

Final Project Manual

Personal Study Posture Corrector

GROUP 7

Songhyun Kim
Seungjae Lim
Somin Lee



Final Project Manual

Project Intro.

- Motivation
- Project Intro
- Key Features



Development Process

- Development Environment
- Software Programming
- Hardware and mobility

Demo Video

- Hardware Settings
- Demo Video
- Web App

Discussion

- Evaluation
- Expectation
- Improvement Points
- Conclusion

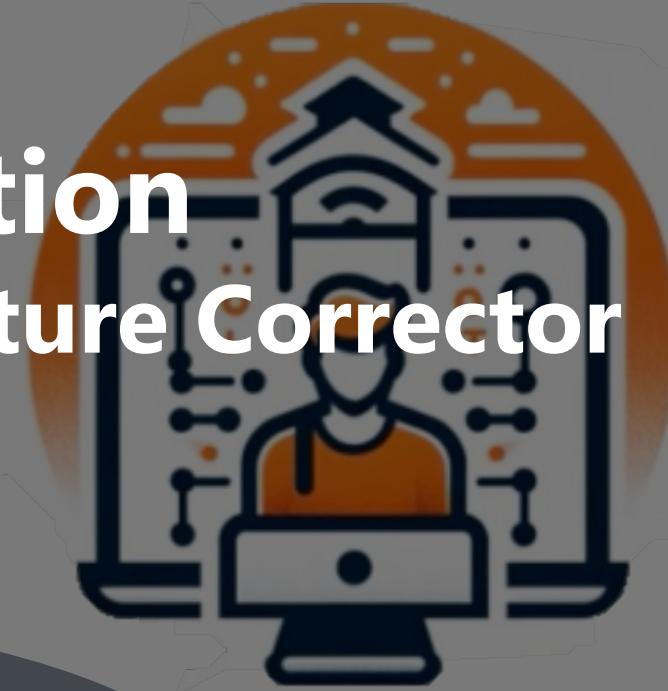
Project Introduction

: Personal Study Posture Corrector

Motivation

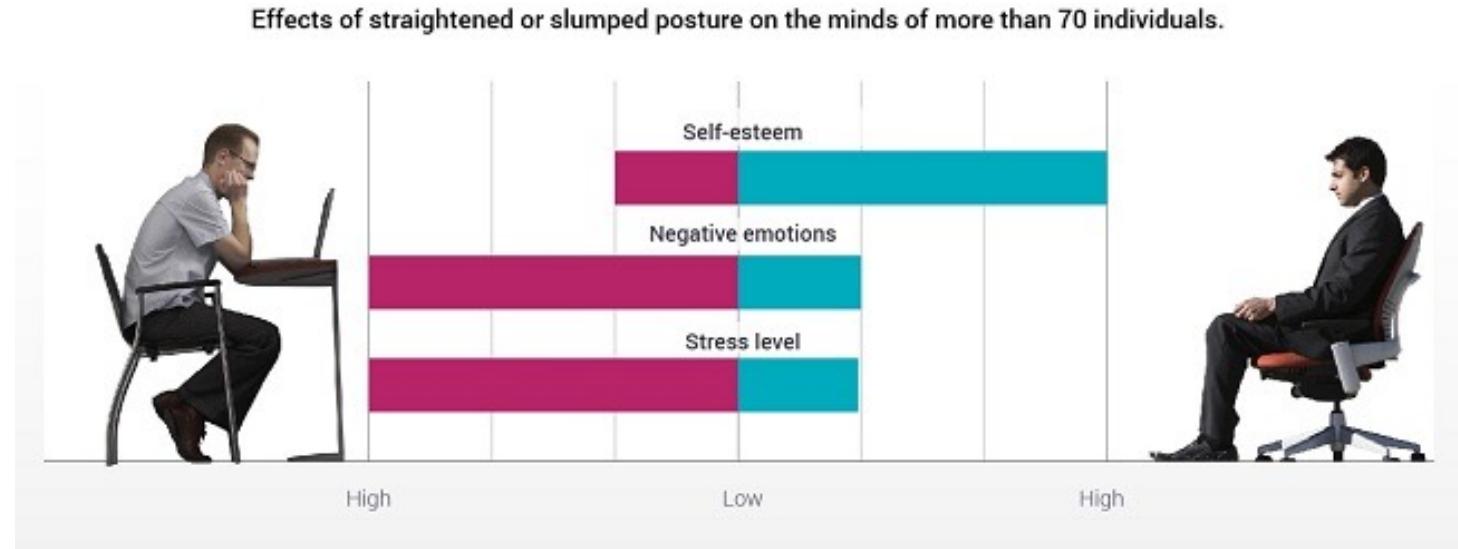
Project Introduction

Key Features



1. Motivation

- Problems with student's bad study postures

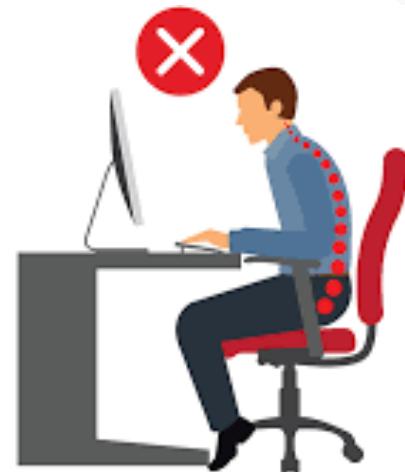


Lee, Myoung-hee, et al. "The effects of smartphone use on upper extremity muscle activity and pain threshold." Journal of Physical Therapy Science, vol. 26, no. 6, 2014, pp. 941–944, doi:10.1589/jpts.26.941.

2. Project Introduction



+



Learning
Management

Posture Care

2. Project Introduction

SmartPosture; Personal Study Posture Corrector(PSPC)

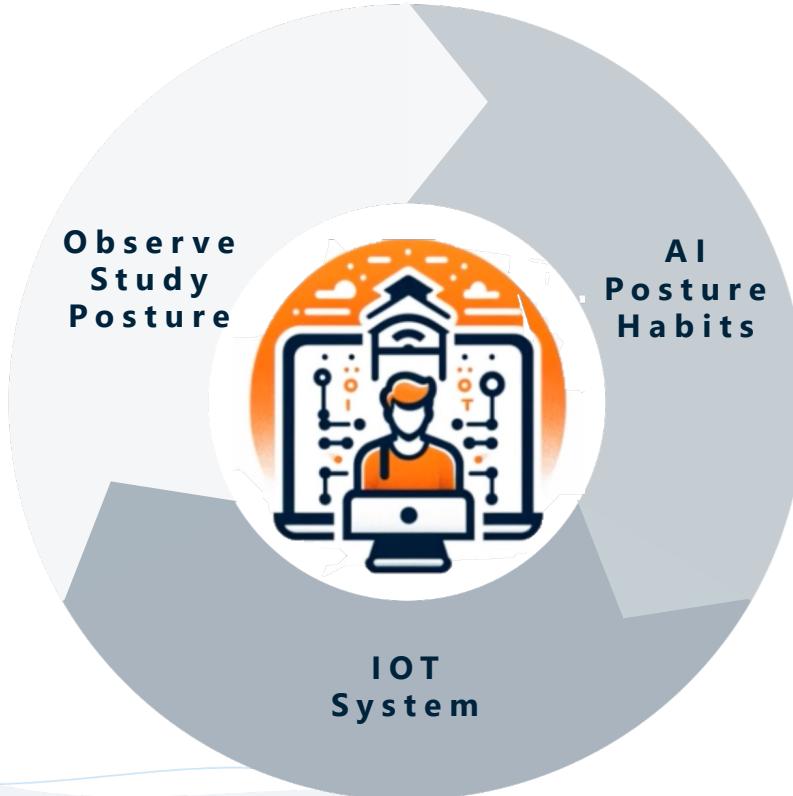
- provide personalized posture care for individuals during their study sessions
- Monitors, records, and advises users on maintaining correct posture



3. Key Features

Measure time how long the user studied in the right posture

; Using computer vision(ResNet)



Analysis and calibration of AI posture habits

; Using Doctor GPT(LLM)

