

## Project Breakdown

Your project has two main components:

- AI Model: A recommendation system that suggests cognitive activities based on the provided parameters.
- Backend API: A Flask-based RESTful API with PostgreSQL to serve activity recommendations.

# Project Documentation: Cognitive Activity System

## Project Overview

This project is an AI-powered Cognitive Activity Recommendation System that generates personalized cognitive activities based on user input (Cognitive Skills, Zone, and Age Range). It uses a fine-tuned T5 model for activity generation and stores results in a PostgreSQL database via a Flask REST API.

---

## Project Approach

### Objective:

Convert raw cognitive activity data into a structured dataset suitable for fine-tuning a text-generation model (T5).

#### ◆ Actions Taken:

1. Loaded the dataset (cleaned\_cognitive\_activities.csv) using Pandas.
2. Extracted cognitive skills from Yes/No columns (Memory, Reasoning, etc.).
3. Generated input prompts for the model:

CopyEdit

"Generate an activity for Zone: Green, Age Range: 4-10, Cognitive Skills: Memory, Reasoning"

Created structured output text in the format:

CopyEdit

Activity: Memory Puzzle

Description: A fun game that enhances memory skills.

Instructions: Step 1: Lay out cards face down. Step 2: Flip two cards at a time...

Materials Required: Cards, Paper

Time Required: 15 minutes

Zone: Green

Objective: Improve memory retention and recall.

4. Saved the final dataset in JSON format (activity\_dataset.json) for training.

## Step 2: Fine-Tuning the T5 Model

Objective:

Train the T5-small model to generate cognitive activities from structured prompts.

### ♦ Actions Taken:

1. Loaded the dataset using `datasets.load_dataset()`.
  2. Tokenized input and output texts using `T5Tokenizer`.
  3. Fine-tuned the model with:
    - Batch size: 8
    - Epochs: 3
    - Sequence length: 256 (input) & 512 (output)
    - Mixed precision (fp16) to optimize GPU usage.
  4. Saved the fine-tuned model (activity\_generator\_model) for deployment.
- 

## Step 3: Testing the Model in Google Colab

Objective:

Ensure the model generates correct activities before deployment.

### Actions Taken:

1. Loaded the trained model from `activity_generator_model`.
  2. Generated test activities using: `python`  
`CopyEdit`  
`print(generate_activity("Memory, Reasoning", "Green"))`
  3. Verified output format to match expectations.
- 

## Step 4: Deploying the Model via Flask API

Objective:

Create a REST API that accepts user input and returns AI-generated cognitive activities.

### Actions Taken:

1. Set up a Flask API (app.py).
2. Loaded the fine-tuned T5 model inside Flask.
3. Created an API endpoint (/generate\_activity) that accepts POST requests with parameters:
  - cognitive\_skills (required)
  - zone (optional)
4. Generated new activities dynamically using the model.

Formatted the API response as JSON with:

json

CopyEdit

```
{
  "Activity": "Memory Puzzle",
  "Description": "A fun game that enhances memory skills.",
  "Instructions": "Step 1: Lay out cards face down. Step 2: Flip two cards at a time...",
  "Materials Required": "Cards, Paper",
  "Time Required": "15 minutes",
  "Zone": "Green",
  "Objective": "Improve memory retention and recall."
}
```

5. Tested the API with curl, Postman, and Python requests.
- 

## Step 5: Connecting the API to PostgreSQL (Optional)

### Objective:

Store and retrieve generated activities using PostgreSQL.

### Actions Taken:

1. Created a PostgreSQL database (cognitive\_activities).
2. Defined a Flask SQLAlchemy model (Activity).
3. Stored generated activities in the database for future use.

## Step 6: How to Run.

- Run the Cleaning file to clean and restructure the dataset
- Run the ModelTraining on Google Colab with cleaned\_cognitive\_activities
- Run app.py at [127.0.0.1:5000/generate\\_activity](http://127.0.0.1:5000/generate_activity)

A curl command in CMD will give out the generated text:

```
curl -X POST "http://127.0.0.1:5000/generate_activity" -H "Content-Type: application/json" -d
"{ \"cognitive_skills\": \"Memory, Reasoning\" }"
```

