: Based on these points, features are calculated. These features could be angles between joints or distances between specific points.
3. Posture Classification (Separate Model): A separate machine learning model, trained for posture classification, takes these features as input. This model then classifies the posture based on the features (e.g., good posture, slouching).

PoseNet provides the essential building blocks (keypoint detection and feature extraction) that can be used for posture classification with additional machine learning models.

**5.2.5 Comparison Table:**

**CNN (Convolutional Neural Network):** Custom-built model trained on the dataset. Offers high accuracy but moderate inference speed. It's moderately easy to use but lacks real-time capabilities.

**MediaPipe:** Utilizes the MediaPipe framework for posture detection. Although it doesn't require explicit training data, it provides high inference speed and moderate accuracy. It's highly user-friendly and supports real-time capabilities.

**OpenPose:** This relies on the OpenPose framework and requires a dataset for training. It offers high accuracy but low inference speed. OpenPose has moderate ease of use and lacks real-time capabilities.

**PoseNet:** Implemented using TensorFlow.js and trained on dataset. It provides high inference speed and moderate accuracy, with high ease of use.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Framework | Training Data | Inference Speed | Accuracy | Ease of  Capabilities | Real-Time Capabilities |
| CNN | Custom | Dataset | Medium | High | Moderate | No |
| Media pipe | Mediapipe | N/A | High | Moderate | High | Yes |
| Open  Pose | Open Pose | Dataset | Low | High | Moderate | No |
| Pose Net | Tensor flow | Dataset | High | Moderate | High | Yes |

After comparing the models, it's evident that each model has its advantages and limitations. However, for this project's specific requirements, MediaPipe stands out as the best choice. It offers a balanced combination of high inference speed, moderate accuracy, ease of use, and real-time capabilities. Additionally, its integration with the MediaPipe framework simplifies development and deployment processes. Therefore, MediaPipe emerges as the optimal solution for posture classification in this project.

**CLUSTERING TOOLS:**

**5.2.6 CONVOLUTIONAL NEURAL NETWORKS (CNNS):**

* Function: Supervised learning models that excel at image recognition tasks.
* Posture Detection Role: Primary approach for posture detection.
* Process:
  + Trained on a large dataset of labeled posture images (good posture, bad posture, etc.).
  + Learned to recognize posture-related features in the images (e.g., joint angles, body alignment).
  + Can classify new unseen images into posture categories based on the learned features.

**5.2.7 K-MEANS CLUSTERING:**

* Function: Unsupervised learning technique for grouping similar data points.
* Posture Detection Role: Limited role, potentially as a preparatory step.
* Process:
  + Groups posture images together based on similarities in their pixel intensities (not posture itself).
  + Doesn't inherently tell you what postures the clusters represent.
  + Might be useful for initial data exploration, but further analysis is needed.

**5.2.8 HIERARCHICAL CLUSTERING:**

* Function: Similar to K-means, it's an unsupervised learning technique for data clustering.
* Posture Detection Role: Limited role, similar to K-means.
* Process:
  + Creates a hierarchy of clusters where similar images are grouped at lower levels and more dissimilar ones are grouped at higher levels.
  + Like K-means, doesn't provide labels for postures.
  + Might be helpful for data exploration, but not ideal for direct posture classification.

CNNs are the workhorse for posture detection. They learn from labeled data and directly classify postures.K-means and Hierarchical clustering can be exploratory tools in the early stages of posture analysis, but they require further steps (like labeling data) to be relevant for posture classification

**5.3 RESULTS AND DISCUSSIONS:**

**5.3.1 PROJECT OUTCOME**

Real-time posture detection leverages OpenCV and MediaPipe for efficient processing of camera streams. OpenCV facilitates camera input handling and basic computer vision tasks, while MediaPipe offers pre-trained pose estimation models. Integration of these tools enables real-time analysis of body posture, vital for applications like fitness monitoring and ergonomic assessment**.**