Development Process : Development Environment

Diagram for Overall
Architecture

Explanation of
Development Environment



1. Diagram for Overall Architecture

Deploy

; NGINX, Github Pages

WebApp

; React

API Server

; FastAPI, MySQL, Docker

Posture Vision Model

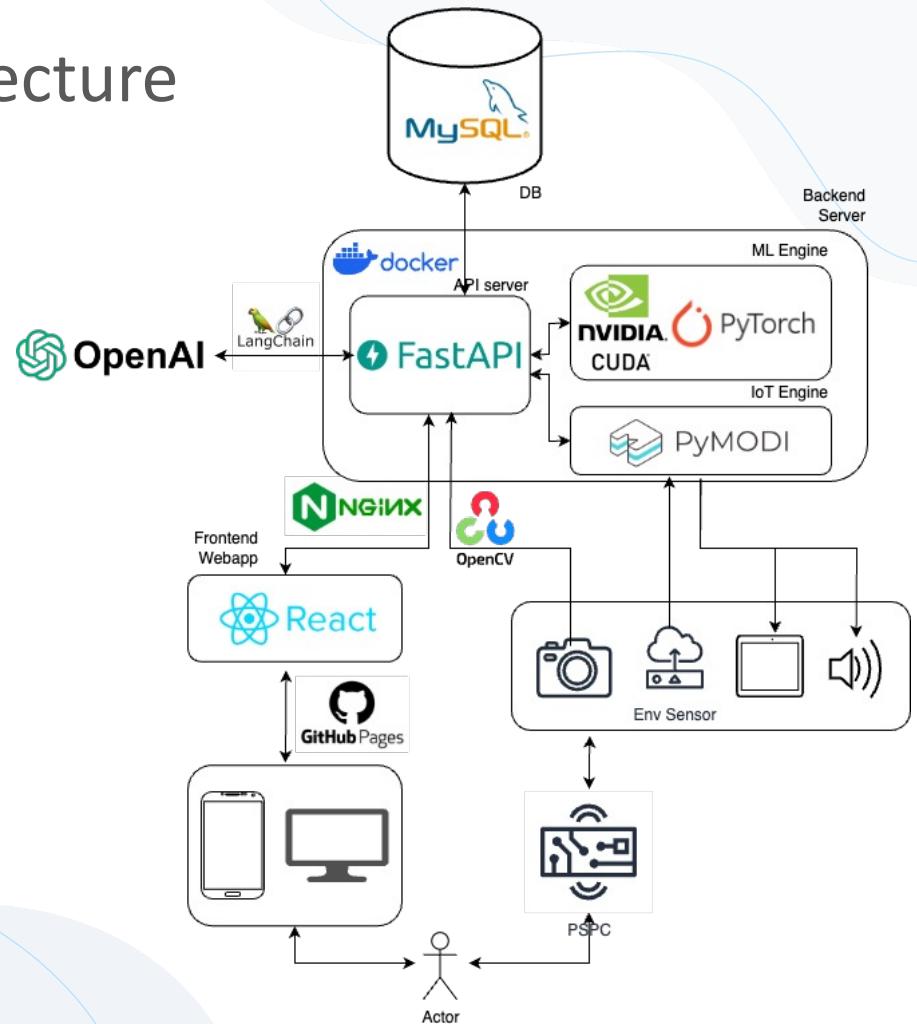
; Pytorch, CUDA, OpenCV

Doctor AI

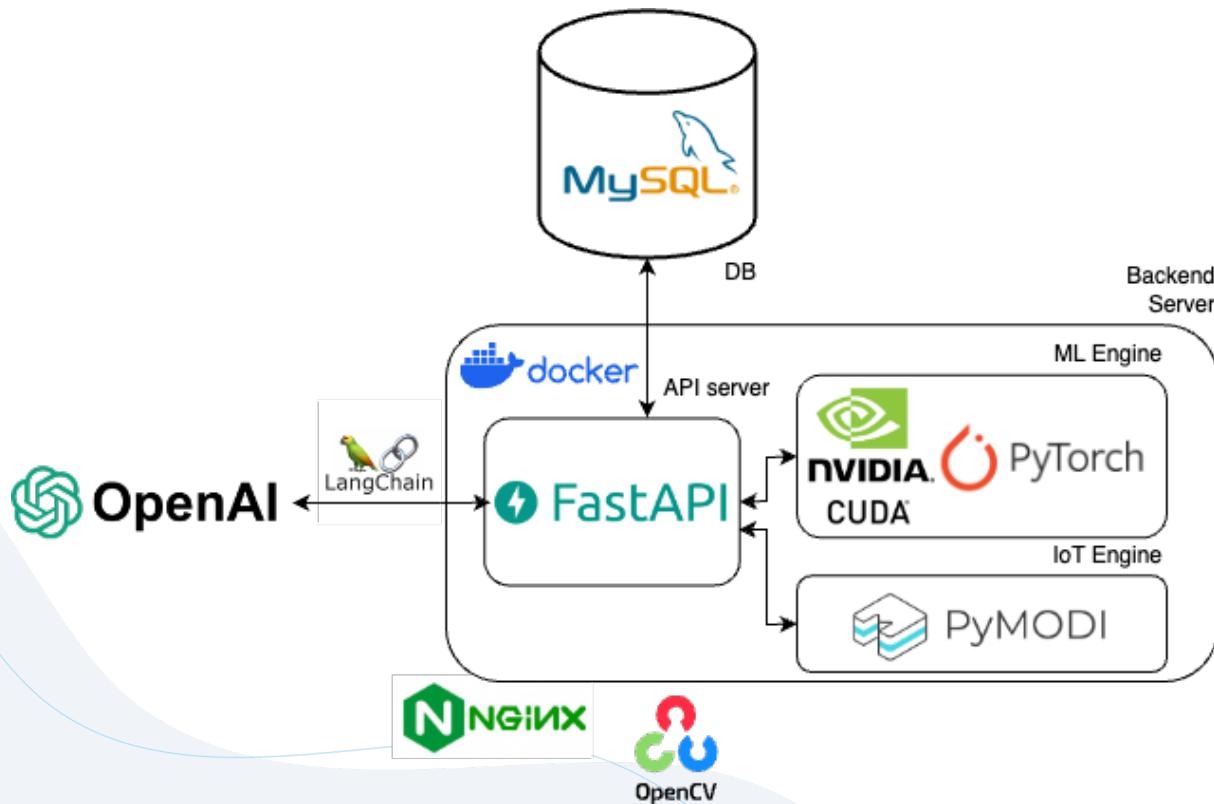
; Langchain, openai api

IOT

; PyMODI

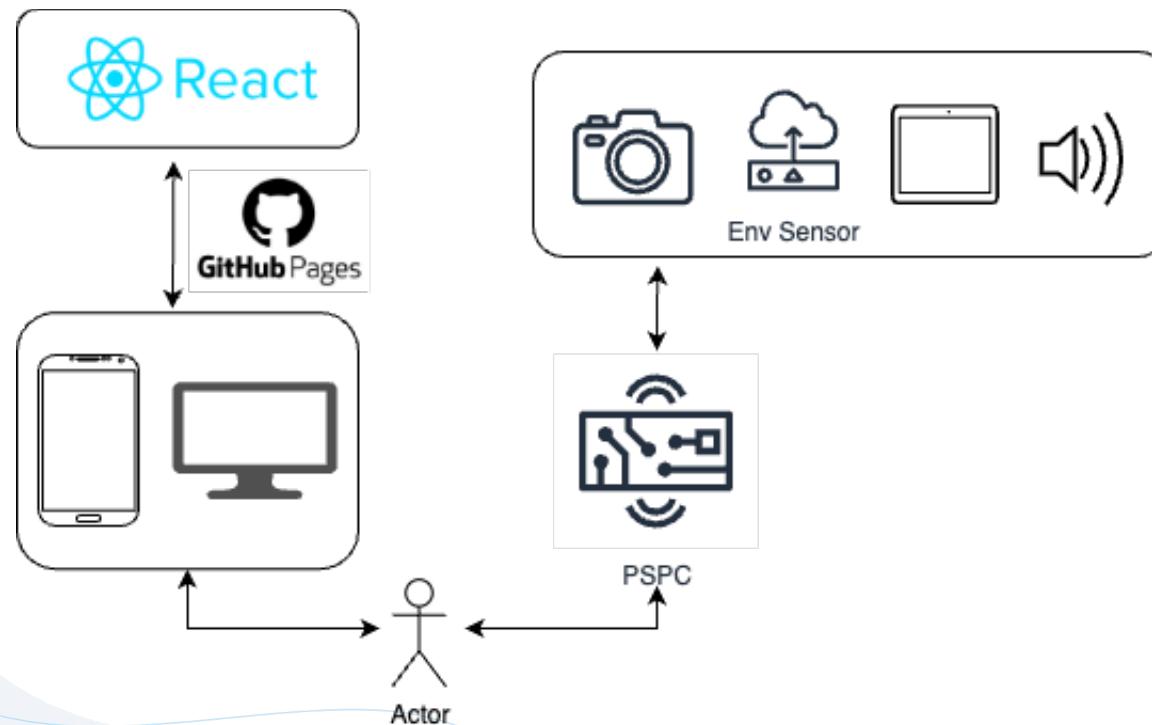


2. Explanation of Development Environment - Backend



Detailed Explanations in GitHub

2. Explanation of Development Environment - Frontend



Detailed Explanations in GitHub

Development Process : Software Programming

Posture Detect Model

Large Language Model

Code Structure

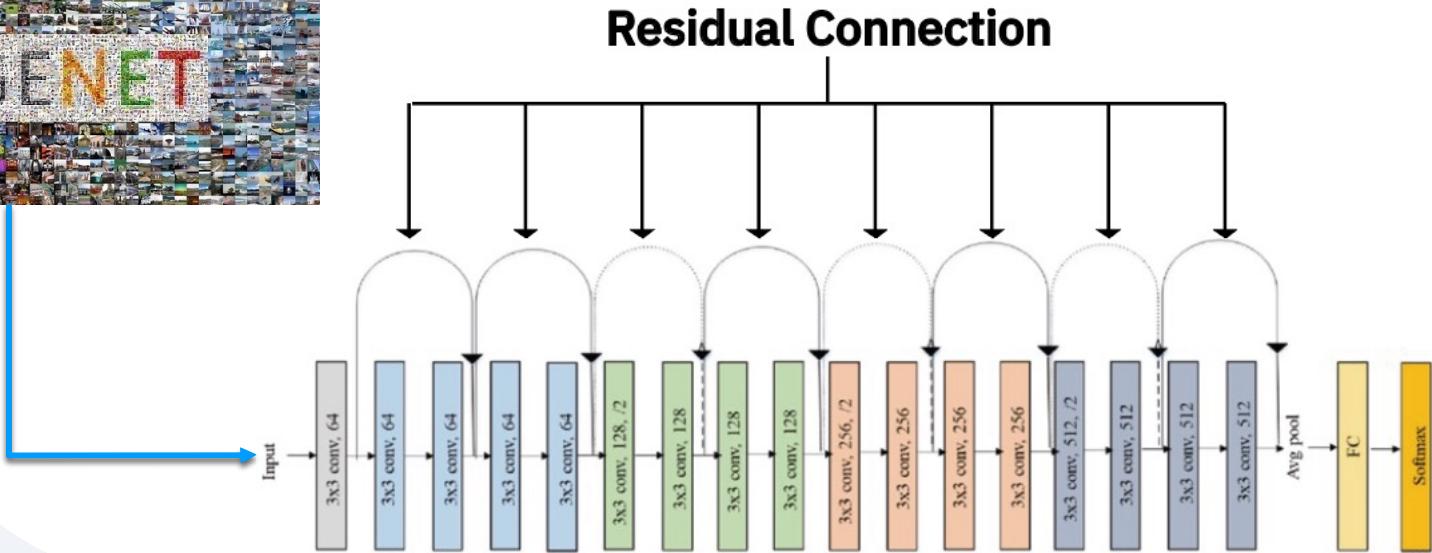
Explanation of Code File

Explanation of Data Base



1. Posture Detect Model – ResNet18

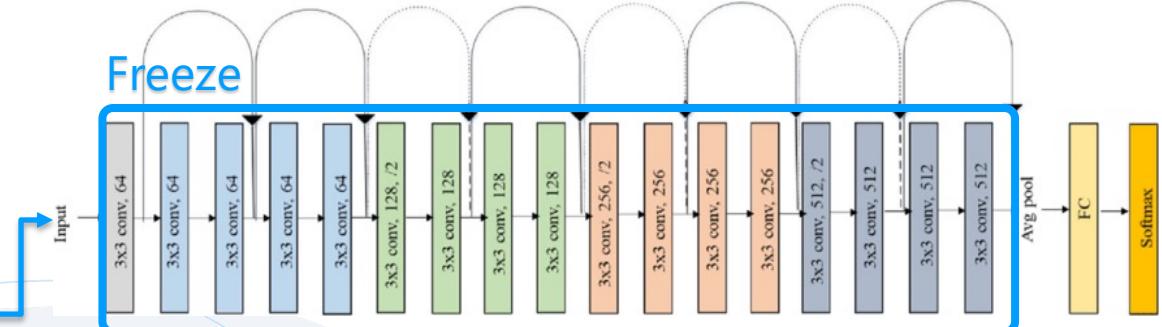
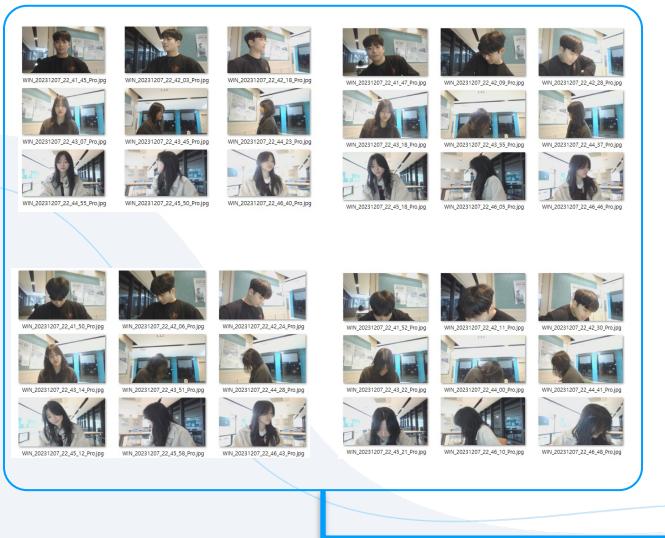
Pretained by ImageNet Dataset



