Role-based vs Chain-of-Thought AI Comparison App

1. Objective

The goal of this project is to compare AI outputs generated using role-based prompts versus chain-of-thought (CoT) prompts.
This helps users learn how different prompt strategies affect the clarity, reasoning, and level of explanation in AI-generated responses.

2. Instructions & Implementation

Step 1: Select a Task

- Example task: *Explain how photosynthesis works* or any other concept requiring explanation.
- The task should be suitable for observing differences between **role-based explanations** and **step-by-step reasoning**.

Step 2: Role-based Prompting

Definition: In role-based prompting, the model is asked to assume a specific role (e.g., teacher, expert, student) and generate output according to that role.

Prompt Example:

```
css                                                    Copy  Edit

You are a high school biology teacher. Explain photosynthesis to students in simple words.
```

Implementation (Hugging Face Model):

- Model: google/flan-t5-large (default)
- Task: text2text-generation
- Parameters: max length, temperature, top-p
- Output Example:
  (Model generates a simplified teacher-friendly explanation.)

Step 3: Chain-of-Thought (CoT) Prompting

Definition: In chain-of-thought prompting, the model is asked to reason step by step before giving the final answer. This improves transparency and detail.

Prompt Example:

```vbnet
Explain photosynthesis step by step, reasoning each step clearly.
```

Implementation (Hugging Face Model):

- **Model: google/flan-t5-base (default)**
- **Task: text2text-generation**
- **Parameters: max length, temperature, top-p**
- **Output Example:**
  (Model generates a detailed stepwise explanation.)

Step 4: Compare Outputs

- The app generates both outputs side by side.
- Diff and reflection tools highlight:
  - Differences in phrasing
  - Reasoning depth
  - Similarities or divergences in content

3. Observations

- **Role-based output** is concise, simple, and oriented to the assumed role.
- **CoT output** provides stepwise reasoning, enhancing interpretability.
- Outputs may vary in **detail**, **clarity**, and **structure** depending on the prompt style.
- CoT is particularly useful for complex or multi-step tasks.

4. Deliverables

- Streamlit App:
  a. Interactive interface to enter prompts and run both models.
  b. Displays outputs, diffs, and reflections in real time.

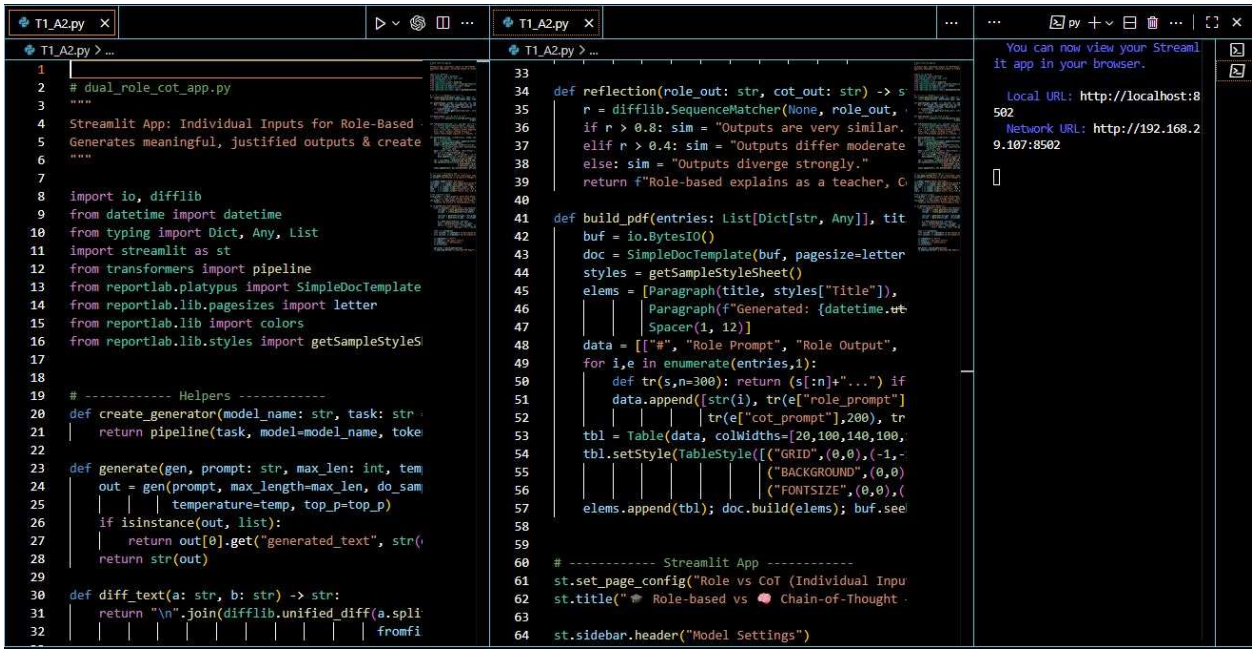

Fig1: Input Snapshot

- PDF Report: Automatically generated with:
    - User inputs (role-based and CoT prompts)
    - Model outputs
    - Diff table highlighting differences
    - Reflections and similarity scores