**Sitting:**

Correct posture



Incorrect posture



**Standing:**

Correct posture

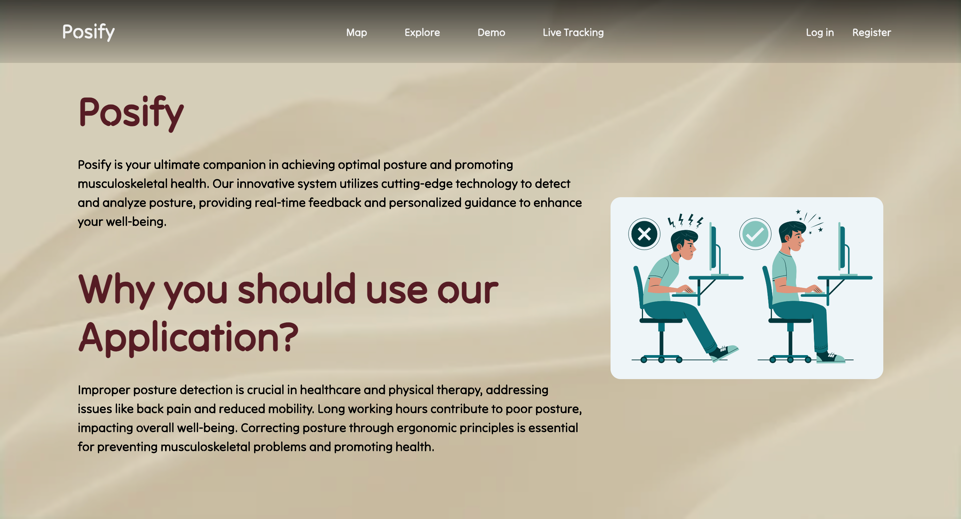


Incorrect posture

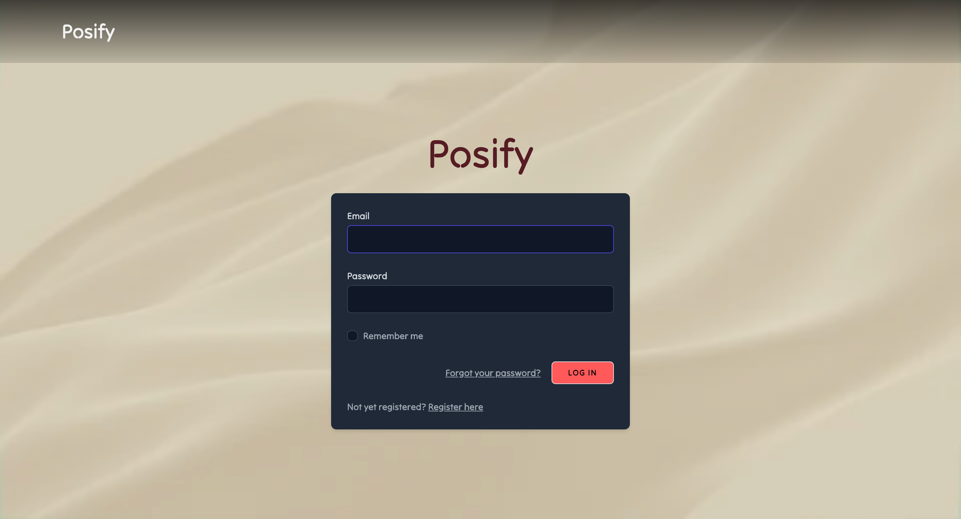


**5.3.2 Web Pages Screenshots**

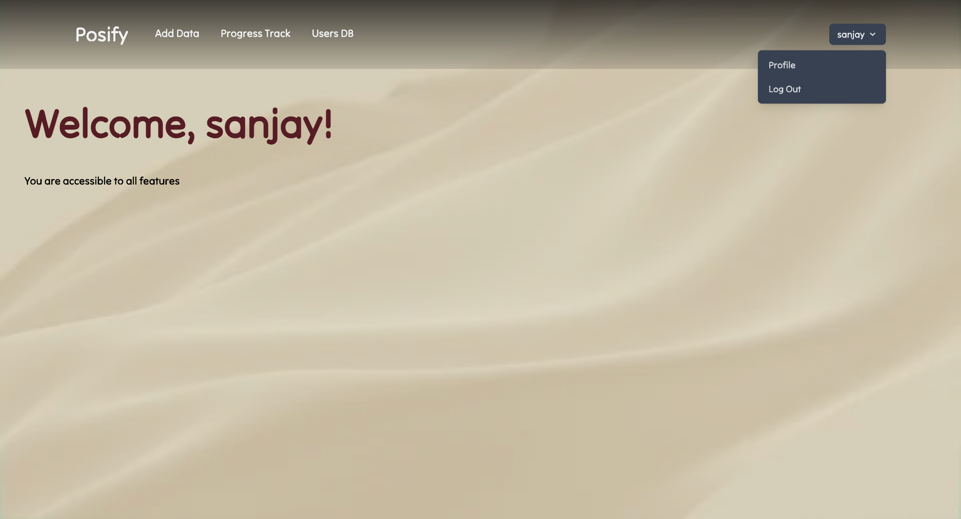
**Index page:**

****

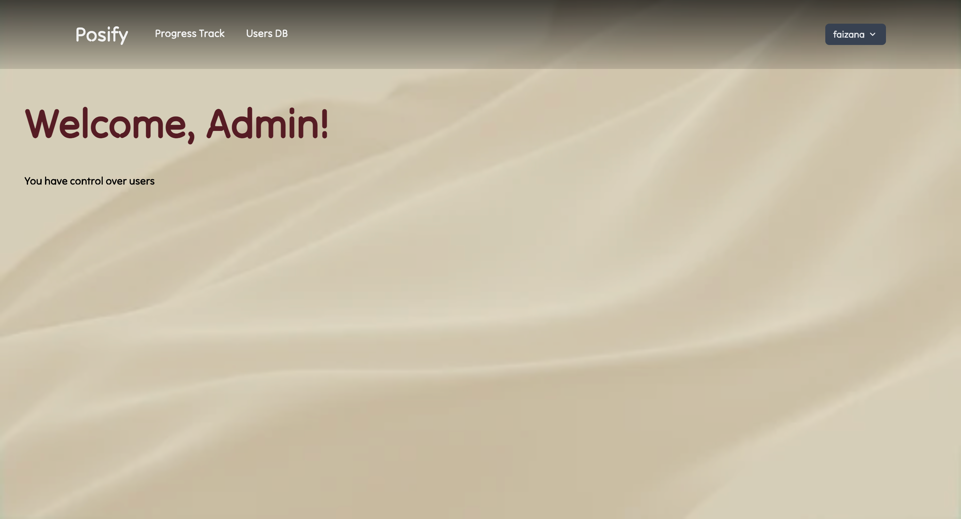
**Login page:**

****

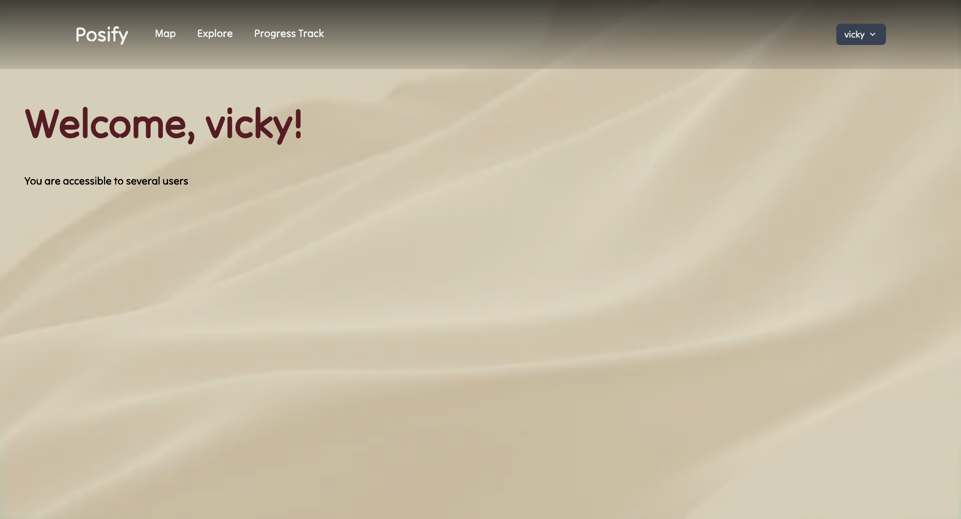
**SuperAdmin page:**

****

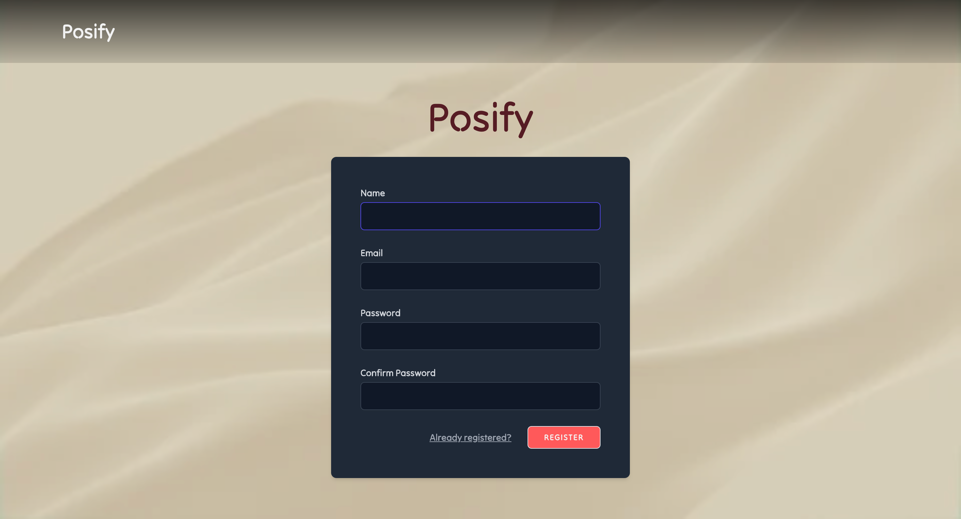
**Admin page:**

****

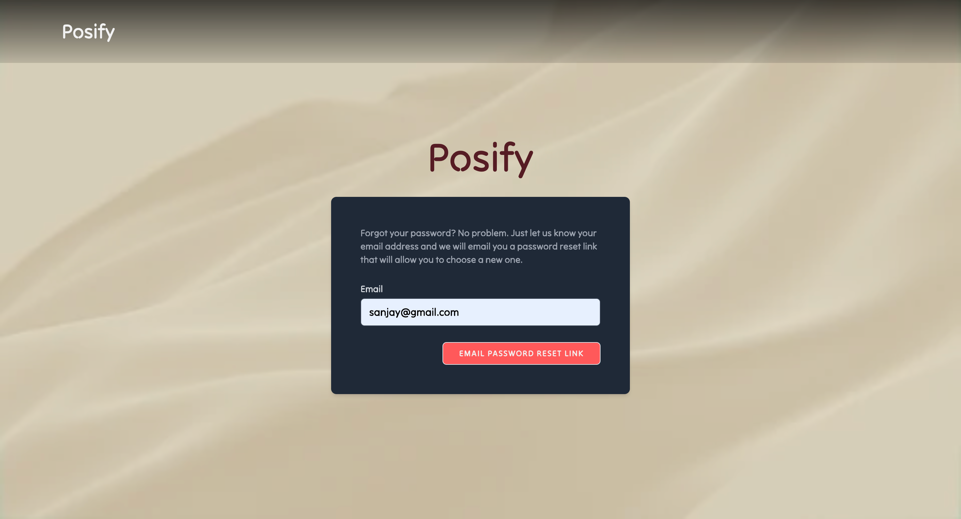
**User page:**

****

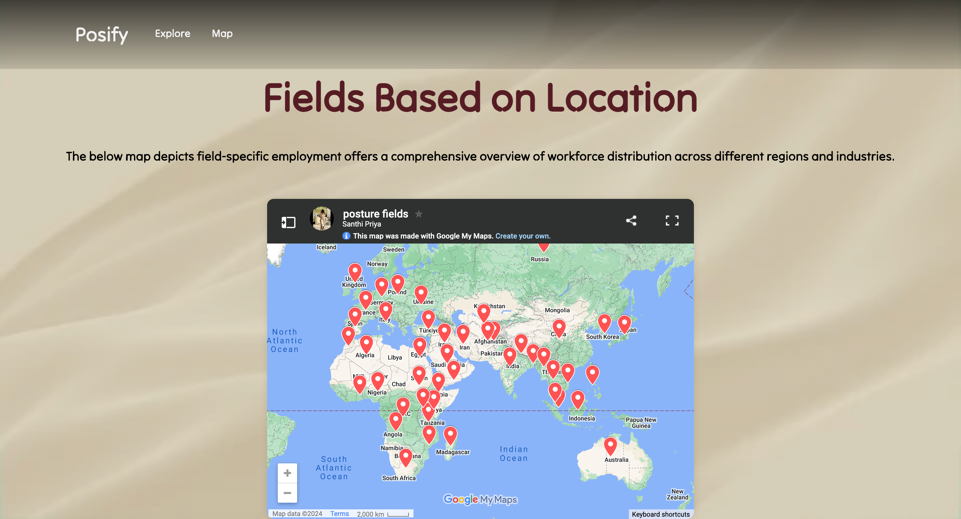
**Signup page:**

****

**Forgot password page:**

****

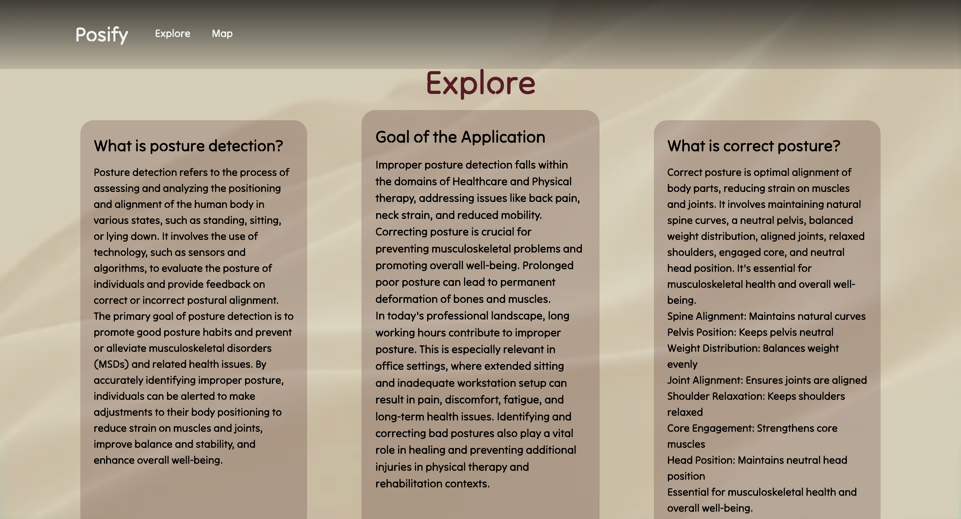
**Map page:**

****

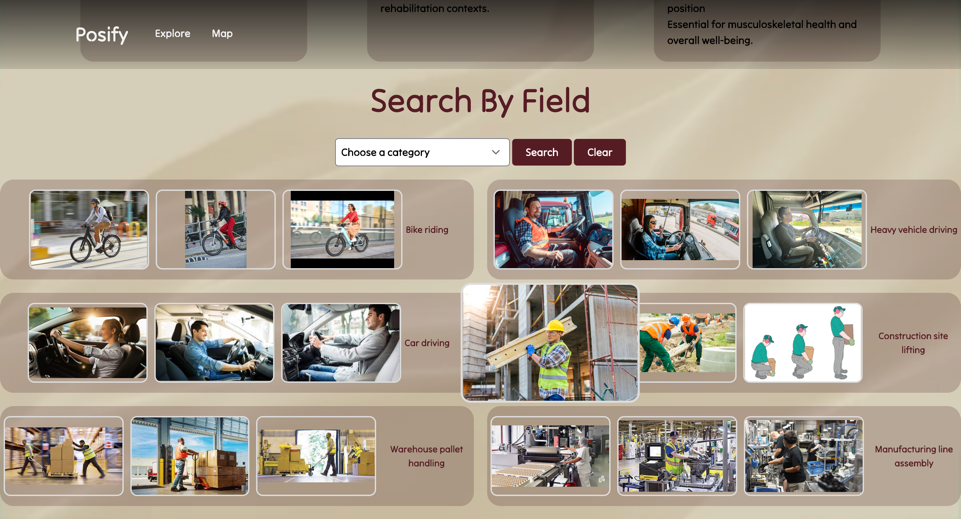
