



Ollscoil
Teicneolaíochta
an Atlantaigh

Atlantic
Technological
University

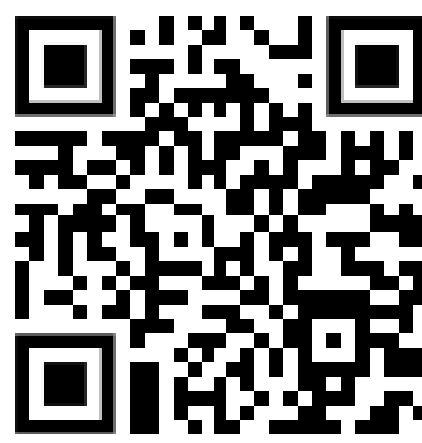
DEP-Fitness

Eoin Lawless – G00396053

Pose Detection using CNNs

B.Sc. (Hons) in Software Development – Final Year Project

GitHub Link:



LinkedIn:



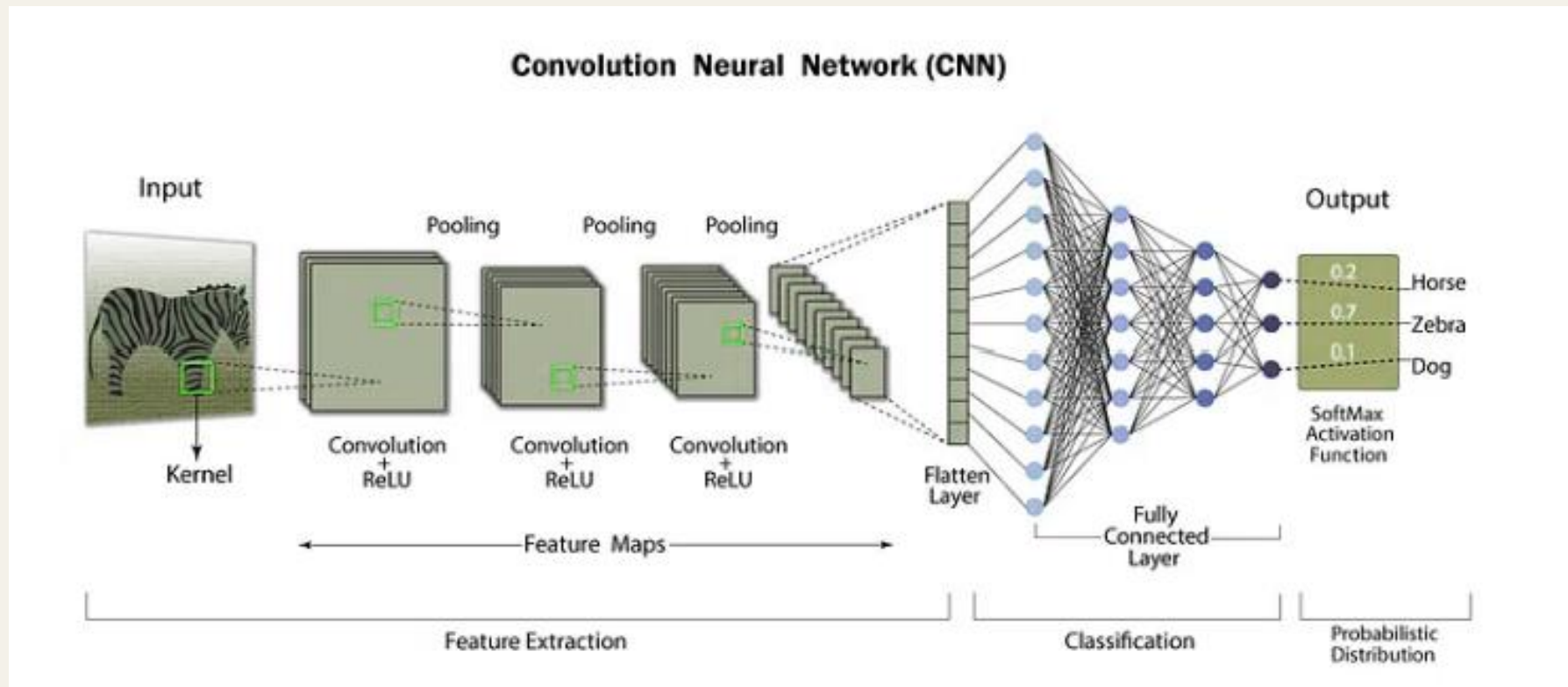
Introduction

This dissertation outlines the development of a fitness application designed to enhance exercise routines through real-time pose detection and feedback. Central to the project is a Convolutional Neural Network (CNN) model, complemented by agile development practices and cybersecurity. The application, developed with React, Firebase, and MediaPipe, offers live feedback to improve workout performance and safety. The project was inspired by advancements in