5. Output

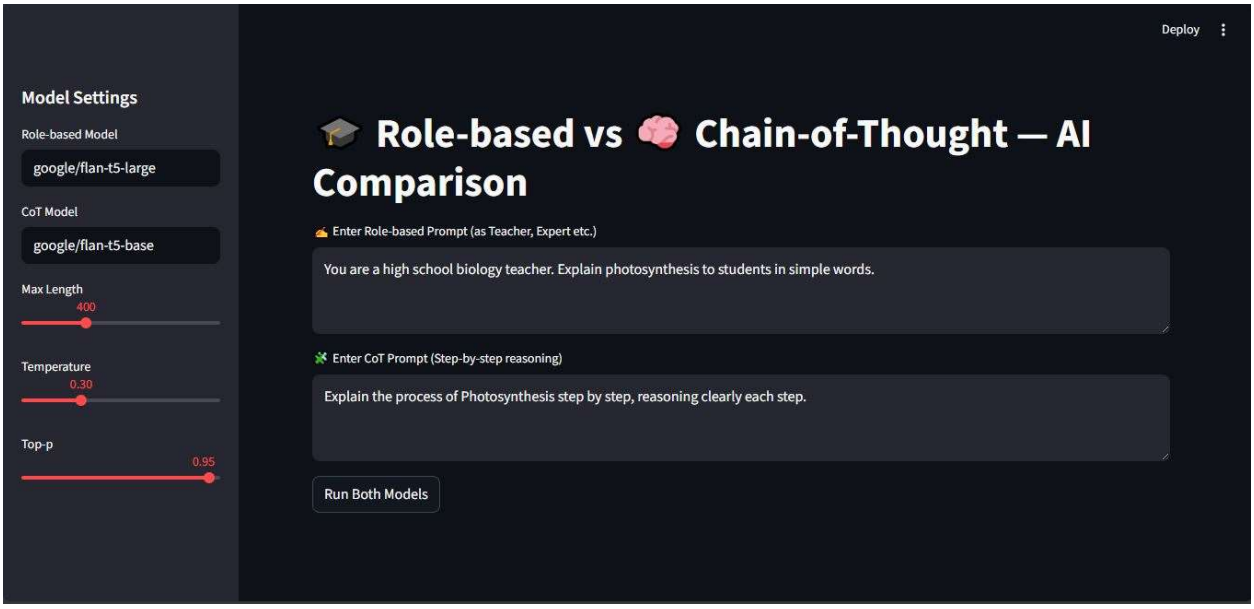

Fig2: Output Snapshot1

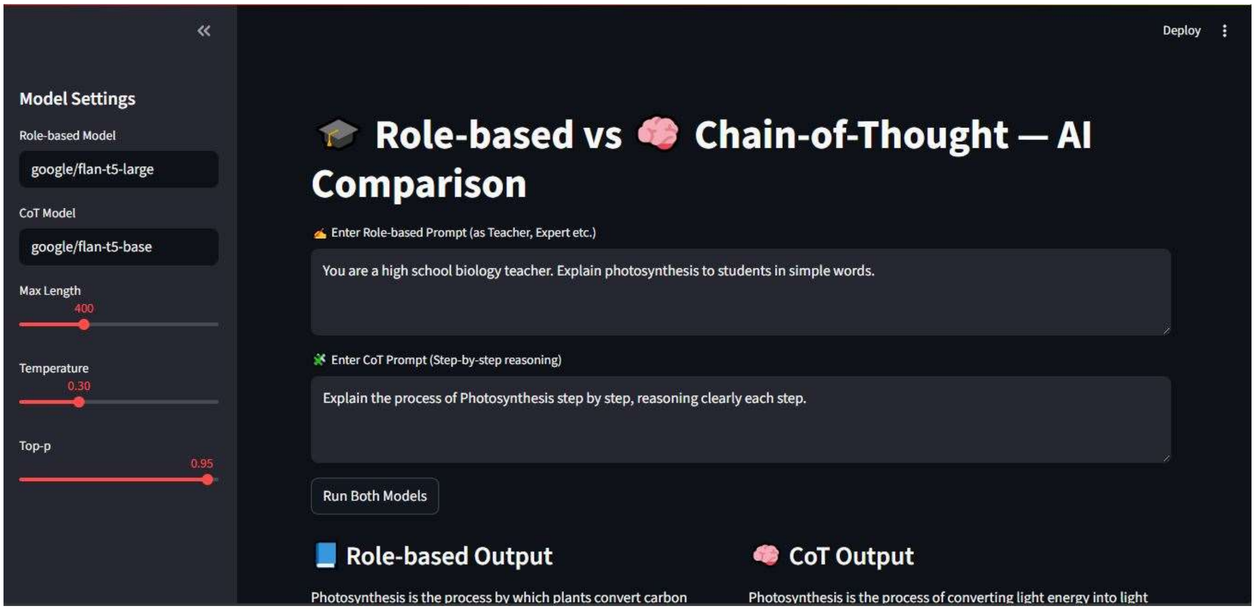

Fig3: Output Snapshot2