**Map interface:**

****

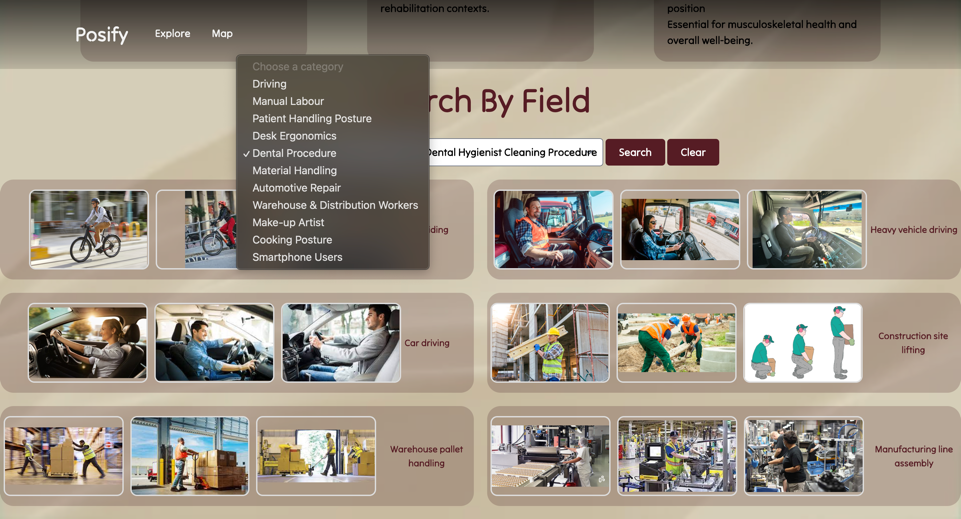
**Explore Page:**

****

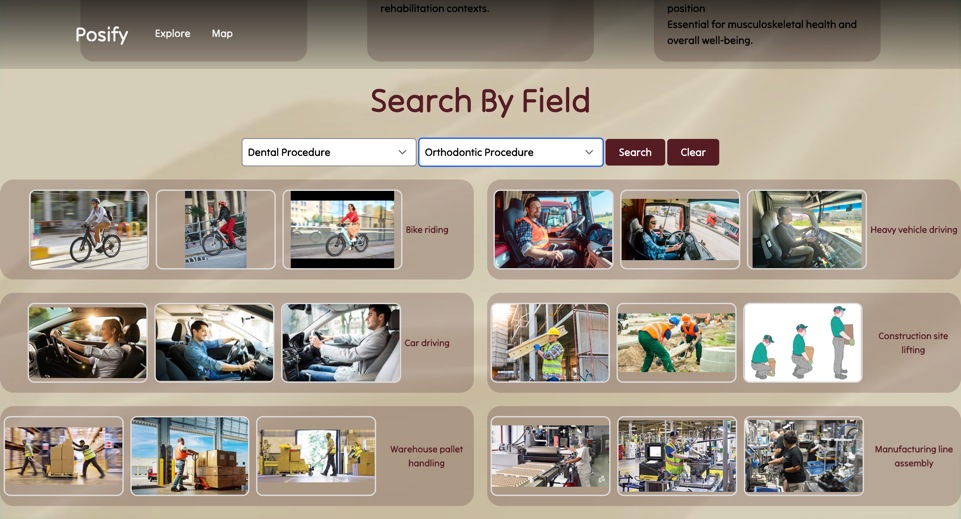
**Search field:**

****

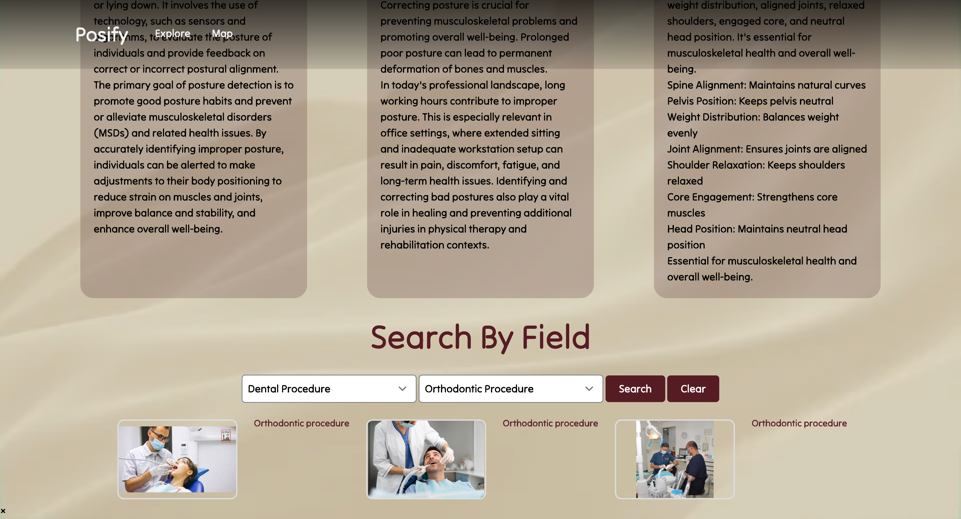
**Category listdown:**

****

**Sub-Category listdown:**

****

**Search result:**

****

**CHAPTER 6: CONCLUSION**

**6.1 CONCLUSION:**

This project proposes a socially relevant mobile application that addresses the growing concern of improper posture and its associated health risks. The application leverages machine learning to detect and correct posture in real time, offering personalized recommendations for individuals across diverse work environments and daily activities. This application can significantly improve