




RehabApp – AI-Powered Physiotherapy Coach

RehabApp is an end-to-end research prototype that:

-  **Trains** a lightweight multi-head CNN + Bi-LSTM (PoseQualityNet-KP) on the **REHAB24-6** physiotherapy corpus.
-  **Evaluates** its accuracy, ablation variants, and inference speed.
-  **Serves** real-time, joint-level feedback through a **FastAPI** back-end and a **React** front-end.

Repository Layout

Path	What's inside
Data/	Contains links and information about the data sources used.
└ Data Source Links.txt	Text file with links/references to the original and processed datasets.
dataprep_model-training_evaluation_inference-scripts/	Data-prep helpers and the main Jupyter notebook
└ model_training_evaluation_and_inference.ipynb	 Data augmentation, model creation, evaluation, ablation study & inference
└ Data-REHAB24-6_New/	Augmented window tensors & meta-data generated by the notebook
Documentation/	Project documentation, reports, and guides.
└ Presentation/	Slides (PDF) and recorded project presentations.
└ Project Report/	Detailed project report (PDF and Latex code)
└ User Guide/	Instructions and guides for using Repo and RehabApp.
rehab-app-backend/	FastAPI service (main.py, model loader, SQLite storage, Dockerfile)
rehab-app-frontend/	React client (pose capture, WebSocket, dashboard UI)
docker-compose.yml	One-shot build & launch for both apps
.gitignore	Ignores editor caches (.vscode/, .qodo/, etc.)

⚡ Running the Application:

1. Clone the repository:

- git clone <https://github.com/IS06-Jithin/RehabApp.git>

2. Command prompt/terminal navigate to RehabApp folder

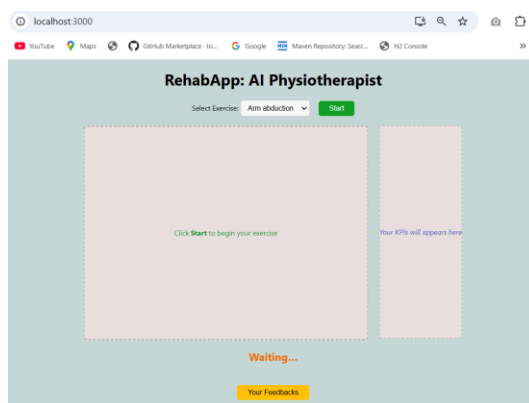
- cd RehabApp

3. Build and run with Docker Compose:

- docker-compose up --build

4. Access the application: Once the build is complete and containers are running:

- **React Client (Frontend):** <http://localhost:3000/> (allow webcam access when prompted)



5. Using the App

- Select the exercise from the dropdown menu and click the **start** button
 - Perform the **selected Exercise**
 - Pay attention to the **visual/Auditory feedback** and correct yourself
 - Upon completion, click the **Stop** button and review your session summary
 - If, have any feedback on the Apps usability click the feedback button and provide it.
-

Prerequisites





- **Docker & Docker Compose:**
 - **Windows / macOS:** Install Docker Desktop from docker.com.
 - **Linux:** Install the Docker Engine and the docker-compose plugin from your distribution's package manager or follow the [official guide](#).
- **Git:** Download and install from git-scm.com.
- **(Optional) NVIDIA Driver + CUDA:**
 - Needed only if you want to benchmark the model on a GPU. CPU inference works out of the box.

Model at a Glance

Architecture

Block	Details
Frame Encoder	1-D CNN (99 → 128 → 512 channels)
Temporal Encoder	2-layer Bi-LSTM (256 hidden units/direction)
Exercise Embedding	6 → 64 MLP
Prediction Heads	Quality (2-way) • 14-Joint Regression • Exercise ID (6-way)
Total Footprint	3.41 M parameters (≈ 13 MB)

Performance Highlights

-  **91.5% F1-score** on repetition quality assessment.
 -  **99.5% F1-score** on exercise identification.
 -  **4.73° Mean Absolute Error** for joint-angle regression.
 -  **~30 FPS** on a mid-range CPU (or ~7,500 windows/second on GPU).
-



License

This project is released under the **MIT License**. The dataset media (REHAB24-6) remain under their original license. Please see the [REHAB24-6 Zenodo record](#) for details.



Citation

If you use RehabApp or its components in your research, please cite it as:

@misc{rehabapp2025,

author = {Jithin Krishnan},

title = {RehabApp: AI-Powered Real-Time Physiotherapy Coach},

year = {2025},

howpublished = {\url{https://github.com/IS06-Jithin/RehabApp}}