Mixed Intelligence

Optimized Code Generator

Group Project Workflow

Mixed Intelligence

Introduction

Welcome to Optimized Code Generator

Purpose

Efficient code generation using advanced AI models.

Mission

Simplify coding tasks, improve quality, and foster learning.

Team Members

Umar Majeed

Laiba Asif

Izwa Areeb

Kainat Ijaz

Rajesh Yarra

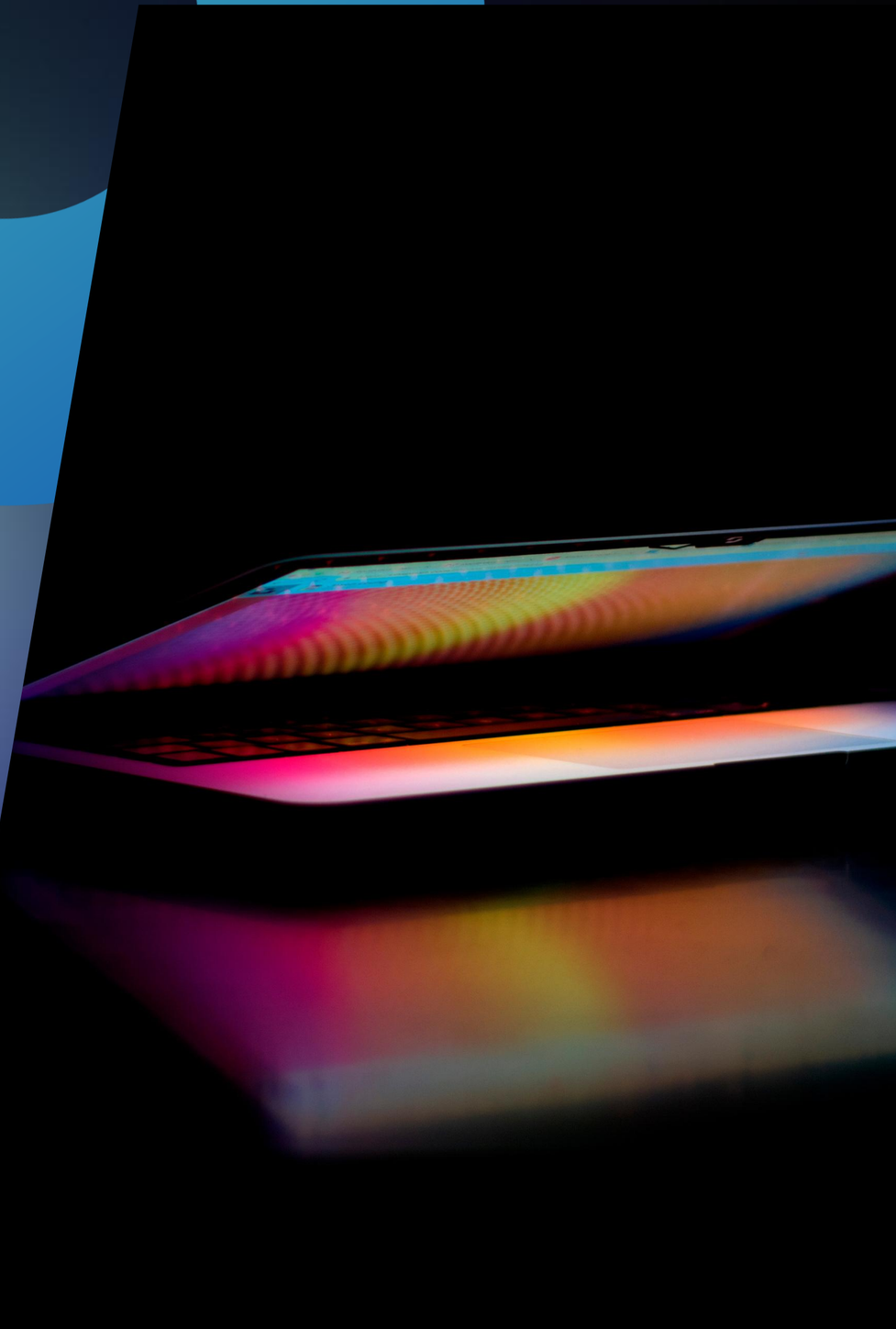
Rosanna Mannan

Project Overview

Platform: Streamlit-based, interactive interface

Core Features:

- Generate optimized code with AI models.
- User can compare the results of the different models
- Line-by-line explanations
- Visualized complexity analysis (time & space)



Mixed Intelligence

Key Features

User Inputs

- Choose programming language (Python, Java, C++, etc.).
- Select base model (o1-preview, o1-mini).
- Compare models (Llama_3.2, GPT4o, Gemini, Mistral.).