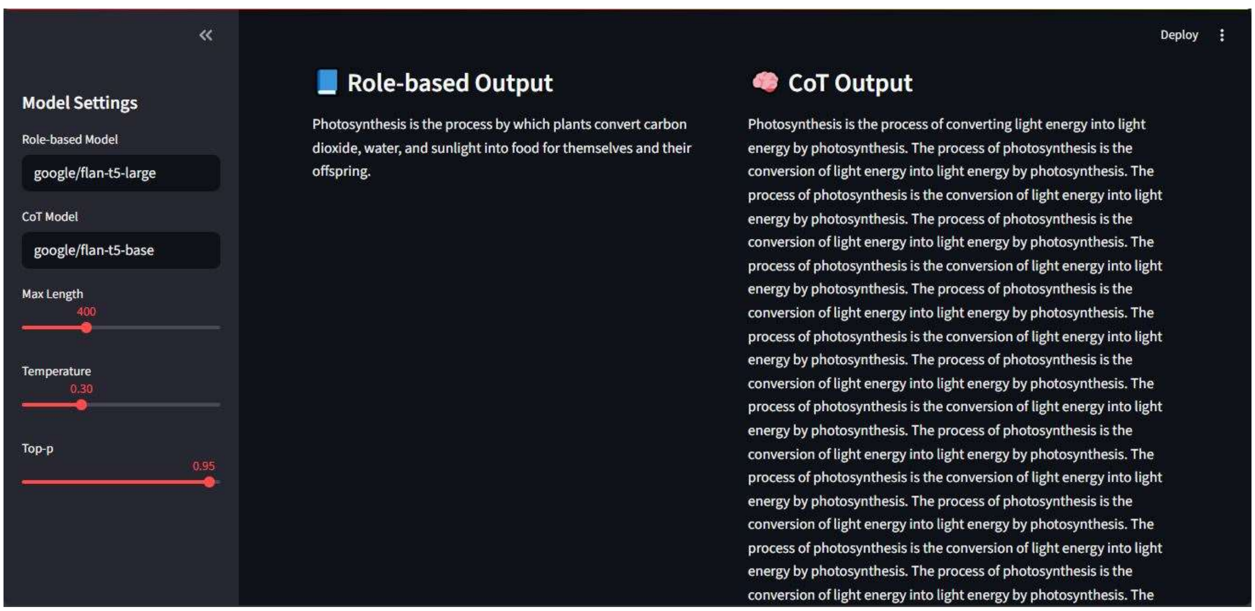

Fig4: Output Snapshot3

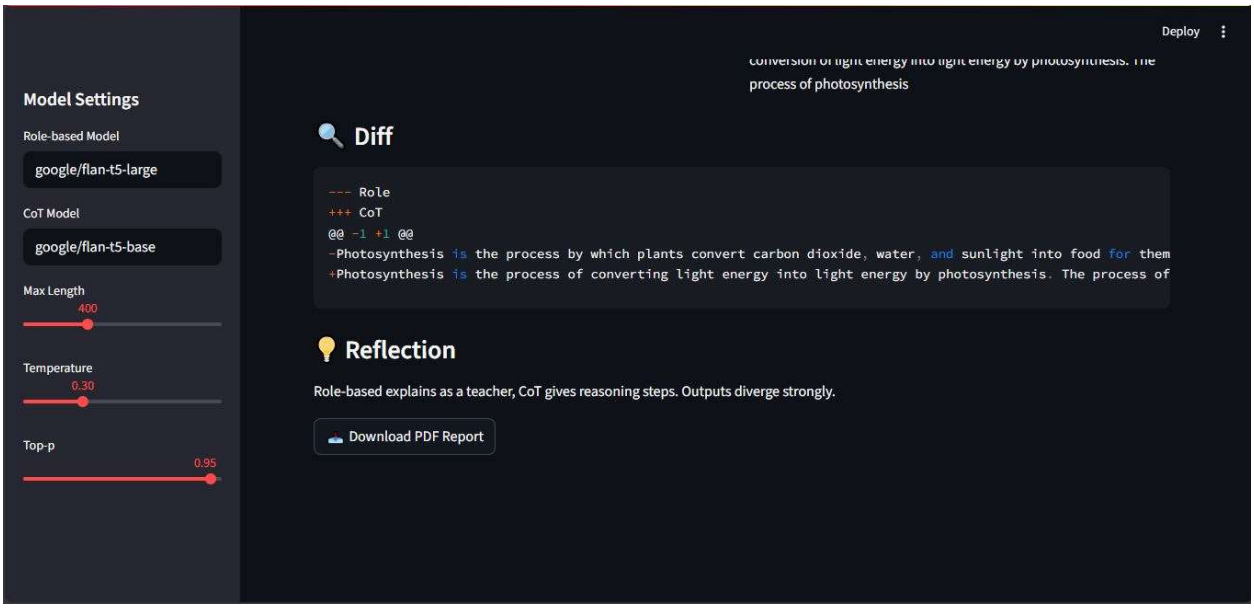

Fig4: Output Snapshot3