fitness technology and a desire to integrate artificial intelligence into exercise management. The main objectives are:

- Developing a Pose Detection Model:** Building an efficient CNN to detect and analyze user movements in real time for correct exercise execution.
- Building a Functional Application:** Creating a React-based application that processes video for instant feedback, tracks user progress, and manages results.
- Applying Cybersecurity and Agile Practices:** Ensuring user data protection with cybersecurity and using agile methodologies for flexible development.

CNN's

One of the main focuses of this project investigates the uses of CNN's abilities on pose detection



Convolutional Neural Networks (CNNs) are structured like human visual perception:

- 1.**Convolutional Layers:** These layers use filters to identify patterns in the input. Each filter scans a small region of the input, producing a map of where certain features are located.
- 2.**Activation Functions:** After convolution, activation functions like ReLU are applied to introduce non-linearities, aiding in learning complex patterns.
- 3.**Pooling Layers:** They down-sample the input, reducing its size while retaining important features. Common types include max pooling and average pooling.
- 4.**Fully Connected Layers:** Typically found at the end of the network, these layers connect every neuron to the next layer, enabling the network to classify inputs based on learned features.

Pose detection models

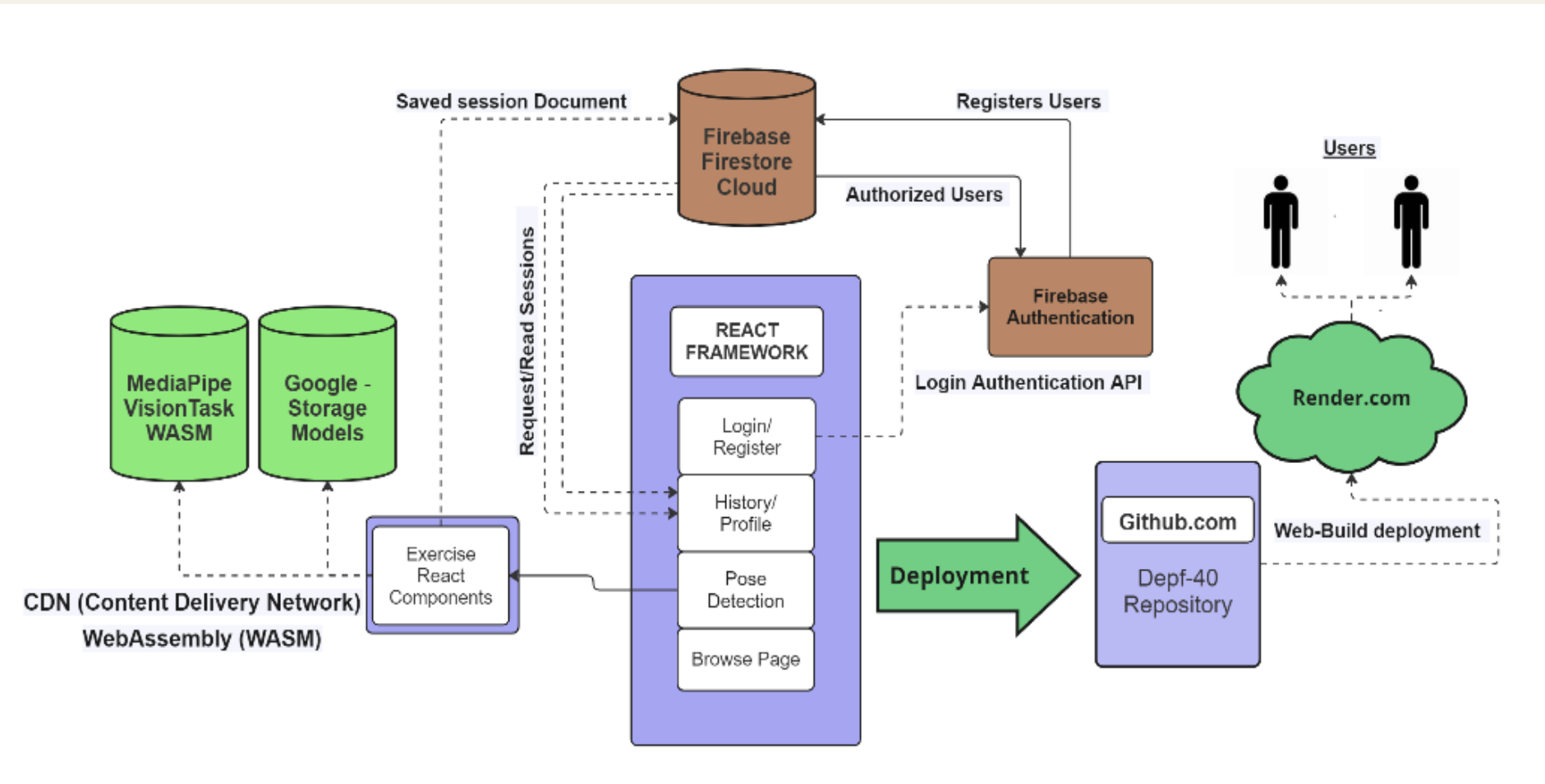
What is pose detection:

Pose detection is a technique where Convolutional Neural Networks (CNNs) identify human body positions in images or videos by detecting specific points known as keypoints, which represent major joints like elbows and shoulders. For effective keypoint detection, CNNs often incorporate modified architectures such as Feature Pyramid Networks, which help identify keypoints at multiple scales.

How is it made:

Training CNNs for pose detection requires a comprehensive dataset with images annotated accurately to show these keypoints. Annotation is usually done manually, where human annotators mark the keypoints on images, or semi-automatically using existing models. Accurate annotations are crucial as they directly affect the learning and accuracy of the pose detection models.

Architecture



Front-end components and user interfaces

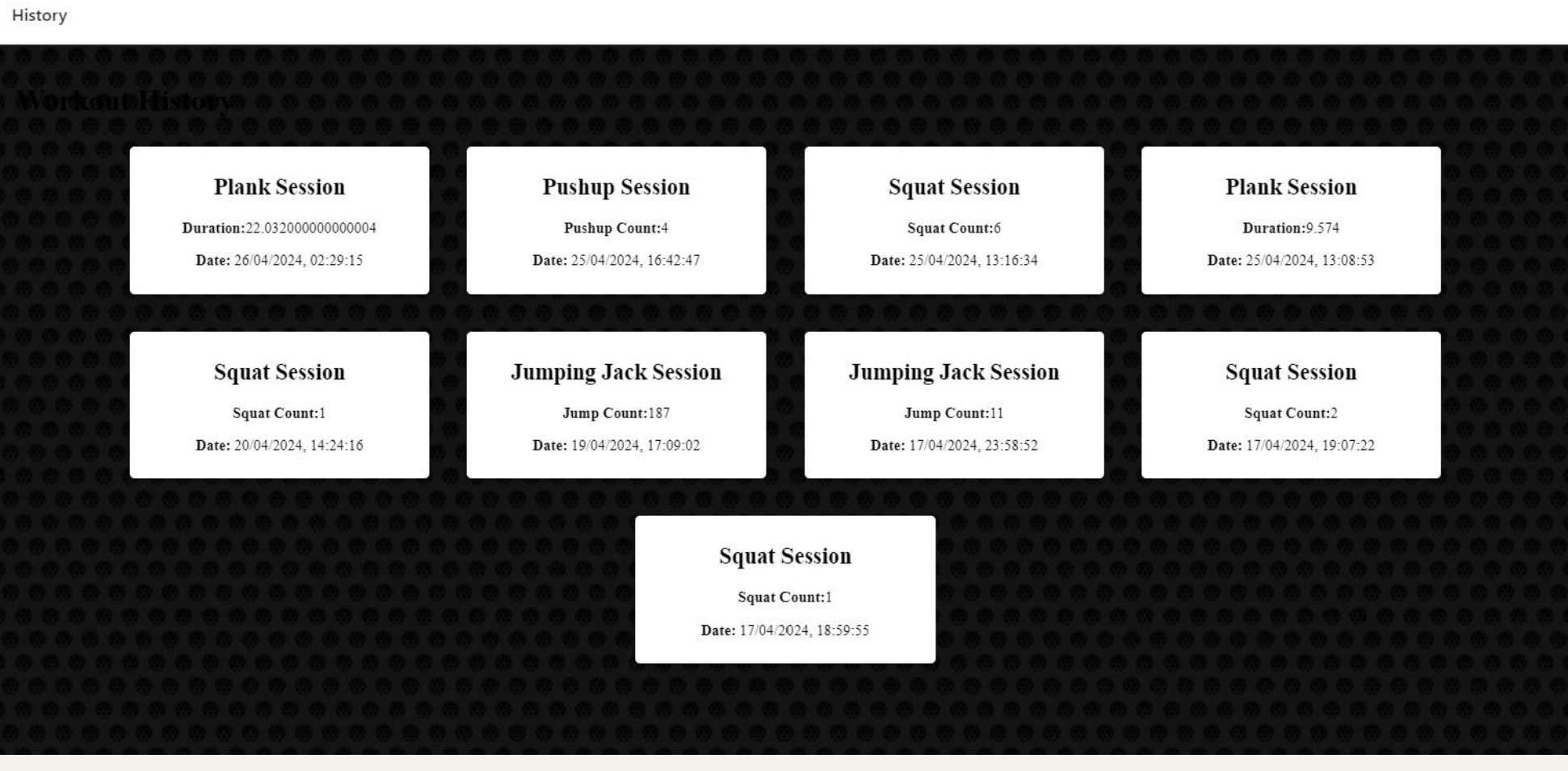
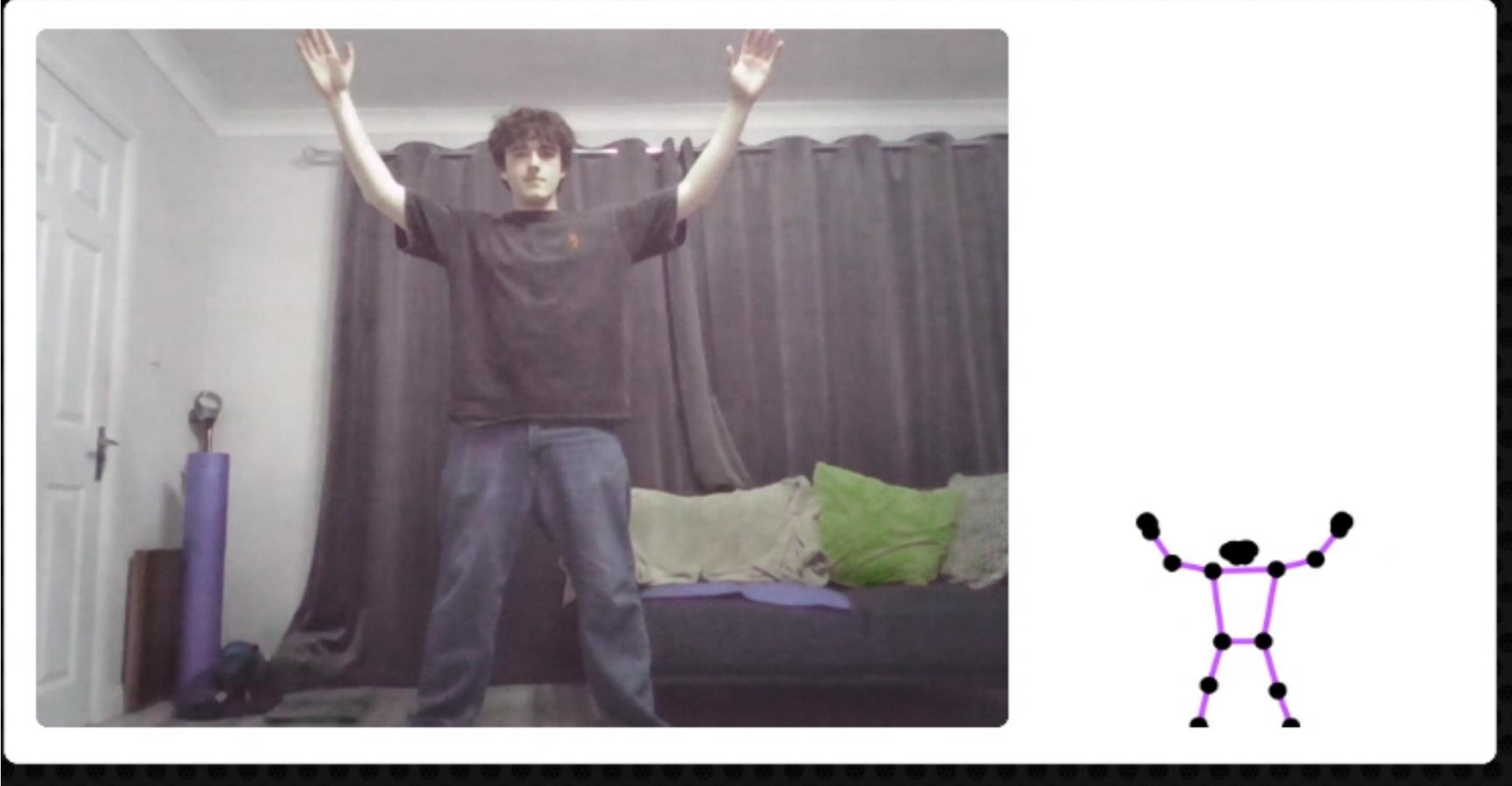
The front-end is built using React Native and managed with React Navigation for routing. The user interface is themed with React Native Paper and includes screens for login, profile stats, historical data, exercise detection, and browsing additional exercises.