public health by promoting good posture and preventing musculoskeletal problems. It caters to a wide range of users, including office workers, manual laborers, healthcare professionals, drivers, and individuals engaged in various physical tasks. The application offers comprehensive guidance for maintaining good posture in everyday scenarios.

**6.2 FUTURE ENHANCEMENT:**

Future discussions on posture detection will delve into deeper machine learning for complex movements and explore pre-trained models for efficiency. Integrating data from wearables and biofeedback mechanisms can provide a more nuanced picture. Imagine smart environments that adjust to your posture or gamified challenges that motivate you to stand tall. Addressing privacy concerns and tailoring the app to specific needs are also crucial. Ultimately, discussions will explore the long-term impact and ensure the app reaches a broad audience.

* 1. **REFERENCES:**

1. Chen, X., Jiang, Y., Hu, C., Li, Z., & Wang, X. (2018). Real-time human posture recognition based on deep learning with LSTM network. Sensors (Switzerland), 18(8), 2791. (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6148738/>)
2. Liu, M., Wang, S., Liu, H., & Wang, C. (2020). Research on posture recognition using convolutional neural networks. IEEE Access, 8, 80038-80047.
3. Rahmani, H., Athmani, A., & Dagoune, A. (2017). Posture recognition using a deep learning approach: Application to elderly fall detection. Computer Vision and Image Processing (ECCV), 10661, 668-683.
