MAGIC MIRROR WITH AI

DATE
15/12/2023

MAGIC MIRROR WITH AI

Presentation are communication tools that can be used as demontrations, lectures, reports, and more. it is mostly presented before an audience.

KBTU UNIVERSITY

PRESENTED BY
Berdibek Magzhan

INTRODUCTION

The project "Magic Mirror with AI" combines the concept of a traditional smart mirror with advanced artificial intelligence capabilities, creating an interactive and intelligent mirror that can provide a range of services and information to users. This innovative project transforms an ordinary mirror into a smart, personalized interface capable of responding to voice commands, recognizing faces, and delivering relevant information.

KEY FEATURES:

Voice Recognition and Interaction:
The magic mirror utilizes advanced speech recognition technology to understand and respond to voice commands. Users can interact with the mirror by simply speaking, enabling a hands-free and intuitive user experience.

Face Recognition:
The mirror is equipped with face recognition capabilities, allowing it to identify users and provide personalized responses. This feature enables the mirror to tailor its content and information based on the individual using it.

Natural Language Processing (NLP):
The mirror incorporates Natural Language Processing (NLP) capabilities powered by platforms like Wit.ai. This allows the mirror to understand and interpret user intentions expressed in natural language, making interactions more conversational and user-friendly.

Information Retrieval and Display:
Through integration with various APIs and modules, the mirror can retrieve and display real-time information such as weather updates, news headlines, calendar events, and more. Users can access relevant data with a simple voice command.

KEY FEATURES:

Personalized Content:
The mirror provides personalized content based on user preferences and historical interactions. It can display tailored information, such as personalized greetings, news articles of interest, and reminders.

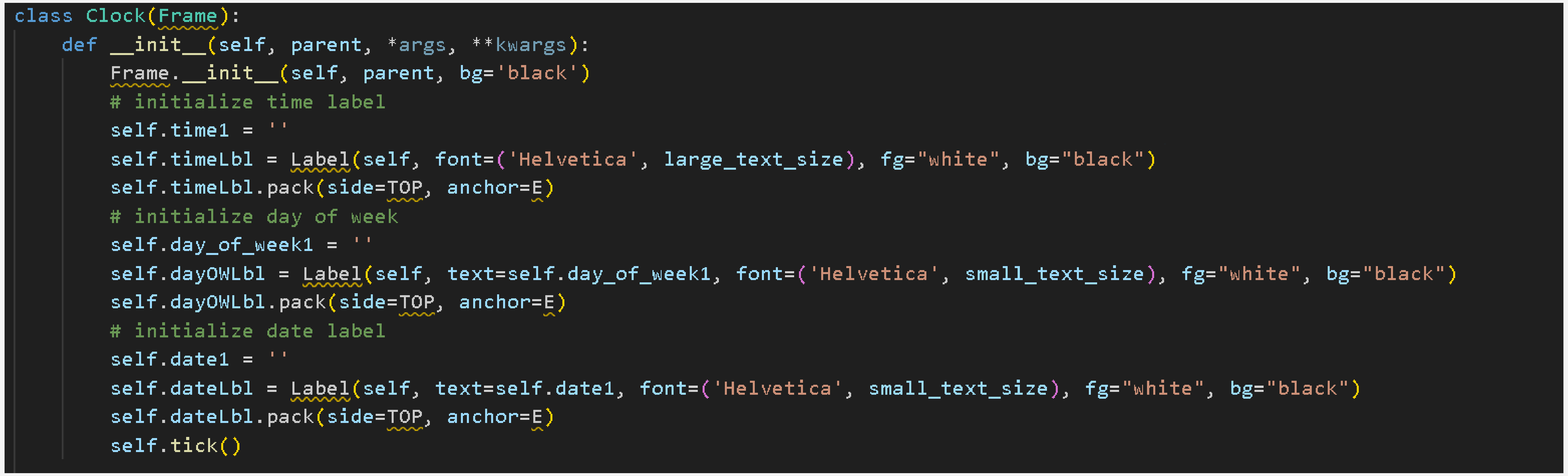
Entertainment and Interactivity:
The mirror is not only informative but also entertaining. It can tell jokes, provide interesting facts, and engage users with interactive features. This adds an element of fun and engagement to the overall user experience.

