Hackathon Project Documentation

FitSyncAI

Overview

Project Title	FitSync AI: Real-Time Fitness Adjustments with LLaMA3
Team Name	AnnoX
Team Members	• Pragna Sri Lalitha Padamata • Pallavi Gudupalli • Rishitha Gadiraju

Phase-1: Brainstorming & Ideation

Objective:
Sticking to a fitness routine can be tough, especially when workouts start feeling repetitive or don't show results. Everyone has different goals, and a one-size-fits-all approach doesn't always work. That's where FitSync AI comes in! Powered by LLaMA3, it tracks your progress and personalizes workouts in real-time, keeping your fitness journey exciting, effective, and suited to your needs.

Key Points:

Problem Statement	• Many people struggle to stay motivated and see progress in their fitness journey due to repetitive workouts and lack of personalized guidance. • There is a need for an AI-driven solution provides real-time workout adjustments, nutrition advice to keep users engaged and on track.
Proposed Solution	• Suggests customized workouts for different body parts based on fitness goals (fat loss, muscle gain, endurance, etc.). • Includes set-based exercises with progress tracking and AI adapts exercises daily to prevent stagnation and ensure effective workouts.
Target Users	• Fitness Enthusiasts – Individuals looking for personalized workouts, progress tracking, and motivation. • Beginners & Busy Professionals – Those needing AI-guided exercises, easy-to-follow routines, and time-efficient fitness plans.
Expected Outcome	• Real-Time AI Coaching & Personalized Workouts . Provides instant feedback on posture, form, and performance while generating adaptive fitness plans based on user goals. • Progress Tracking.

Phase-2: Requirement Analysis

Objective

*The technical and functional requirements for the **FitSyncAI** platform.*

Key Points:

Technical Requirements	✅ Frontend – Streamlit (UI), OpenCV (video processing) ✅ Backend – Python, NumPy, Math Library (calculations) ✅ AI & CV – Mediapipe/OpenCV (pose detection) ✅ Hardware – Webcam (real-time exercise tracking)
Functional Requirements	Posture Correction Users can register, log in, and update their profiles to access personalized fitness experiences Performance Tracking & Analytics Personalized Workout Plans/ Customizable Workout remainders
Constraints and Challenges	AI must analyze movement instantly without lag. AI needs large datasets to accurately tailor workouts. Incorrect posture detection due to occlusions or misalignment. **or** AI may struggle with variations in body types, lighting, or background noise. Syncing real-time biometric data across different devices can be challenging.

