TAMASOMA JYOTIRGAMAYA
VIGNANA
JYOTHI

VNRVJIET

VALLURUPALLI NAGESWARA RAO
VIGNANA JYOTHI INSTITUTE OF
ENGINEERING & TECHNOLOGY

AUTONOMOUS INSTITUTION

KRITHOMEDH

AI-WEEK
Vibe Coding Hackathon

Team Details

Team Member Name	Phone Number
Sajja Lakshmi	9182337878
Ayushi Singh	8639402980
Nadimpalli Navya	7287039999

Problem Statement

Problem: STEM students struggle to understand physics word problems due to difficulty visualizing abstract concepts like motion, forces, and relationships.

Goal: Build an AI system that converts natural language physics problems into interactive visual simulations.

Core Requirements:

1. Natural language input - Accept physics word problems as text
2. AI interpretation - Extract entities, parameters, and physical relationships
3. Auto-generate simulations - Create dynamic, interactive visualizations
4. Real-time manipulation - Let users adjust parameters and instantly see effects
5. Intuitive interface - Simple UI for exploration and "what-if" scenarios

Proposed Solution

Core Concept

Transform abstract physics problems into visual, interactive simulations that students can explore and understand intuitively.

Key Features

- **Visual Scenario Selector**- Choose from physics modules (Projectile, Free Fall, Friction)
- **Real-Time Parameter Control-** Adjust values with sliders, see instant updates
- **Live Canvas Animation**- Smooth 60fps physics rendering
- **Smart Insights**- Physics calculations and what-if predictions during simulation

System Architecture

Three-Layer Design

1. Frontend Layer (Vanilla JS + HTML5 Canvas)

- ScenarioManager - Handles module loading and UI rendering
- BaseScenario - Abstract class for all physics modules
- Individual Scenarios - Projectile, FreeFall, Friction (pluggable modules)
- Canvas Renderer - 60fps animation engine

2. Backend Layer (Flask/Python)

- REST API endpoints
- Physics calculation engine
- AI integration layer (ready for LLM)

3. AI Layer

- Problem text parsing (Groq)
- Scenario classification
- Parameter extraction

System Architecture

- **Data Flow**

  User Input → Scenario Selector → Parameter Controls → Physics Engine → Canvas Renderer → Visual Output + Insights

- **Modular Architecture**

```
BaseScenario (Abstract)
 ├──── ProjectileScenario
 ├──── FreeFallScenario
 ├──── FrictionScenario
```

Tech Stack

Frontend

- HTML5 Canvas- High-performance physics animations
- JavaScript (ES6) - Modular architecture with classes
- CSS3 - Modern UI with gradients and animations

Backend

- Flask- Lightweight Python web framework
- Python 3.x- Physics calculations and server logic

AI Integration

- Grok API - Natural language problem parsing
- LM- Scenario classification and parameter extraction

Development Tools

- Git- Version control
- pip - Python package management

Thank You