AI-driven Decision Making:
The mirror uses artificial intelligence algorithms to make decisions and choose appropriate responses based on the context of user queries. This enhances the mirror's ability to adapt to various scenarios and user needs.

Modular Architecture:
The project follows a modular architecture, making it extensible and allowing for easy integration of new features and functionalities. Modules for speech recognition, knowledge retrieval, vision, and more work seamlessly together to create a cohesive system.

IMPLEMENTATION

Setup Environment:
Install pip.



```python
class Clock(Frame):
    def __init__(self, parent, *args, **kwargs):
        Frame.__init__(self, parent, bg='black')
        # initialize time label
        self.time1 = ''
        self.timeLbl = Label(self, font=('Helvetica', large_text_size), fg="white", bg="black")
        self.timeLbl.pack(side=TOP, anchor=E)
        # initialize day of week
        self.day_of_week1 = ''
        self.dayOWLbl = Label(self, text=self.day_of_week1, font=('Helvetica', small_text_size), fg="white", bg="black")
        self.dayOWLbl.pack(side=TOP, anchor=E)
        # initialize date label
        self.date1 = ''
        self.dateLbl = Label(self, text=self.date1, font=('Helvetica', small_text_size), fg="white", bg="black")
        self.dateLbl.pack(side=TOP, anchor=E)
        self.tick()
```

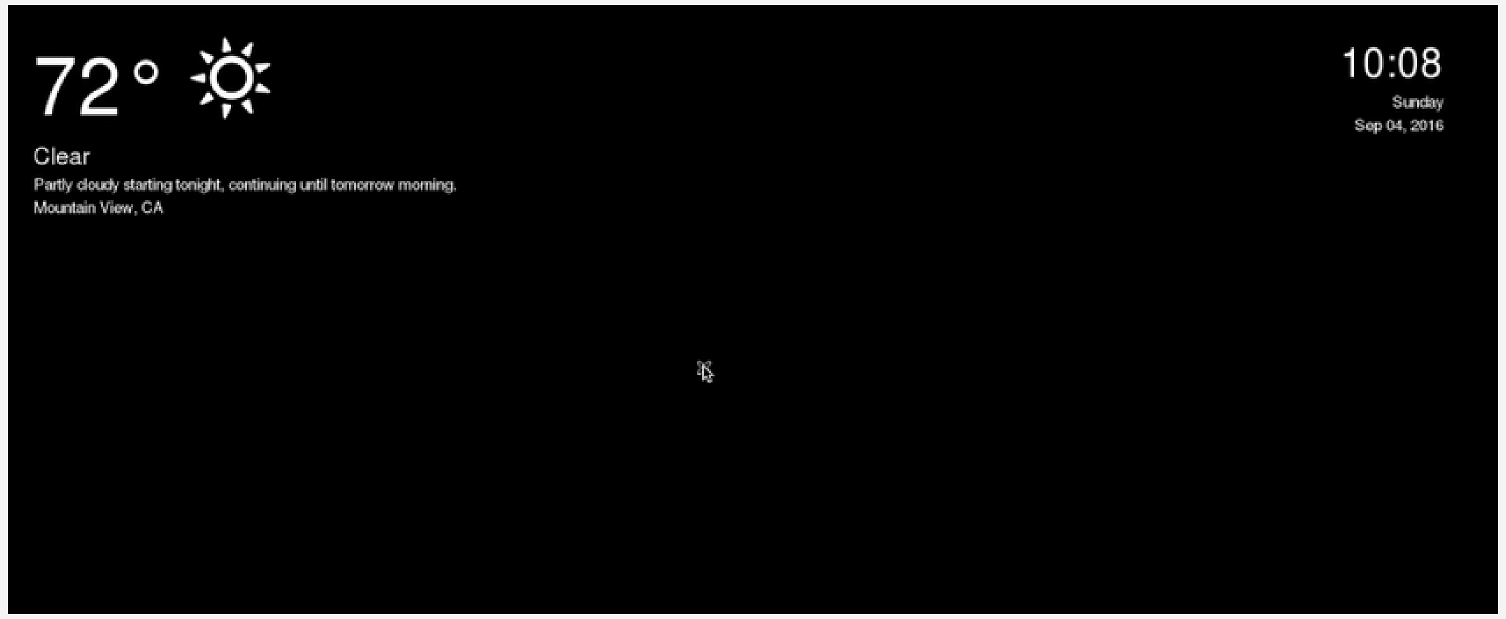
CODE:

brief overview of key components:

1. **Clock:**
    - Widget for displaying the current time, day of the week, and date.
    - Uses the Tkinter library to create the graphical interface.
    - Automatically updates the displayed time every 200 milliseconds.
2. **Weather:**
    - Widget for displaying weather information, such as temperature, forecast, and location.
    - Uses the Dark Sky API to obtain weather data.
    - Weather icons are represented as images corresponding to the current weather conditions.
3. **News:**
    - Widget for displaying news headlines.
    - Utilizes the Google News RSS feed to fetch the latest news.
4. **Calendar:**
    - Widget for displaying calendar events (not yet implemented).
5. **FullscreenWindow:**
    - Class for creating a fullscreen window containing clocks, weather information, and news.

The code also includes event handling, such as toggling fullscreen mode with the "Enter" key and exiting fullscreen mode with the "Esc" key.

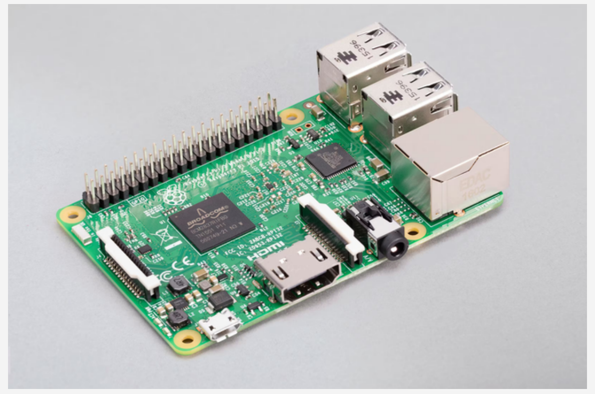
RESULT:



72°
Clear
Partly cloudy starting tonight, continuing until tomorrow morning.
Mountain View, CA

10:08
Sunday
Sep 04, 2016

+



-->

Connect Raspberry PI 3 by hdmi to monitor

Make the construction and
use acrylic mirror

AI IMPLEMENTATION

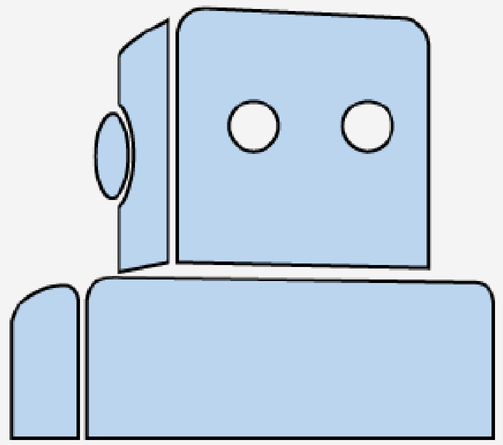
Implementing the "Magic Mirror with AI" project involves integrating various technologies and building the necessary components. Below is a high-level outline of the steps involved in implementing this project:

Setup Environment:
First I set up the development environment with the required software and dependencies. It includes installing Python, necessary libraries, and frameworks for AI, speech recognition, and other functionalities. Also Ìve downloaded the stable version of Node.js,Ruby and Homebrew. For openCV Ìve used this guide https://www.learnopencv.com/install-opencv-3-on-yosemite-osx-10-10-x/.