4. Xia, L., Chen, X., Wang, J., Liao, Y., & Li, J. (2020). An adaptive and efficient human posture recognition system based on deep residual learning. Sensors (Switzerland), 20(15), 4375. (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7511572/>)
5. Boutell, C., Aminian, K., Looze, M. P., Shull, P. B., Cleeland, C. S., & Fairbanks, J. G. (2008). The London back study: objective assessment of spinal posture in office workers. Spine (Philadelphia, Pa. 1976), 33(24), 2594-2600. (<https://pubmed.ncbi.nlm.nih.gov/19020346/>)
6. Kim, J., Park, C., Kim, J., & Jeong, Y. (2018). Development of a wireless posture monitoring system using a single inertial measurement unit. Sensors (Switzerland), 18(11), 3704. [(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6272285/)]((https:/www.ncbi.nlm.nih.gov/pmc/articles/PMC6272285/))
7. Muenchen, R., Slayter, J. M., Hodgins, J. K., & Erskine, J. L. (2013). Identifying posture transitions using wireless inertial sensor networks on the back. Journal of Biomechanics, 46(12), 2143-2150[. (https://pubmed.ncbi.nlm.nih.gov/23850004/)](.%20(https:/pubmed.ncbi.nlm.nih.gov/23850004/))
8. Chen, S., Wang, J., Wang, H., & Li, Q. (2016). A smartphone-based posture recognition system for preventing musculoskeletal disorders. Personal and Ubiquitous Computing, 20(5), 987-1000.

WEBSITE REFERRED:

<https://learnopencv.com/building-a-body-posture-analysis-system-using-mediapipe/>