1. Posture Detect Model

Finetuning with Custom Dataset (Freeze backbone)

- Made by team7, correct 250, shoulder 250, neck 250, waist 250

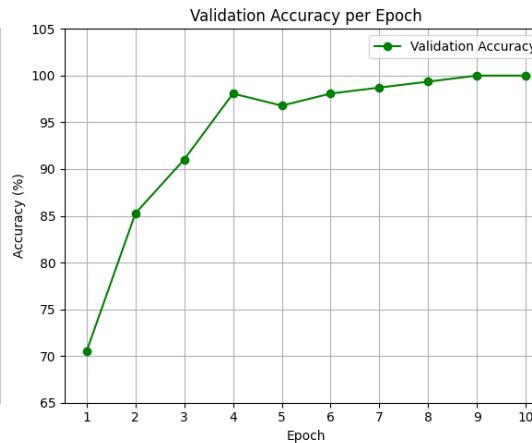
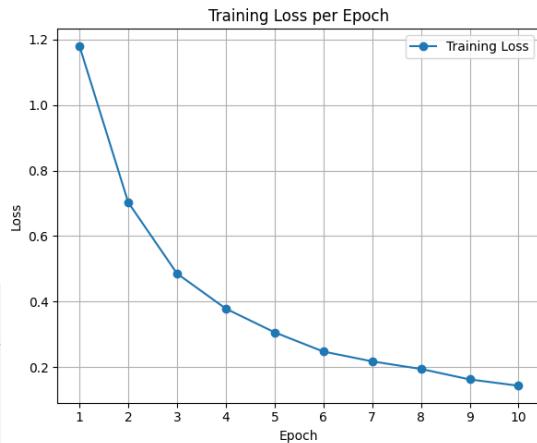


1. Posture Detect Model

Finetuning with Custom Dataset (Freeze backbone)

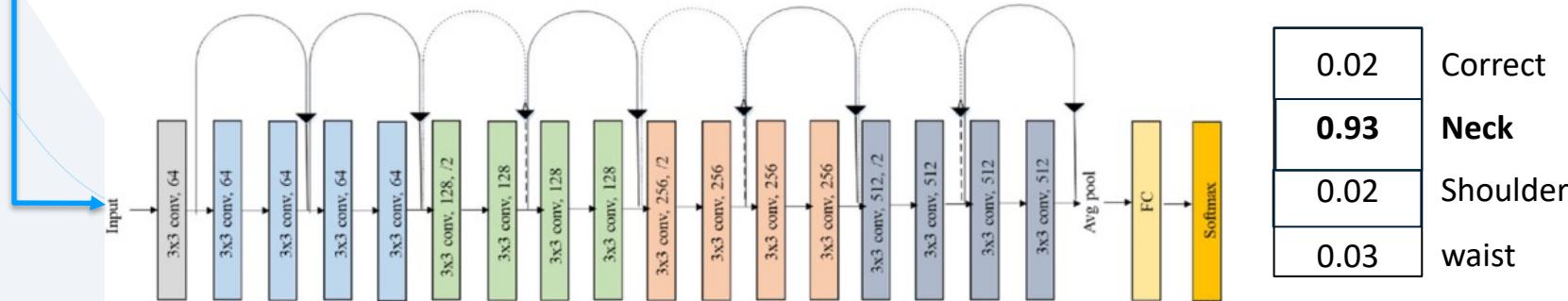
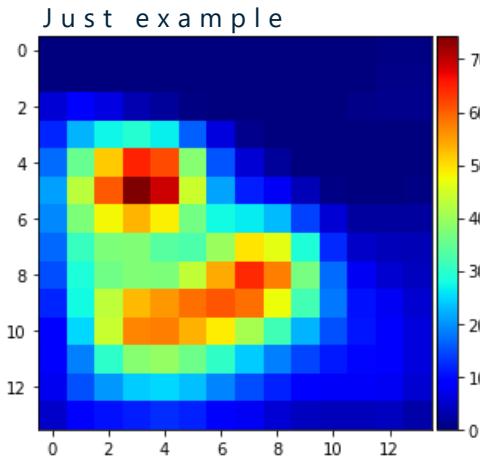
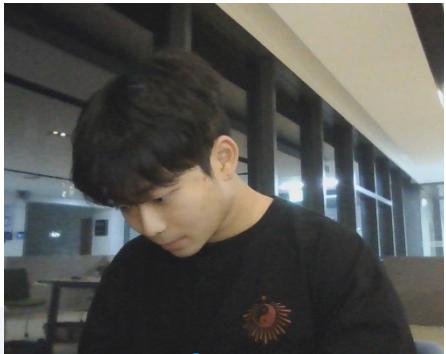
- ⇒ Initial Accuracy High -> good initialized by pretrained
- ⇒ Fast, Good Converge -> Loss decrease, Accuracy increase fast
(before 10 epochs train finish)

- ⇒ Successful transfer learning



1. Posture Detect Model - Inference

ResNet 18



2. Large Language Model

GPT-3.5 Turbo

Output gpt -3.5 LLM using langchain for medical records



| MODEL | DESCRIPTION | CONTEXT WINDOW | TRAINING DATA |
|--------------------|---|----------------|----------------|
| gpt-3.5-turbo-1106 | Updated GPT 3.5 Turbo <small>New</small> The latest GPT-3.5 Turbo model with improved instruction following, JSON mode, reproducible outputs, parallel function calling, and more. Returns a maximum of 4,096 output tokens. Learn more. | 16,385 tokens | Up to Sep 2021 |
| gpt-3.5-turbo | Currently points to gpt-3.5-turbo-0613. | 4,096 tokens | Up to Sep 2021 |
| gpt-3.5-turbo-16k | Currently points to gpt-3.5-turbo-0613. | 16,385 tokens | Up to Sep 2021 |

2. Large Language Model

- **Prompt Engineering**

- Implementing prompt engineering for effective use of existing Large Language Models (LLMs)
- Involves techniques like role assignment, example usage, and keyword incorporation

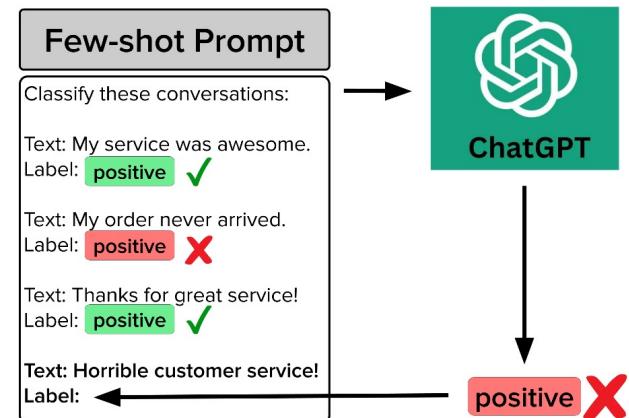
2. Large Language Model

1. Few-shot Prompting

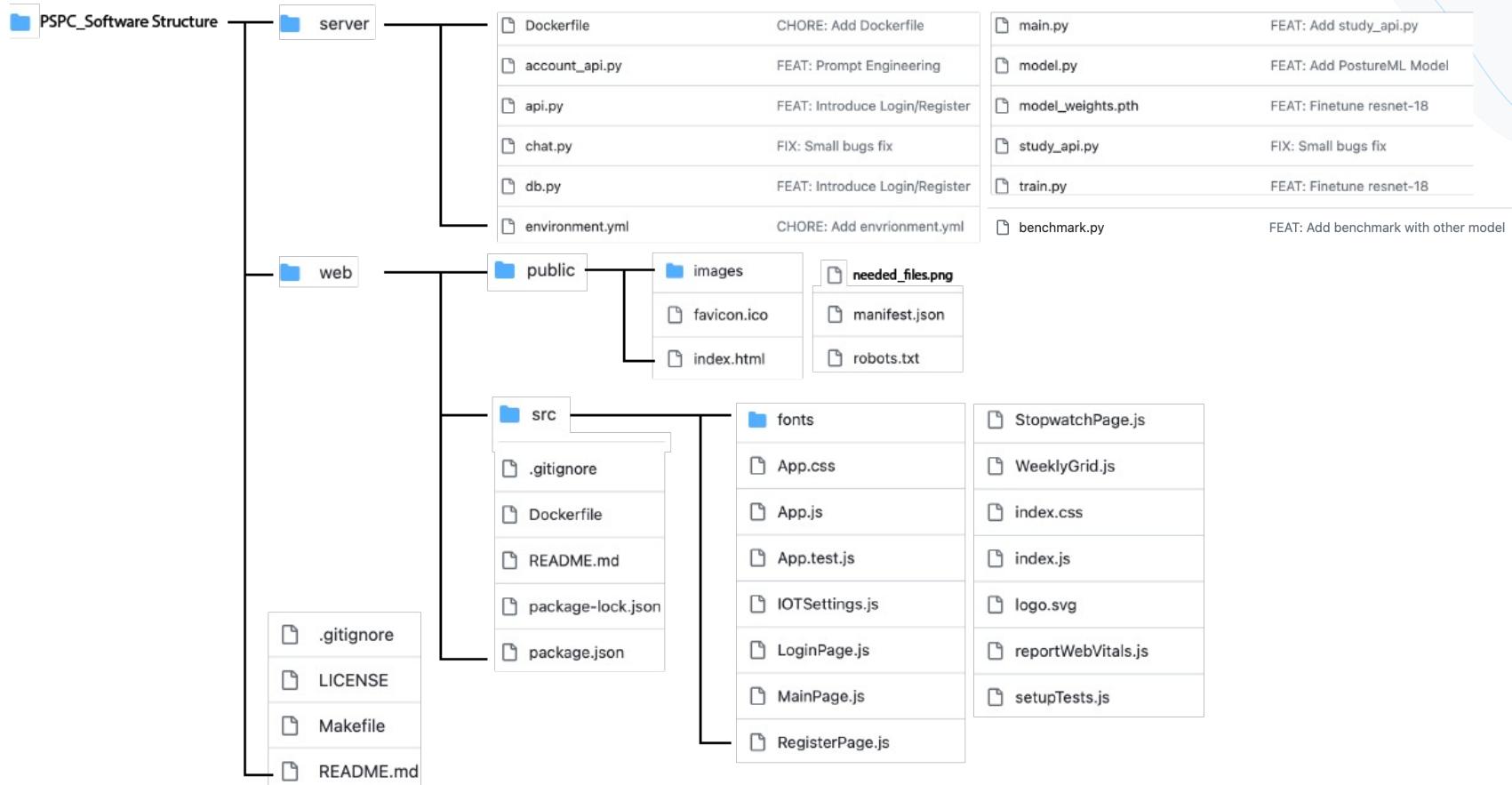
- Designed to elicit better performance in situations like diagnostics and feedback
- Aims to reduce instances of hallucination in responses