Back-end services including databases and server-side applications

Firebase is used for backend services, including user authentication (FirebaseAuth), data storage (Firestore), CDNs, and RapidApi. Firebase provides a robust, scalable backend solution with easy integration into the React Native framework.

Examples of Pose Detection

The application of our project allows the counting of exercises like jumping jacks and planking with additive helpful feedback and You can see your session saved with data and count.



Website: Try Yourself

Curious to learn more? Simply scan the QR code below or visit our website: <https://dep-fitness.onrender.com/>

You will need to sign into and verify your email during the beginning steps to use the Application.

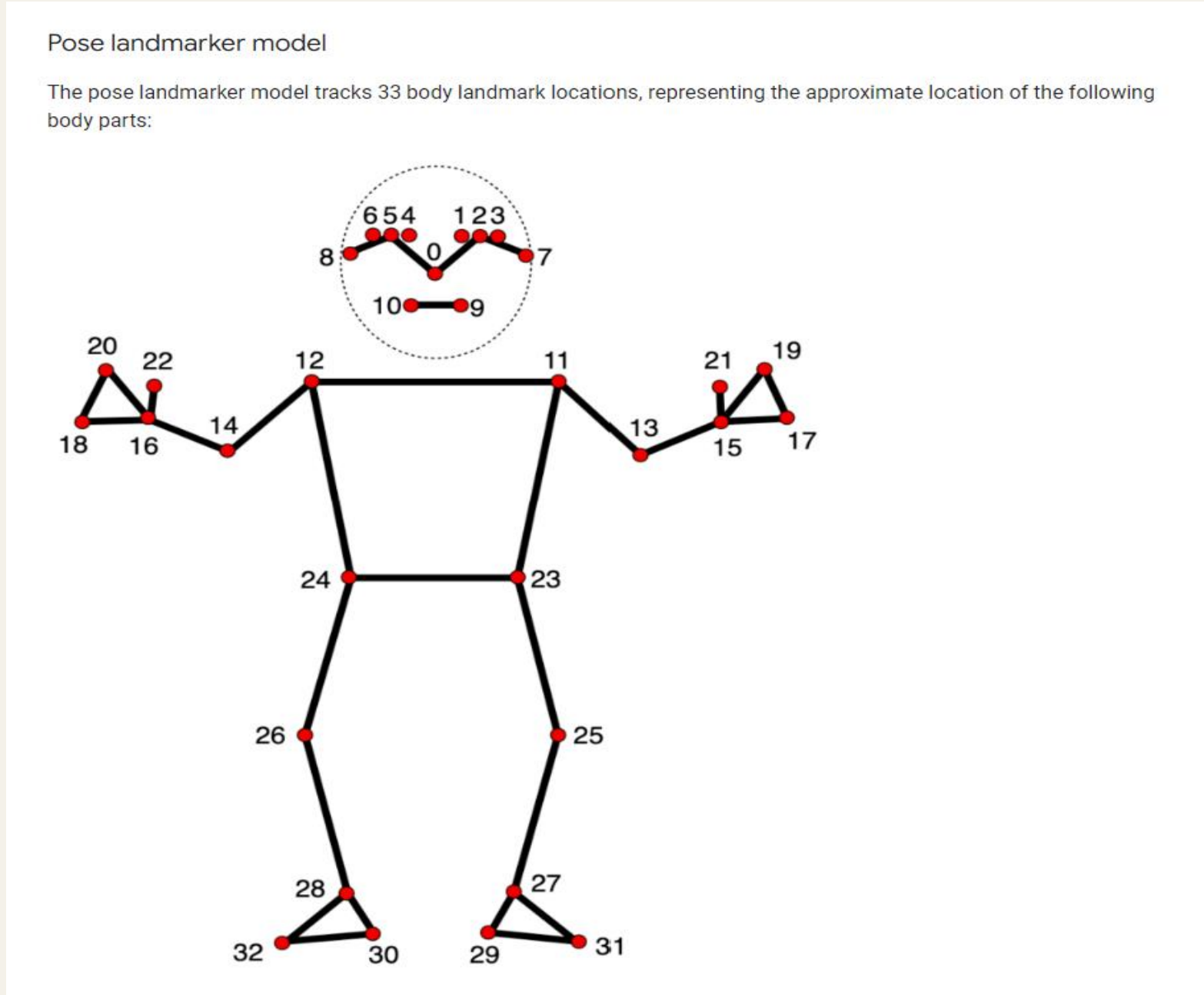
The Application is available on Platforms like Android and Laptops If using IOS, you will be able to access the basic infrastructure, but unfortunately the Scan features will not work with the camera configurations.



MediaPipe Models

MediaPipe is a cross-platform framework by Google that excels in real-time multimedia processing, like pose detection. It uses a modular framework to build complex data processing pipelines, ideal for applications requiring instant feedback from video inputs.

For the Application, I used their Lite Pose Landmark model, this model is made of 32 landmarks. These produced nodes which I could calculate the angles of for precision configuration of exercises. The model is unlike most formats using a “task”