Analysis

- Code time and space complexity with interactive graphs

How It Works

User Input

Enter coding problem, choose language & models.

Code Generation

Models generate code and explanations

Complexity Analysis

Visualized complexity results (time & space)

Comparison Model

Evaluate code from multiple models side by side.

Technology Stack

- Frontend: Streamlit for rapid UI development.
- AI Models: OpenAI o1-preview, o1-mini, Llama_3.2, GPT4o, Gemini, Mistral.
- Backend: Python with libraries like streamlit, time, custom modules.

Testing & Results

Tested on recent MetaHackerCup 2024 problems.

HACKER CUP

Problem A: Subsonic Subway

13 points

OpenAI o1-mini

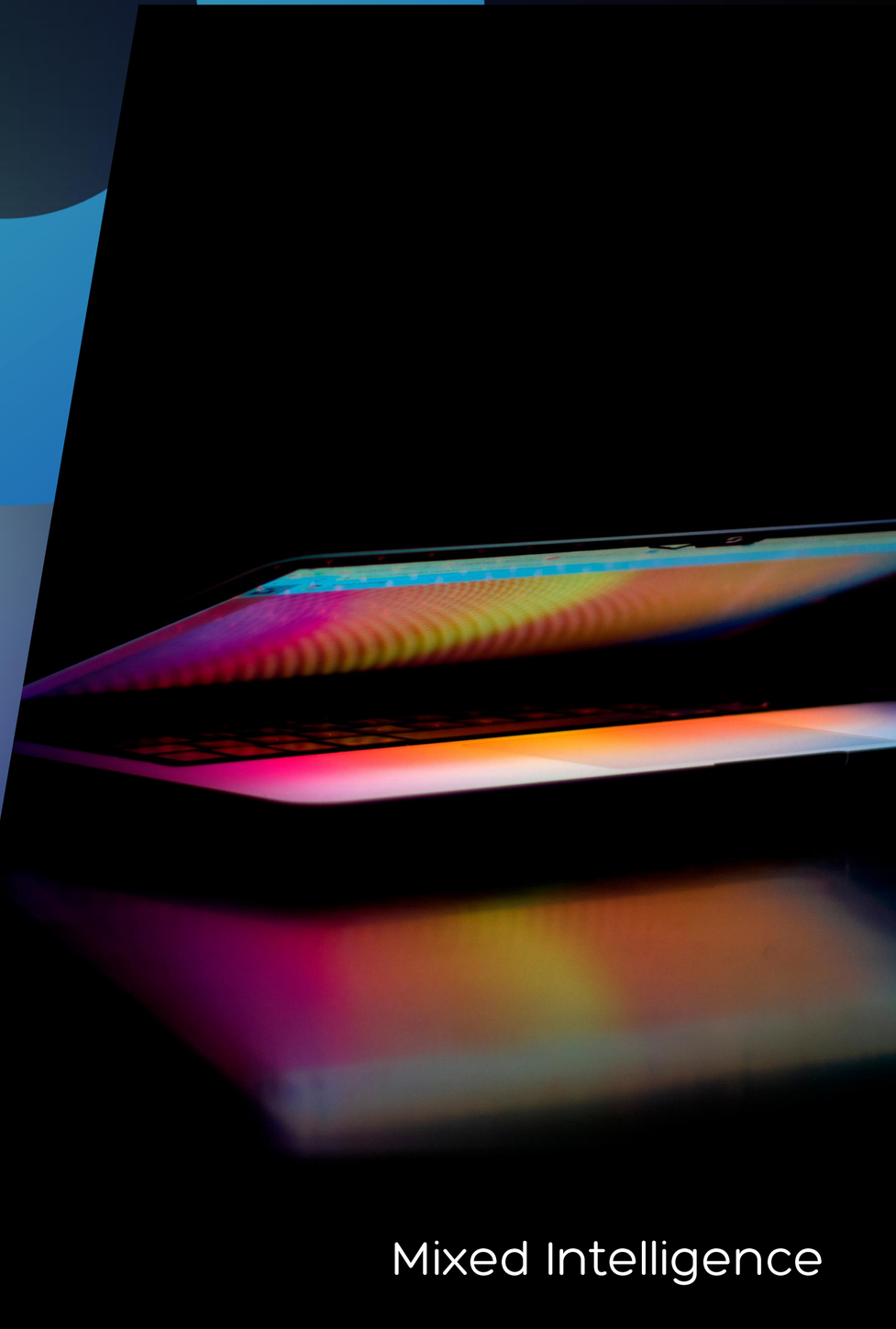
Problem B: Prime Subtractorization

18 points

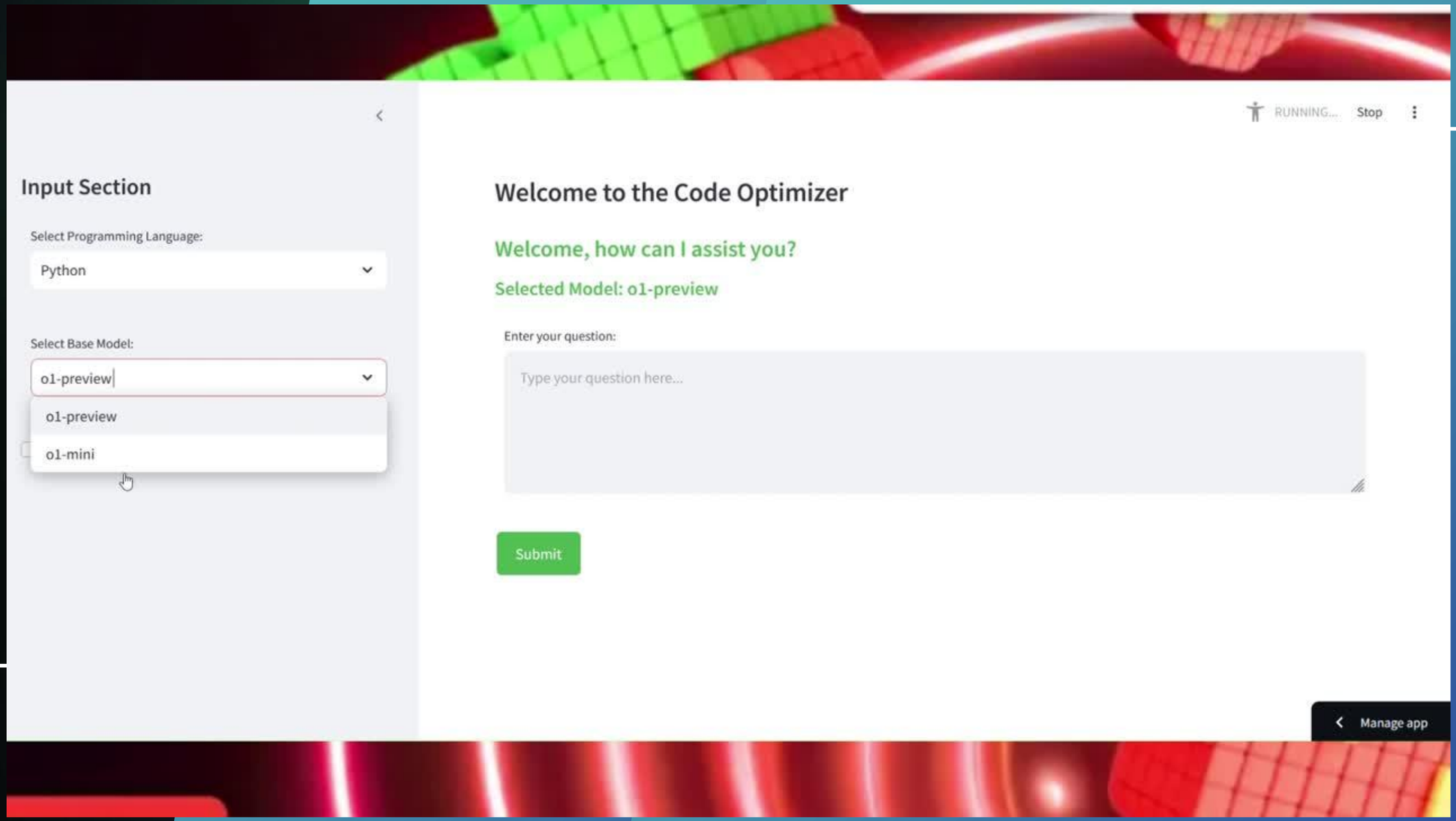
Problem C: Substantial Losses

20 points

DEMO



Mixed Intelligence


RUNNING...
Stop
Input Section
Select Programming Language:
Python
Select Base Model:
o1-preview
o1-preview
o1-mini
Welcome to the Code Optimizer
Welcome, how can I assist you?
Selected Model: o1-preview
Enter your question:
Type your question here...
Submit
Manage app

Future Enhancements

- Multi-model comparison expansion.
- User authentication.
- Advanced complexity visualization.

Thank You

Mixed Intelligence