2. OpenAI Prompt Engineering Guide

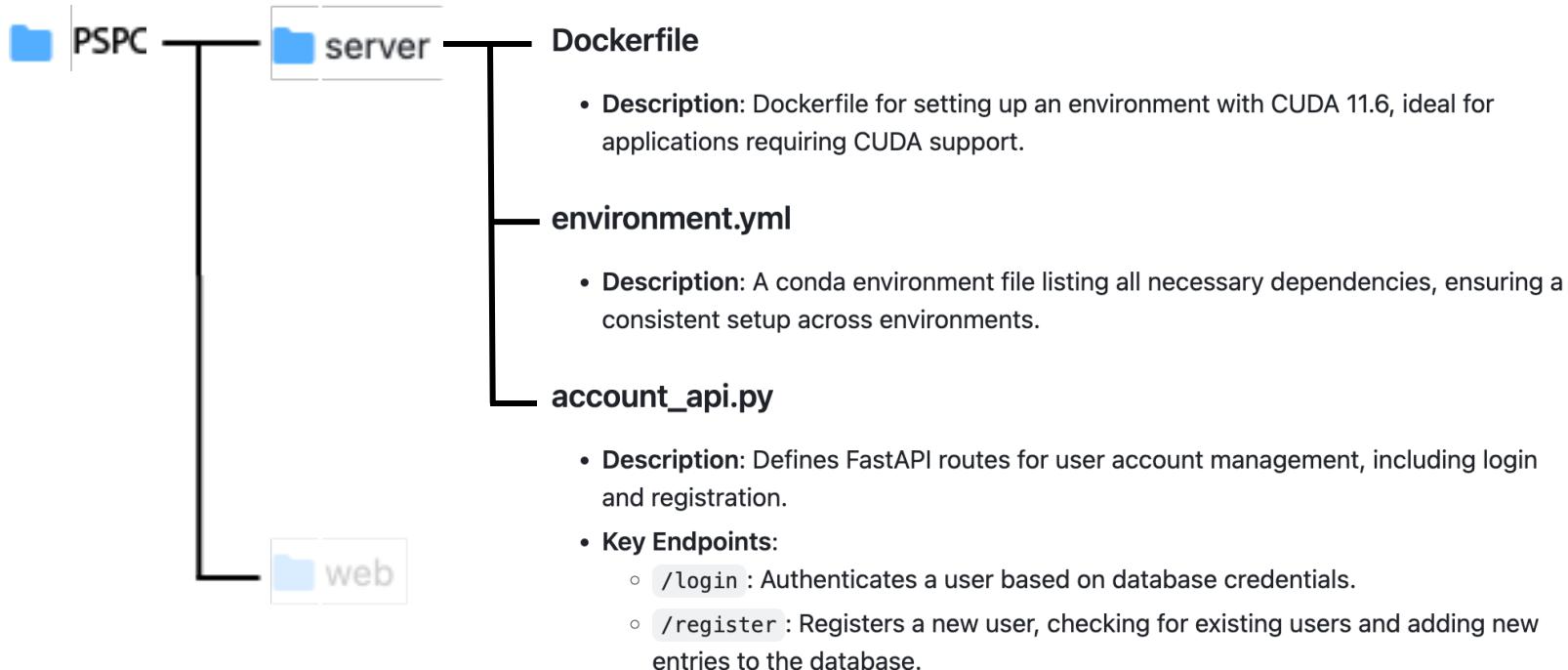
- Applying techniques to avoid directives, specificity, and inaccuracies
- Reference and implementation of guidelines from OpenAI's Prompt Engineering Guide



3. Code Structure



4. Explanation of Code File



4. Explanation of Code File



chat.py

- **Description:** Manages chat functionalities using the `langchain` library, focusing on generating responses with OpenAI's GPT models.
- **Key Function:**
 - `llm_advice` : Sets up a chat scenario for posture advice using language models.

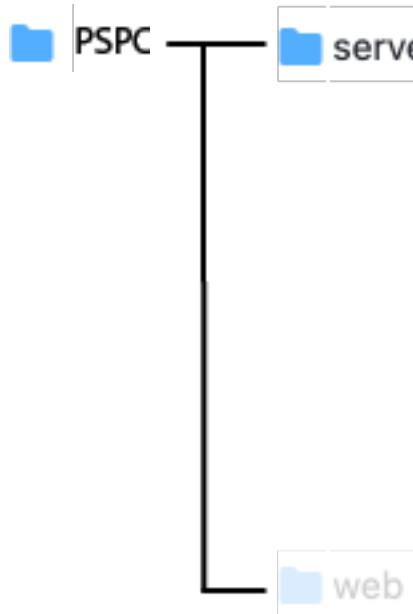
db.py

- **Description:** Manages database connectivity, using `mysql.connector` to establish connections to a MySQL database.
- **Key Function:**
 - `get_db_connection` : Creates and returns a connection to the MySQL database.

main.py

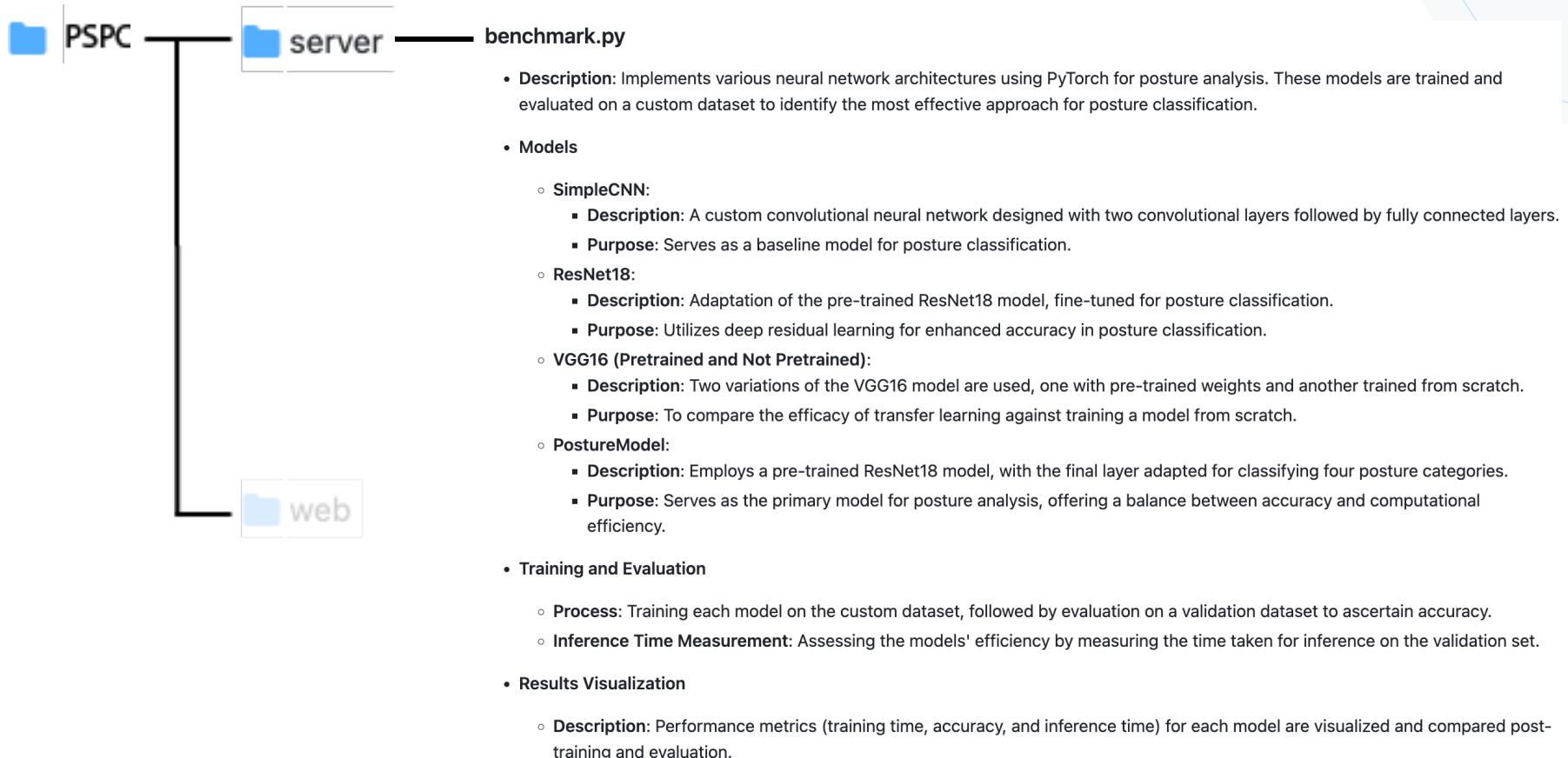
- **Description:** The entry point for the FastAPI application, setting up CORS middleware, and integrating various routers.
- **Key Features:**
 - CORS middleware for frontend interaction.
 - Integration of routers from `api`, `account_api`, and `study_api`.

5. Explanation of Code File

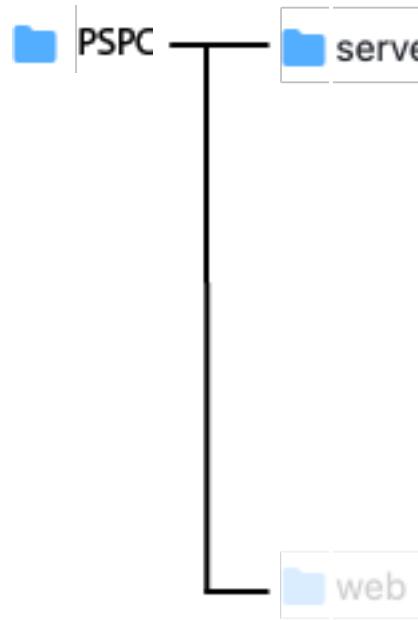


- **Description:** This file hosts the `PostureModel` class, which leverages a deep learning model for posture analysis in images. It employs the ResNet18 model, a renowned convolutional neural network, pre-trained on the ImageNet dataset. ImageNet's extensive visual database is pivotal for training deep neural networks in diverse visual recognition tasks. The ResNet18 model is chosen for its efficiency and accuracy, making it ideal for image classification tasks, such as posture analysis.
- **Key Class:** `PostureModel`
- **Usage:**
 - Instantiate `PostureModel` and utilize the `analyze_image` method with an image input for posture classification. The use of a pre-trained and fine-tuned ResNet18 model ensures robust feature extraction and efficient learning, making the model highly effective for practical posture analysis applications.