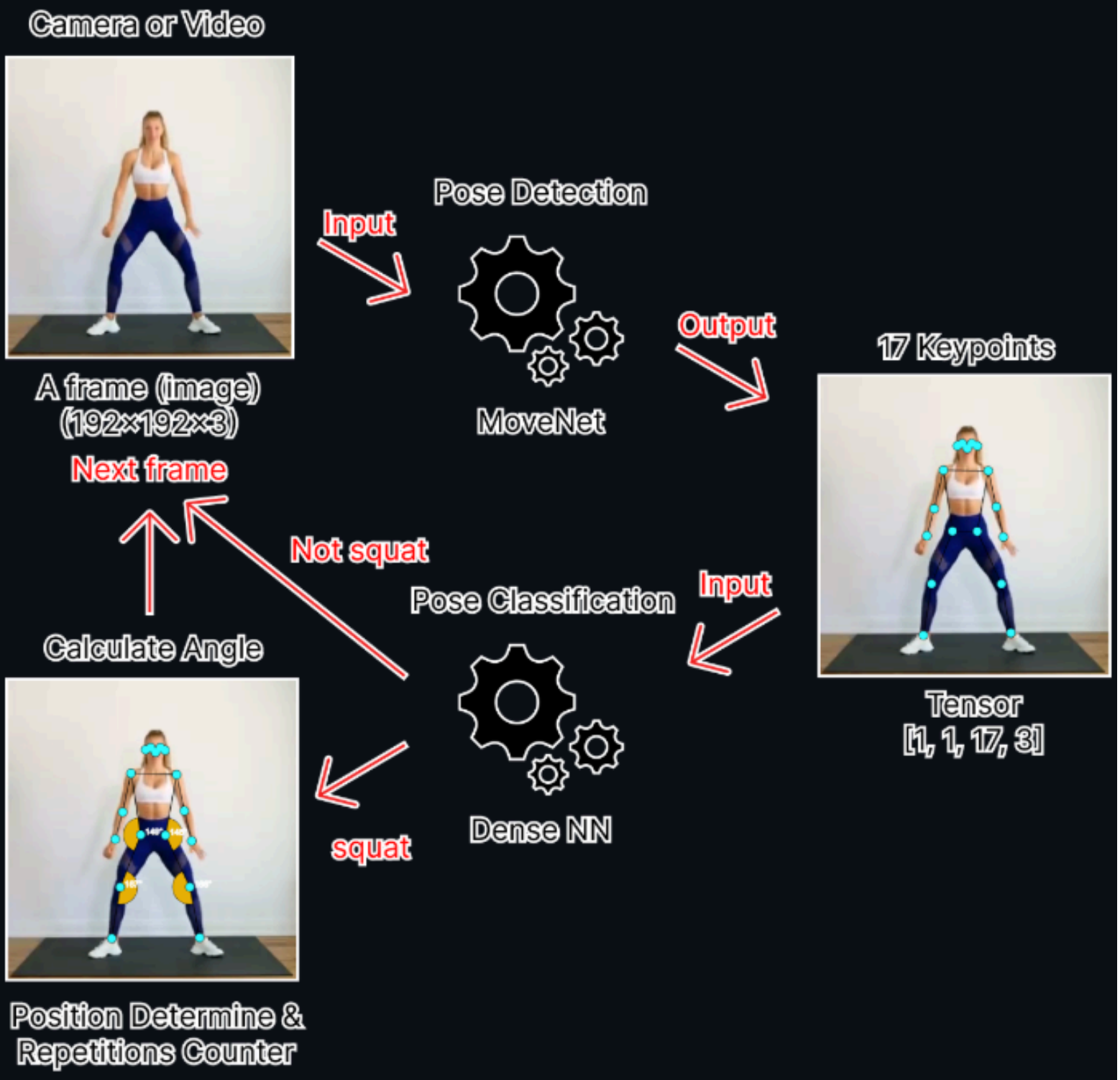
Phase-3: Project Design

Objective

Develop the architecture and user flow of the **FitSyncAI** platform.

How it Work

Image data which is obtained from video or webcam will be processed by pose detector using the MoveNet model to generate keypoints. Keypoints are used for repetition calculations and input for classifying workout types with Dense Neural Network (DNN) model.



Key Points:

System Architecture	• Backend & AI: Python, Mediapipe/OpenCV (pose detection), Flask/FastAPI (API). • Hardware: Webcam (tracking), AI model (analysis), cloud/local processing. 🚀 • Frontend: Streamlit (UI), OpenCV (video), Plotly (visuals).
User Flow	• User selects a workout goal (e.g., fat loss, muscle gain). • AI analyzes movements using pose detection and provides real time feed back • User receives guidance, tracks progress/nutritions.
UI/UX Considerations	• Personalized & Adaptive Interface • AI-Driven Workout Insights & Analytics • Workout Session UI & Customization • User Onboarding & Personalization

Phase-4: Project Planning(Agile Methodologies)

Objective

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned to	Dependencies	Expected outcome
1	Environment Setup & AI Model Integration	🔴 High	6 hours (Day1)	End of day1	Rishitha Pallavi	AI Di usion Model, Python, Streamlit setup	AI Model Integrated & functional
1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of day1	Pragna	UI framework setup	Basic UI with input fields
2	Ai- Trainer (pose detection) Features	🔴 High	3 hours (Day 1)	End of day1	Pallavi	AI model response, UI elements	AI-generated Moves with customization
2	Error Handling and Debugging	🔴High	1.5 hours (Day 2)	Mid -day 2	Pragna Rishitha	API logs, UI inputs	Improved AI performance & stability
3	Testing and UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid -day 2	Pallavi Pragna	AI response,UI layout completed	Responsive UI, Improved user Experience
3	Final Presentation & Deployment	🟢 Low	1 hour (Day 2)	Mid -day 2		Working prototype	Demo-ready platform

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1 and Day 2)

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)
🔴 High: AI model integration (6h) → Functional AI Model
🟡 Medium: UI development (2h) → Basic Input UI
Sprint 2 – Core Features & Debugging (Day 1 & 2)
🔴 High: Pose detection (3h) → AI-generated Moves
🔴 High: Debugging (1.5h) → Improved performance
Sprint 3 – Testing & Deployment (Day 2)
🟡 Medium: UI testing (1.5h) → Enhanced UX
🟢 Low: Final deployment (1h) → Demo-ready platform

Phase-5: Project Development

Objective

Break down development tasks for efficient completion.

Technology Stack

Frontend:
✅ Streamlit - For the web-based UI and interactive elements
✅ OpenCV - For capturing and processing video frames
✅ Plotly - For data visualization and analytics

Backend & Computation:
✅ Python - Core programming language
✅ NumPy - For numerical calculations and interpolations
✅ Math Library - For angle calculations using atan2 and degrees

Computer Vision & AI:
✅ PoseDetector (likely using Mediapipe or OpenCV) - For pose estimation and detecting key body points

Hardware & Integrations:
✅ Webcam (via OpenCV) - Capturing real-time exercise data

Challenges and fixes

Ensuring **real-time responsiveness** in UI and video processing requires optimization techniques like multi-threading and caching.

Accurate pose estimation depends on stable key point detection, which can be improved with better lighting and threshold adjustments.

Reducing computational **delays** involves optimizing numerical calculations, enhancing hardware settings, and refining AI model efficiency.

Phase-6: Functional & Performance Testing

Objective

Ensure accurate AI-driven workout tracking, seamless user experience and system efficiency under load.

Objective:

Ensure that the FitSync App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	AI posture correction during squats	AI should detect incorrect posture and provide real-time feedback	⏳ 70% Accuracy	Tester 1
TC-002	Functional Testing	Nutrition tracker with visualization	App should display accurate nutrition data with graphical insights	✅ 100% Accuracy	Tester 2
TC-003	Performance Testing	AI response time under 1 minute	AI should provide real-time feedback within 60 seconds	⚠️ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect calorie tracking values	Data accuracy should be improved	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices	UI should work seamlessly on mobile & desktop	❌ Failed - UI misaligned on tablets	Tester 4
TC-006	Deployment Testing	Host the app with cloud sync capabilities	App should be accessible online with synced data	⏳ On Processing	DevOps

Final Submission:

1. Project Report Based on the templates: Link
2. Demo Video (3-5 Minutes):Link
3. GitHub/Code Repository : Link
4. Presentation