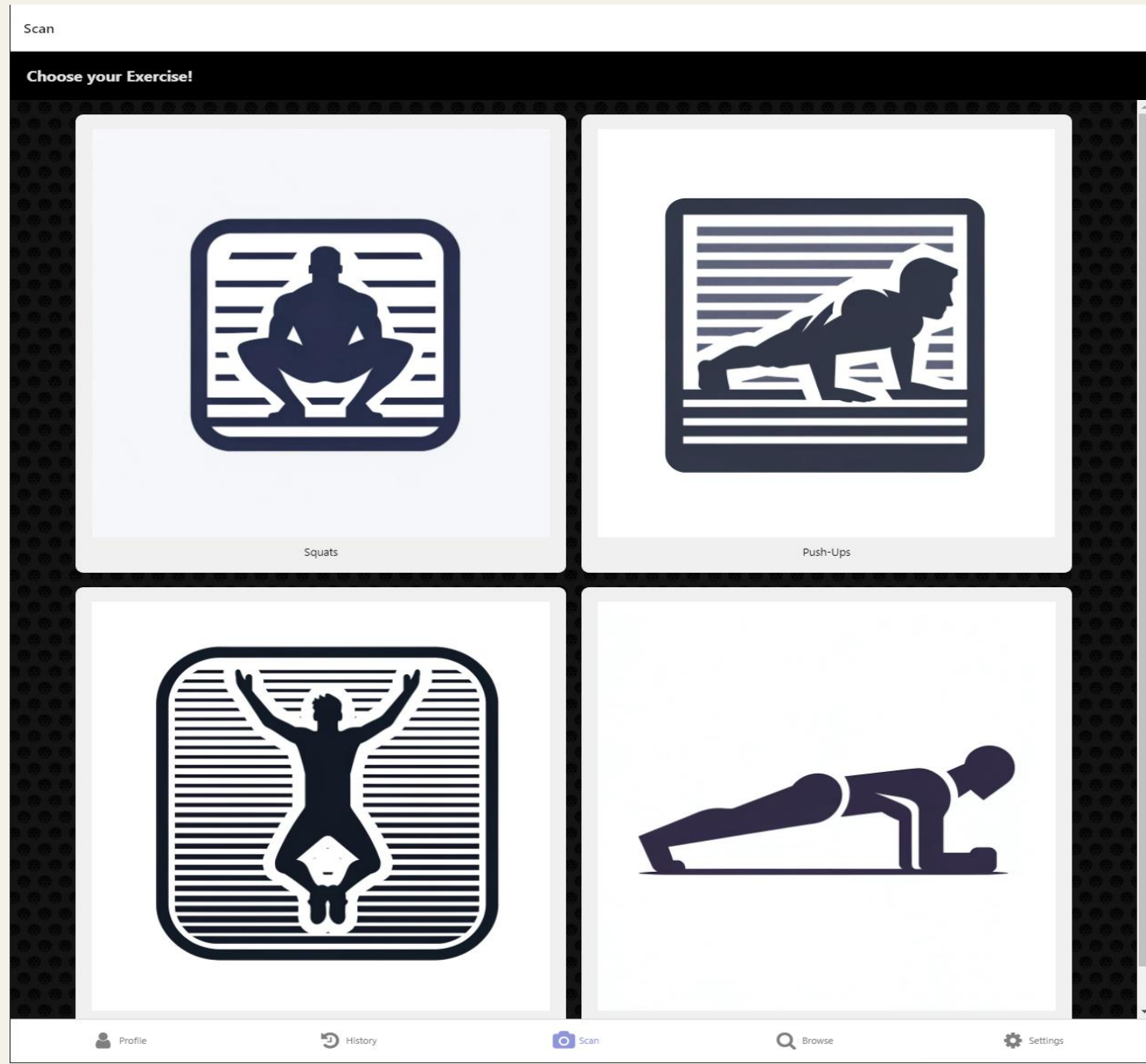


The App

The features of the app

- Login** The login screen uses Firebase Authentication to manage user access securely, ensuring only authorized access to the app's features.
- Navbar** The Navbar facilitates navigation across the app, linking to the Profile, History, and Settings sections.
- History/Profile** Users can view their workout records in the History section and view their entire data focused on a graph in the Profile section, both sourced from Firebase.
- Browse Page** Allows users to search for exercises via an external API, displaying detailed exercise information including animations and targeted muscles.
- Settings** Provides options for a user account to log in or sign out and offers general app information and team introductions.

- Pose Detection Page** This page, built with React Native and React Navigation, displays exercises like squats and push-ups and includes a MediaPipe integration for real-time pose detection using a WASM module for efficient processing.
- Exercise Detection Components:**
 - Squat Detection:** Tracks squats, analyzing form and depth, with feedback provided in real-time.
 - Push-Up Detection:** Monitors push-up form, focusing on elbow angle and body proximity to the floor.
 - Plank Detection:** Assesses plank duration and form by analyzing shoulder and hip angles.
 - Jumping Jack Detection:** Ensures proper arm and leg coordination during jumping jacks.



Acknowledgements

I would like to give thanks to my group supervisor:
Andrew Beatty

I would also like to thank my teammates:
Pedro Figueiredo (Cybersecurity)
Chayapol Hongsrirumuang (Devops)

Also, a special thanks to **Mediapipe** for their free-to-use models.