5. Explanation of Code File



5. Explanation of Code File



study_api.py

- **Description:** API router managing study-related data and functionalities, including data fetching, environmental data retrieval, posture advice, and study updates.
- **Key Endpoints:**
 - /study-data : Aggregates and retrieves study data.
 - /env : Fetches environmental data.
 - /advise : Provides posture advice.
 - /update-study : Updates study logs and interacts with pymodi based on posture analysis.

train.py

- **Description:** Script for training a CNN model for posture classification, using a pre-trained ResNet18 model and custom dataset.
- **Key Processes:**
 - Data loading and transformation.
 - Model adaptation and training.
 - Validation and evaluation.
 - Visualization of training progress.
 - Saving the trained model.

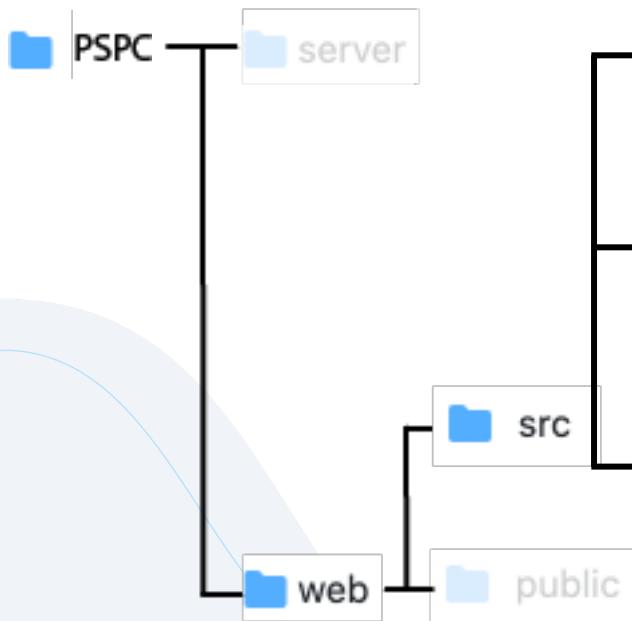
5. Explanation of Code File



Dockerfile

- **Description:** This Dockerfile sets up the environment for the React front-end application. It details the steps to containerize the React app, ensuring a consistent setup for development and deployment.

5. Explanation of Code File



src/App.js

- **Description:** The central component of the React application, `App.js` orchestrates the overall layout and navigation. It integrates key components such as `LoginPage`, `StopwatchPage`, `WeeklyGrid`, and `IOTSettings`, and manages global state and routing.

src/LoginPage.js

- **Description:** Handles the user login interface. This component sends a POST request to the `process.env.REACT_APP_API_URL}/login/` endpoint with the user's credentials. On successful login, it updates the application's state to reflect the user's logged-in status.

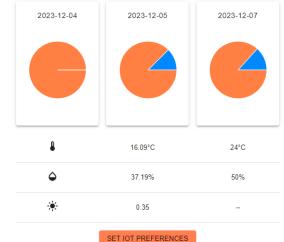
src/RegisterPage.js

- **Description:** Manages the user registration process. It captures user input for username, email, and password, and sends this data in a POST request to the `process.env.REACT_APP_API_URL}/register/` endpoint. It handles both successful registrations and errors, such as when a user already exists.

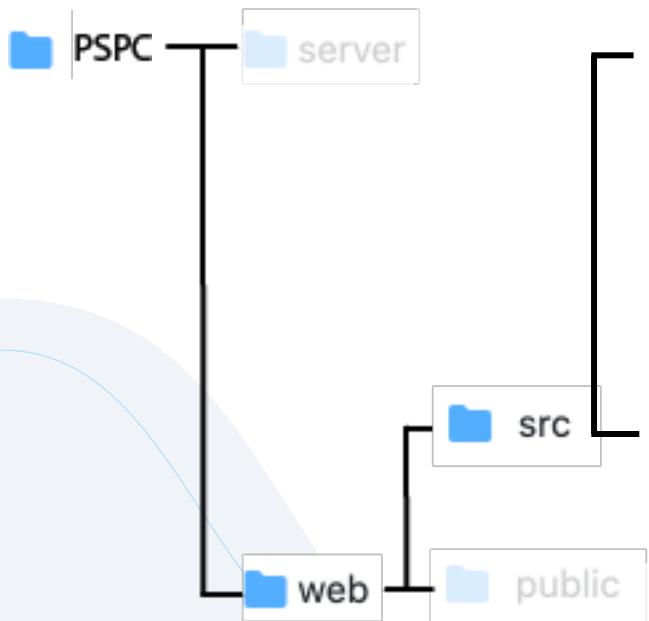
START STUDY



Based on the statistics, it appears that you may be experiencing neck and waist issues due to posture. To fix this, it is recommended to maintain a neutral spine, engage in regular stretching and strengthening exercises for the neck and waist muscles. Additionally, taking breaks and adjusting your workstation ergonomics can also help improve your posture and reduce discomfort.



5. Explanation of Code File



src/StopwatchPage.js

- **Description:** Functions as a timer for study sessions. This component periodically sends POST requests to the `process.env.REACT_APP_API_URL}/update-study` endpoint every 60 seconds to update the server with ongoing session data. It displays warnings based on the server's response, particularly related to the user's posture.

src/WeeklyGrid.js

- **Description:** Provides a visual representation of the user's weekly study data. This component uses the `studyData` prop to display a pie chart for each day's study sessions, showing the breakdown of correct and incorrect posture times. It features an interactive modal for detailed daily analysis, enhancing user engagement.

5. Explanation of Data Base ; MySQL

1. Table: `log`

- **Description:** Stores logs of users' posture status.
- **Fields:**
 - `id` : Integer (Primary Key, Auto-incremented)
 - `date` : Datetime
 - `status` : Varchar(255)
- **Example Entry:**

| <code>id</code> | <code>date</code> | <code>status</code> |
|-----------------|---------------------|---------------------|
| 1 | 2023-12-04 20:49:40 | neck |
| 2 | 2023-12-04 20:50:32 | shoulder |
| 3 | 2023-12-04 20:51:14 | correct |

- **Purpose:** Enables tracking of user posture over time for feedback and analysis.

5. Explanation of Data Base ; MySQL

2. Table: `users`

- **Description:** Maintains user account information.
- **Fields:**
 - `id` : Varchar(255) (Primary Key)
 - `passwd` : Varchar(255)
 - `email` : Varchar(255)
- **Example Entry:**

| <code>id</code> | <code>passwd</code> | <code>email</code> |
|-----------------|---------------------|----------------------|
| johndoe | 1234 | john.doe@example.com |
| janedoe | 1234 | jane.doe@example.com |
| user123 | 1234 | user123@example.com |

- **Purpose:** Essential for user management, handling authentication, and storing contact information.

Development Process : Hardware and Mobility

Used MODI Modules

Model Design

Used Functions

Web Camera Usage

3D printing Structure



1. Used MODI Modules



Display Module

Used for
Displaying information

Speaker Module

Used to provide
Voice information

Environment Module

Used for
Environmental
recognition

Network Module

Used for
Uploading code

2. Model Design



3. Used Functions



**Display
Module**

Displays the message to the user if he/she is studying in the wrong posture



**Environment
Module**

Checks the environment(temperature, humidity, brightness) and sends the current value to the server



**Speaker
Module**

Output a warning sound for 3 seconds if the user is not studying in the right posture

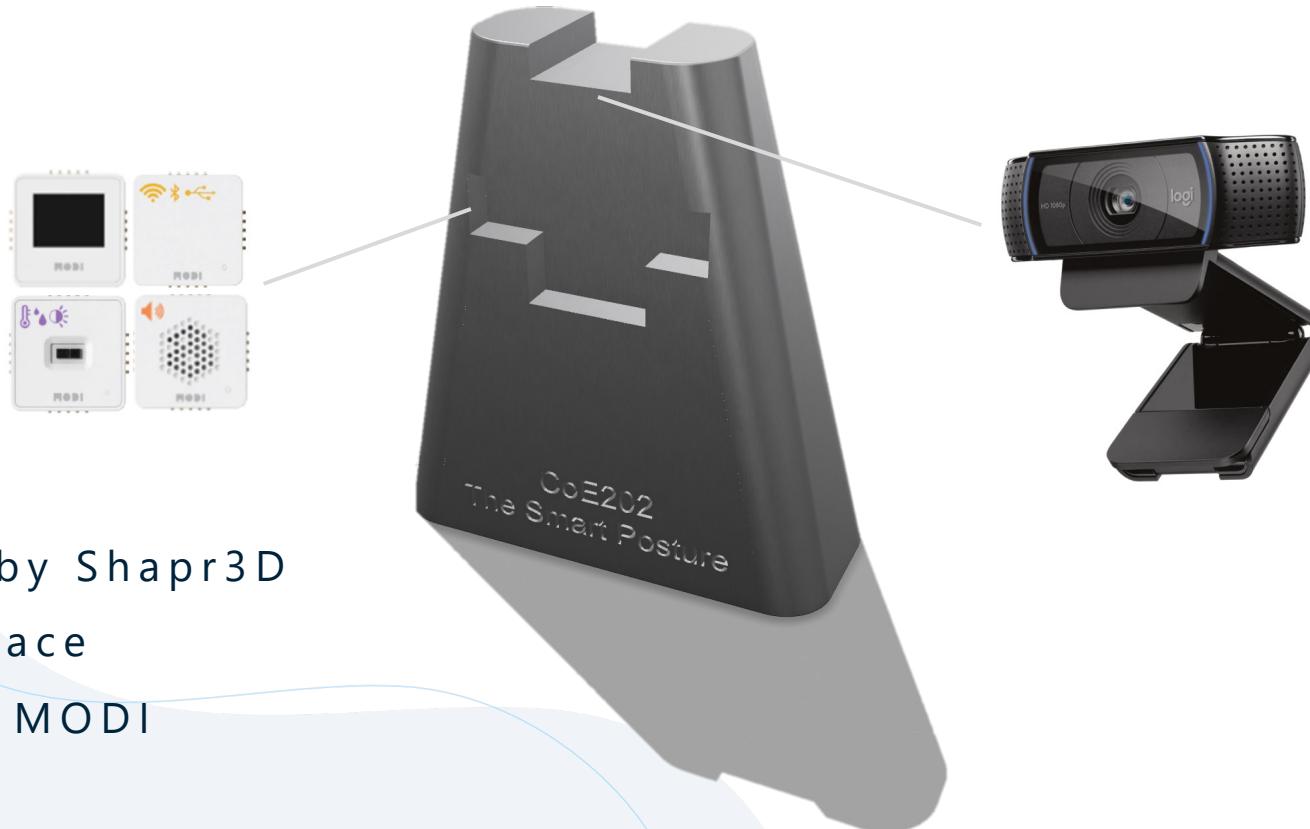
4. Web Camera

- Logitec C920 HD PRO webcam
- Used for real-time recording
- Applicable to
every internet – connected device



C920 HD PRO WEBCAM

5. 3D printing Structure



- Designed by Shapr3D
- Used to place
webcam & MODI

DEMO Video

Hardware Settings

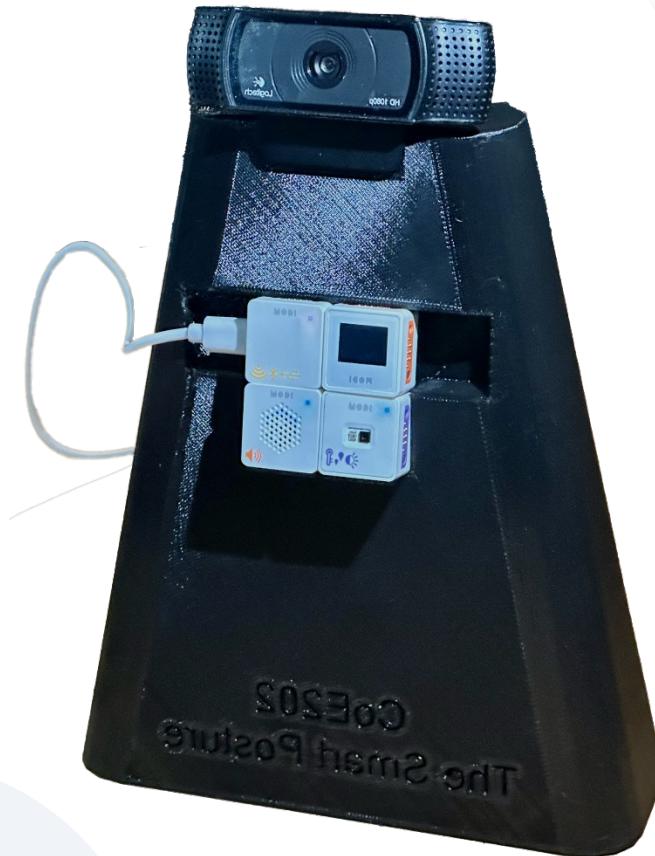
Demo Video

Web App

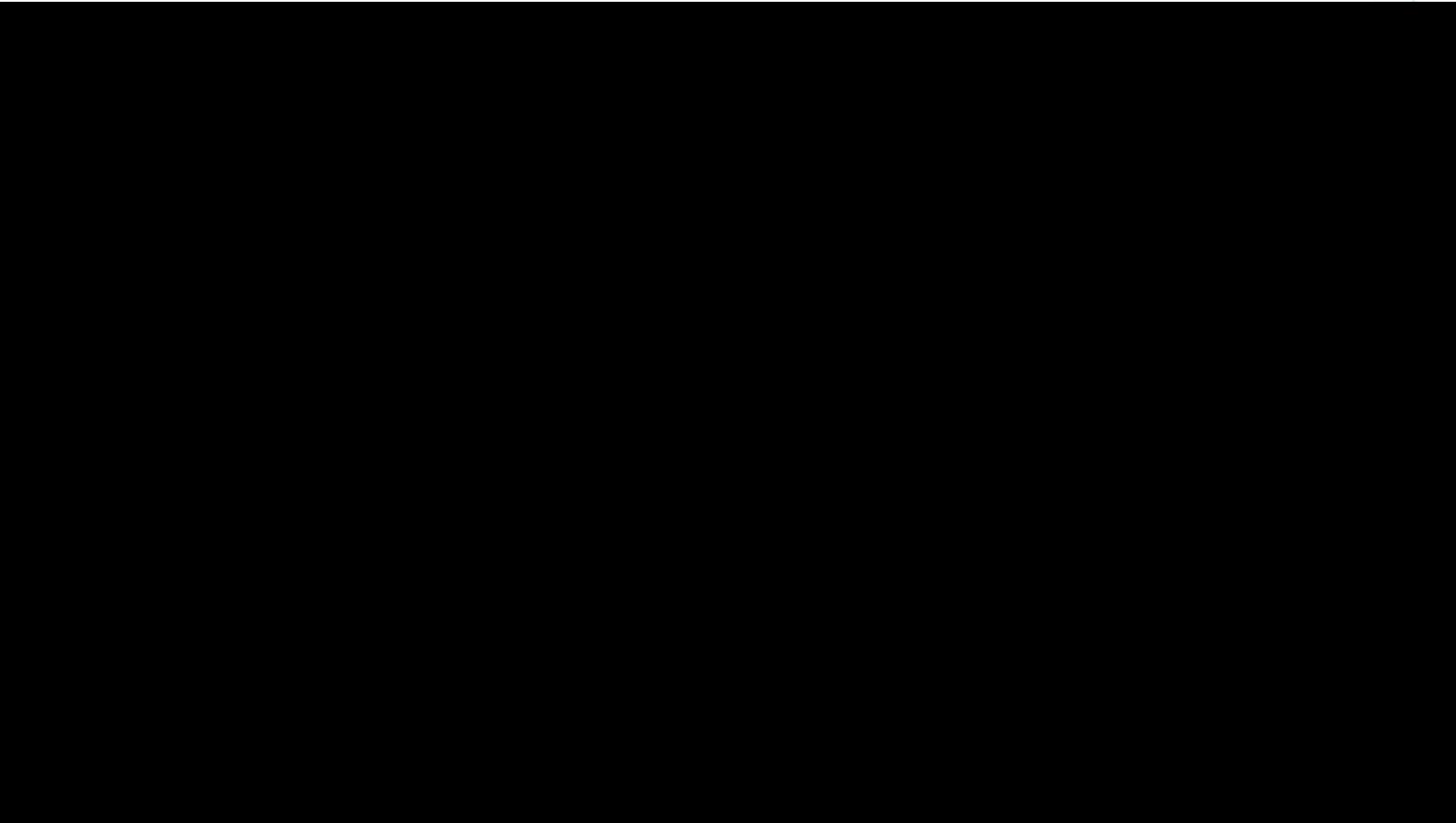


1. Hardware Settings

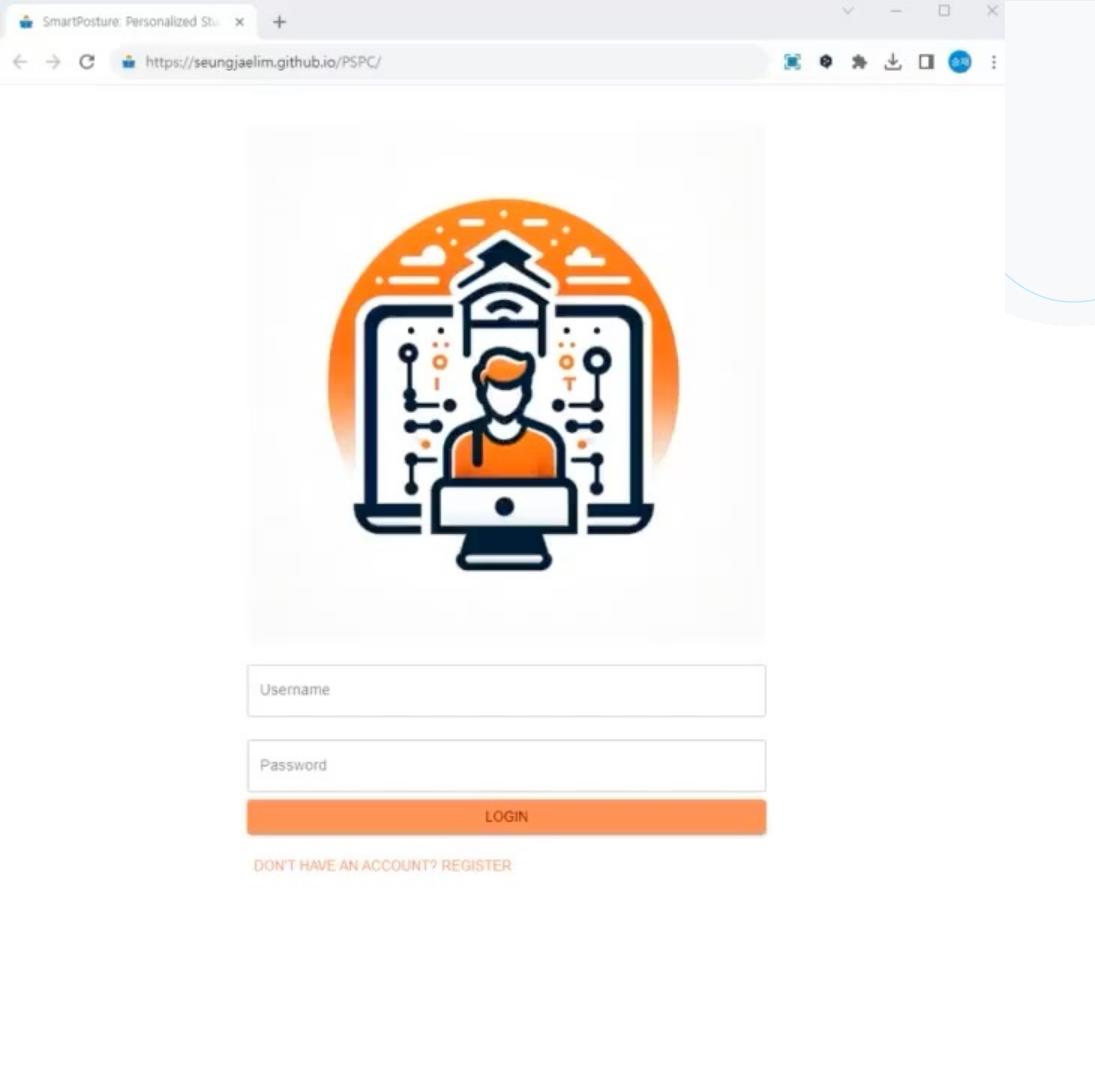
- 3D printed structure to handle web camera and MODI
- web camera and MODI connected to laptop



2. Demo Video: How to use



2. Demo Video : Web application



2. Demo Video: Terminal

The screenshot shows a terminal window within a code editor interface. The terminal tab is active at the top, with other tabs like PROBLEMS, OUTPUT, DEBUG CONSOLE, PORTS, and POSTMAN CONSOLE visible. The terminal itself has a dark background and displays the following text:

```
(base) PS D:\PSPC\server> co
```

The cursor is positioned at the end of the command 'co'. In the bottom right corner of the terminal window, there is a small preview pane showing two open files: 'node web' and 'powershell server'. The preview pane has a light blue header bar with icons for zooming and closing.

At the very bottom of the screen, the status bar displays the following information:

main* ↻ ⌂ 0 ▲ 10 ⌂ 0
Ln 24, Col 28 Spaces: 4 UTF-8 CRLF { Python 3.10.10 64-bit ⌂ Go Live ⌂

3. Web app



Initial page



Register

00:00:09

STOP

Warning: Please correct your posture!

Detecting
Bad posture

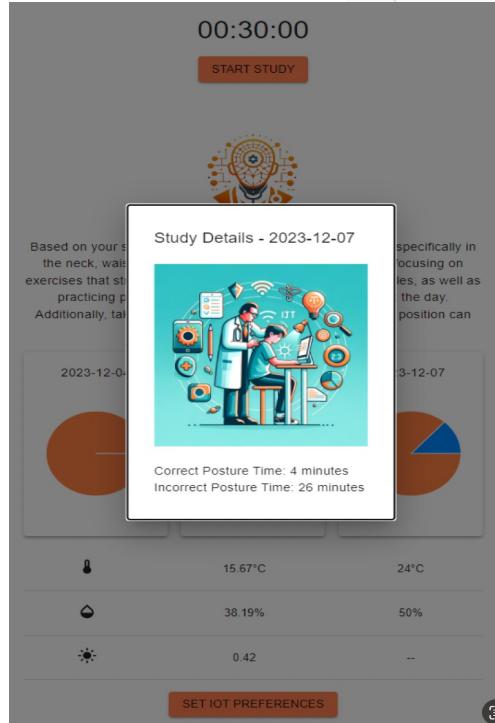


Log in

00:00:23

STOP

Recording
Study



Premeasured
record

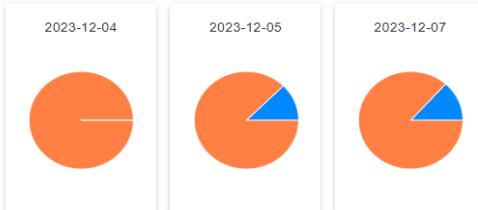
3. Web app

00:30:00

START STUDY



Based on the statistics, it appears that you may be experiencing neck and waist issues due to poor posture. To correct this, it is recommended to maintain a neutral spine alignment and practice regular stretching and strengthening exercises for the neck and waist muscles. Additionally, taking breaks and adjusting your workstation ergonomics can also help improve your posture and reduce discomfort.

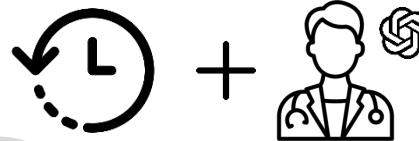


| | |
|-----------------|---------|
| Temperature | 16.09°C |
| Humidity | 37.19% |
| Light intensity | 0.35 |

SET IOT PREFERENCES

Premeasured record

Customized feedback based on user's study posture record by doctor GPT



+

Current
Temperature
Humidity
Light intensity

compared to

Preferred
Temperature
Humidity
Light intensity

Setting optimal study environment using IoT

Discussion

Evaluation

Advantages & Expectation

Improvement points

Conclusion



1. Evaluation – Baseline

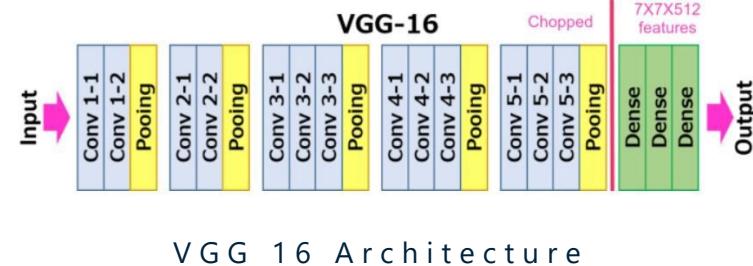
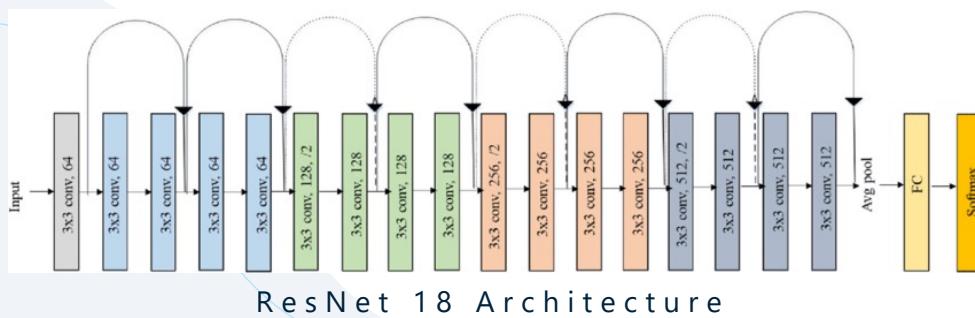
Our model: PostureModel (pretrained ResNet18 + custom architecture)

Baseline 1: Simple CNN

Baseline 2: ResNet18 (Not Pretrained)

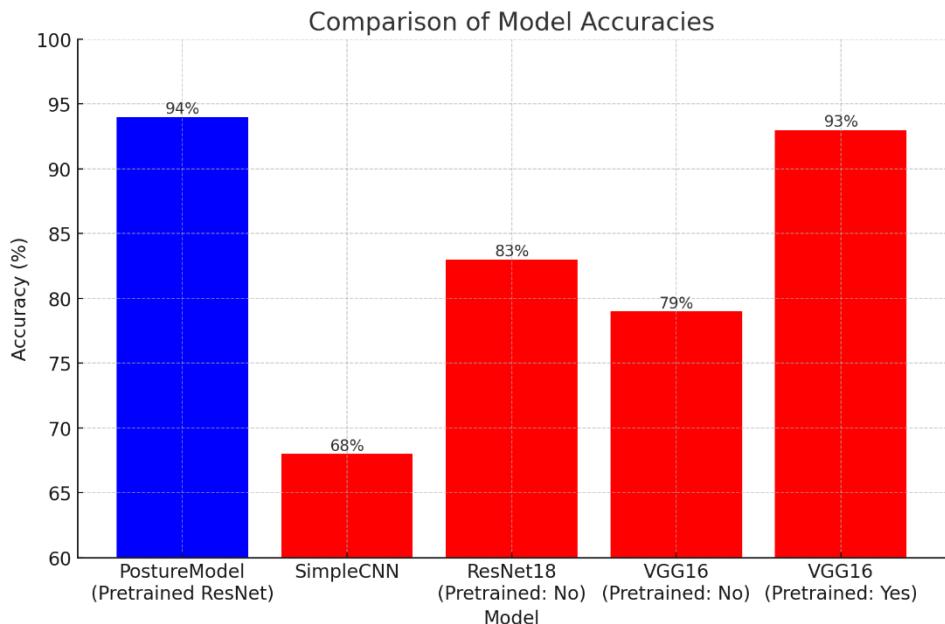
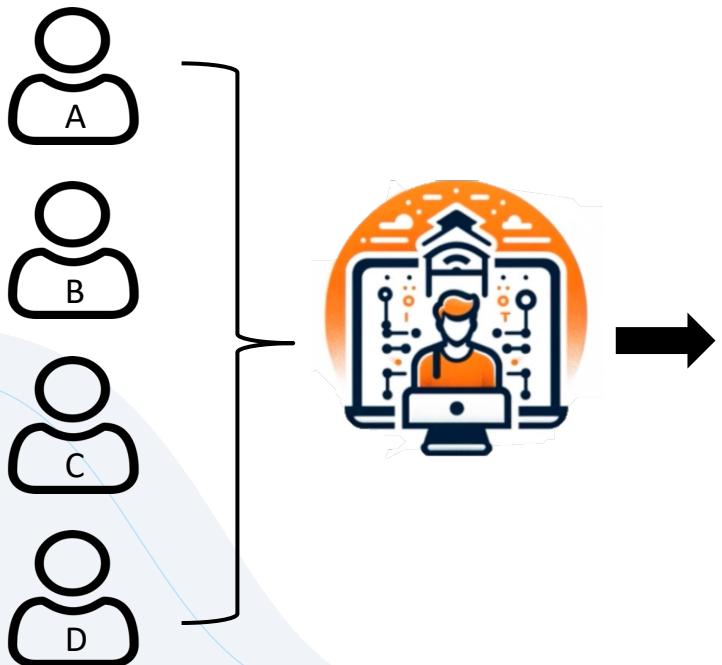
Baseline 3: VGG16 (Not Pretrained)

Baseline 4: VGG16 (Pretrained)



Evaluation Metric: Accuracy, Training time, Inference Time

1. Evaluation - Accuracy



25 times per person, total 100 times of comparison

1. Evaluation - Accuracy

- **Pretraining Efficiency**
 - When trained for 10 epochs, using a pretrained model shows better performance
 - Rapid convergence and reaching global optima
- **Stacking Architecture**
 - SimpleCNN shows excellent performance in validation
 - Low test accuracy → overfitting
 - Expected reason : shallow model only extract lower features, leading to bias
- **Residual Connection**
 - (Compared with VGG) Addition of residual connections in the model prevents gradient vanishing
→ effective learning
 - ResNet18 : outperforms in scenarios without pretraining

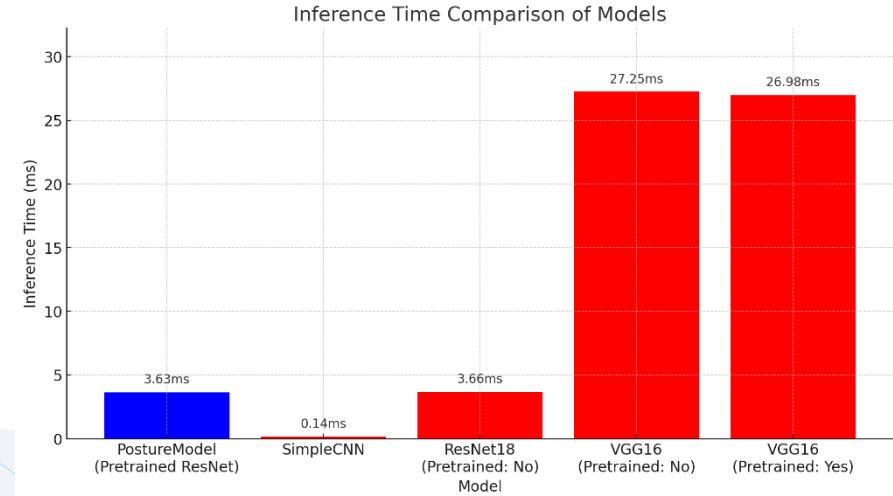
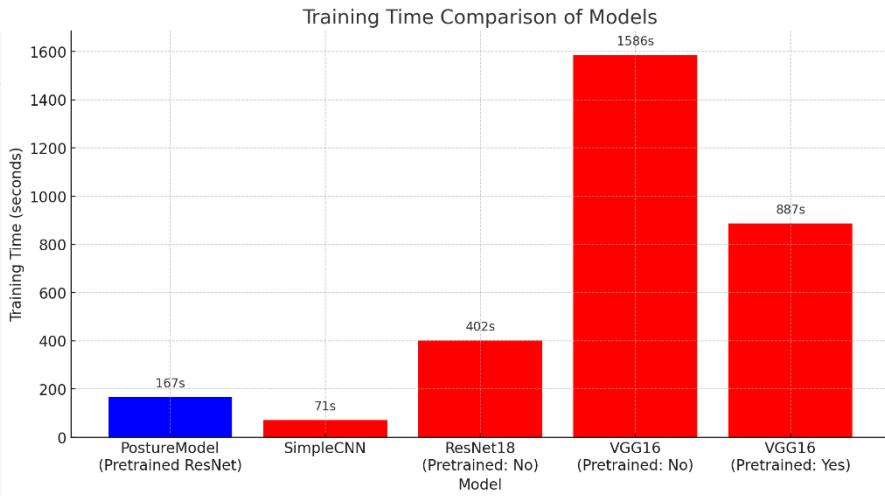
1. Evaluation - Throughput

- **SimpleCNN**

- Offers very fast inference speed
- Lacks expressive power (shallow model architecture)

- **ResNet vs VGG**

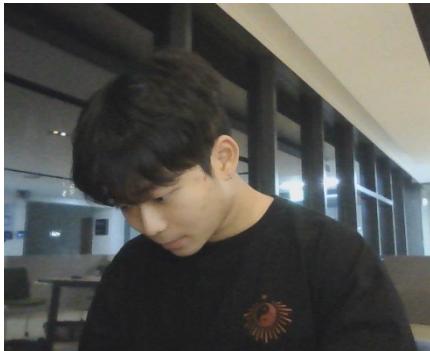
- Accuracy : ResNet \approx VGG
- Inference Time (faster) : ResNet \gg VGG \rightarrow **ResNet18**



1. Evaluation – Inference Example



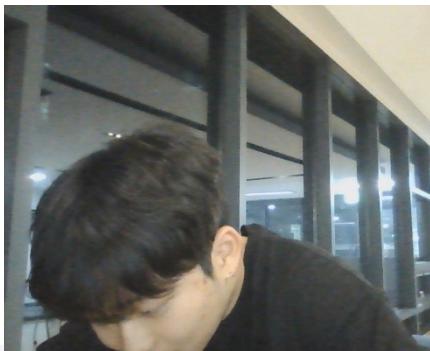
| | |
|-------------|----------|
| 0.91 | Correct |
| 0.06 | Neck |
| 0.02 | Shoulder |
| 0.01 | waist |



| | |
|-------------|----------|
| 0.02 | Correct |
| 0.93 | Neck |
| 0.02 | Shoulder |
| 0.03 | waist |



| | |
|-------------|----------|
| 0.00 | Correct |
| 0.03 | Neck |
| 0.96 | Shoulder |
| 0.01 | waist |



| | |
|-------------|----------|
| 0.00 | Correct |
| 0.02 | Neck |
| 0.01 | Shoulder |
| 0.97 | waist |

2. Advantages & Expectation

Advantages



Posture Correction User Awareness

Provide real-time feedback to users



Customized Feedback

Offer personalized tips & recommendations based on the user's specific posture habits



Study Time Tracking

Tracked study sessions and categorizes data. Study habits over time and identify patterns visualized on website.



Portability

Used anywhere by mobile website

2. Advantages & Expectation

Expectation



Health Improvement

Aims to reduce the risk
of musculoskeletal issues

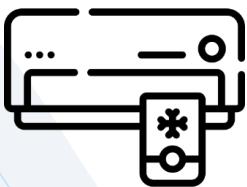
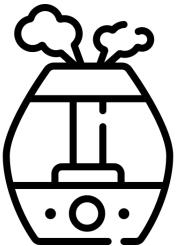


Improved Concentration & Learning Efficiency

Higher levels of concentration
Enhances learning experience

; High demanded for people who study for a long time

3. Improvement Points



Possible Improvement

IoT

Connection with home appliances such as air conditioner, humidifier..etc

Set suitable environment for study using collected data by MODI

Application

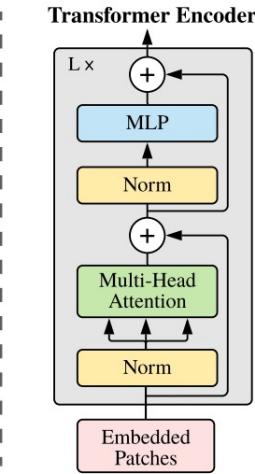
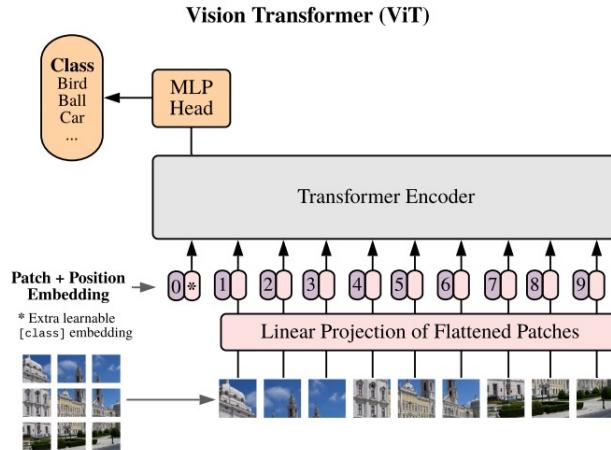
Applicable to activities that require correct posture (game, reading books.. Etc)

3. Improvement Points

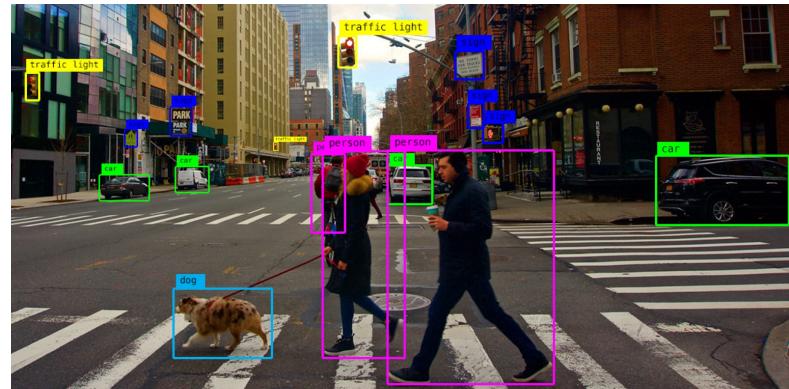
Possible Enhancement

Improved Machine Learning Models

Increase Accuracy
ViT



Single device-Multi User Service
YOLOv5



3. Improvement Points

Possible Improvement

Improved Data Forms

Potential for Multimodal Integration

Incorporation of various data types
beyond just vision data

Enhance the model's robustness and adaptability

Ensemble Techniques

Considering the implementation of ensemble methods
Combines multiple models for improved performance and reliability

4. Conclusion

SmartPosture ; Personal Study Posture Corrector(PSPC)

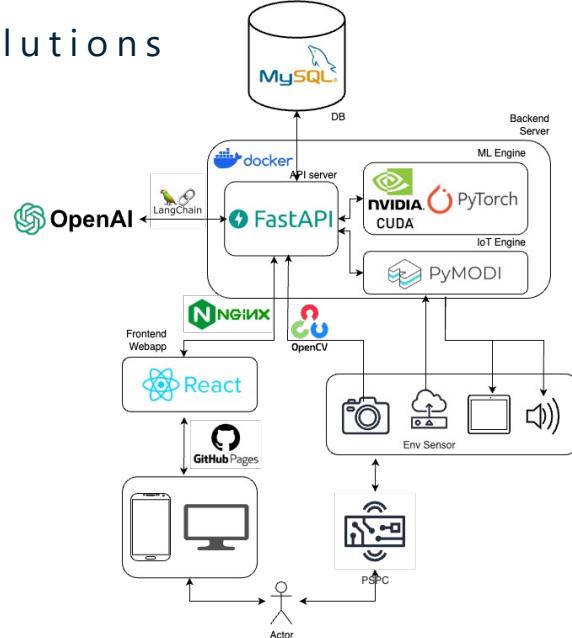


KEY FEATURES

1. Observe Study Posture
2. Provide AI Posture Habits & Solutions
3. IOT System

OVERALL ARCHITECTURES

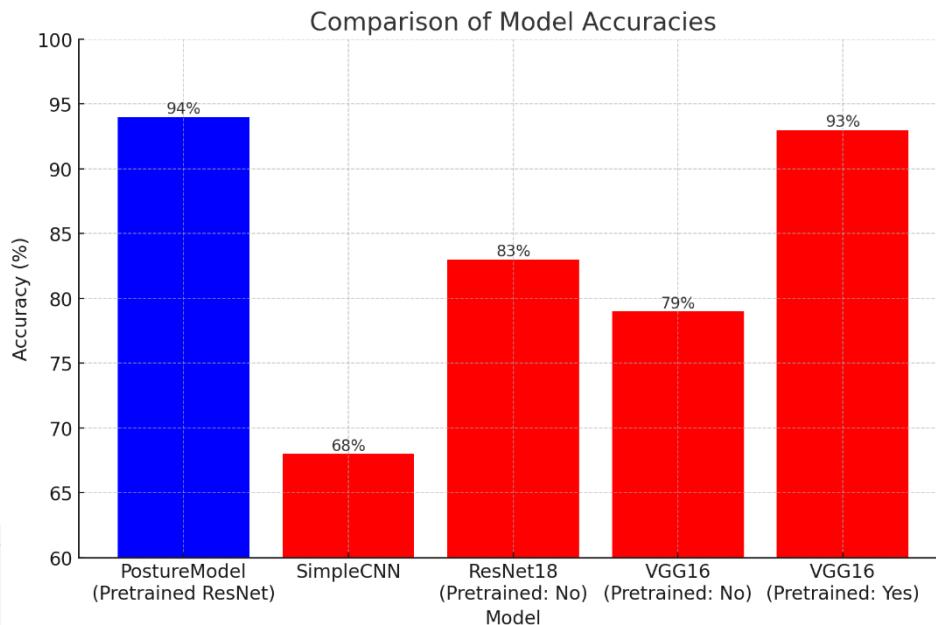
1. Deploy ; NGINX, Github Pages
2. WebApp ; React
3. API Server ; Docker, FastAPI, MySQL
4. Posture Vision Model ; Pytorch, CUDA, OpenCV, DoctorGPT
5. Doctor AI ; Langchain, openai api
6. IOT; PyMODI



4. Conclusion

SmartPosture ; Personal Study Posture Corrector(PSPC)

MODEL ACCURACIES



4. Conclusion

SmartPosture ; Personal Study Posture Corrector(PSPC)

MODEL SELECTION

SimpleCNN | ResNet | VGG

| | | | |
|----------|------|--------|------|
| Speed | Fast | Normal | Slow |
| Accuracy | Low | High | High |

Accuracy and inference time both compliant

ResNet

4. Conclusion

SmartPosture ; Personal Study Posture Corrector(PSPC)

ADVANTAGES & EXPECTATION

**Posture Correction
User Awareness** + **Customized
Feedback** + **Study Time
Tracking** + **Portability**



**Health
Improvement** & **Improved
Learning Efficiency**

Compatibility and Integration

Various devices and operating systems
Integration with other productivity tools or
platforms

4. Conclusion

Full Code Provided :

<https://github.com/SeungjaeLim/PSPC>