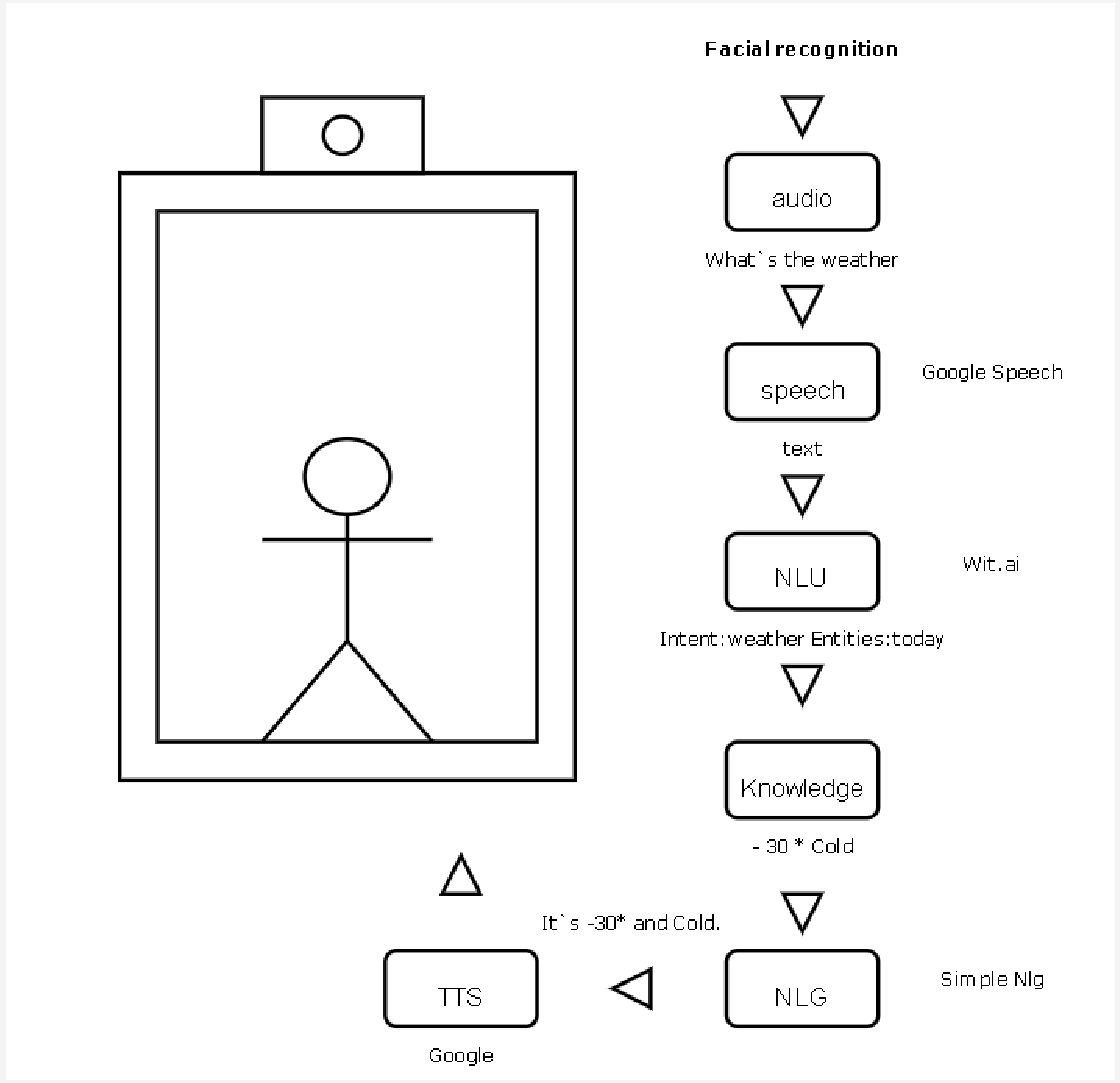
AI IMPLEMENTATION

Speech Recognition:
To Implement speech recognition to enable the mirror to understand and process voice commands we will use portable speaker.

How it works:



NLP=NLU+NLG

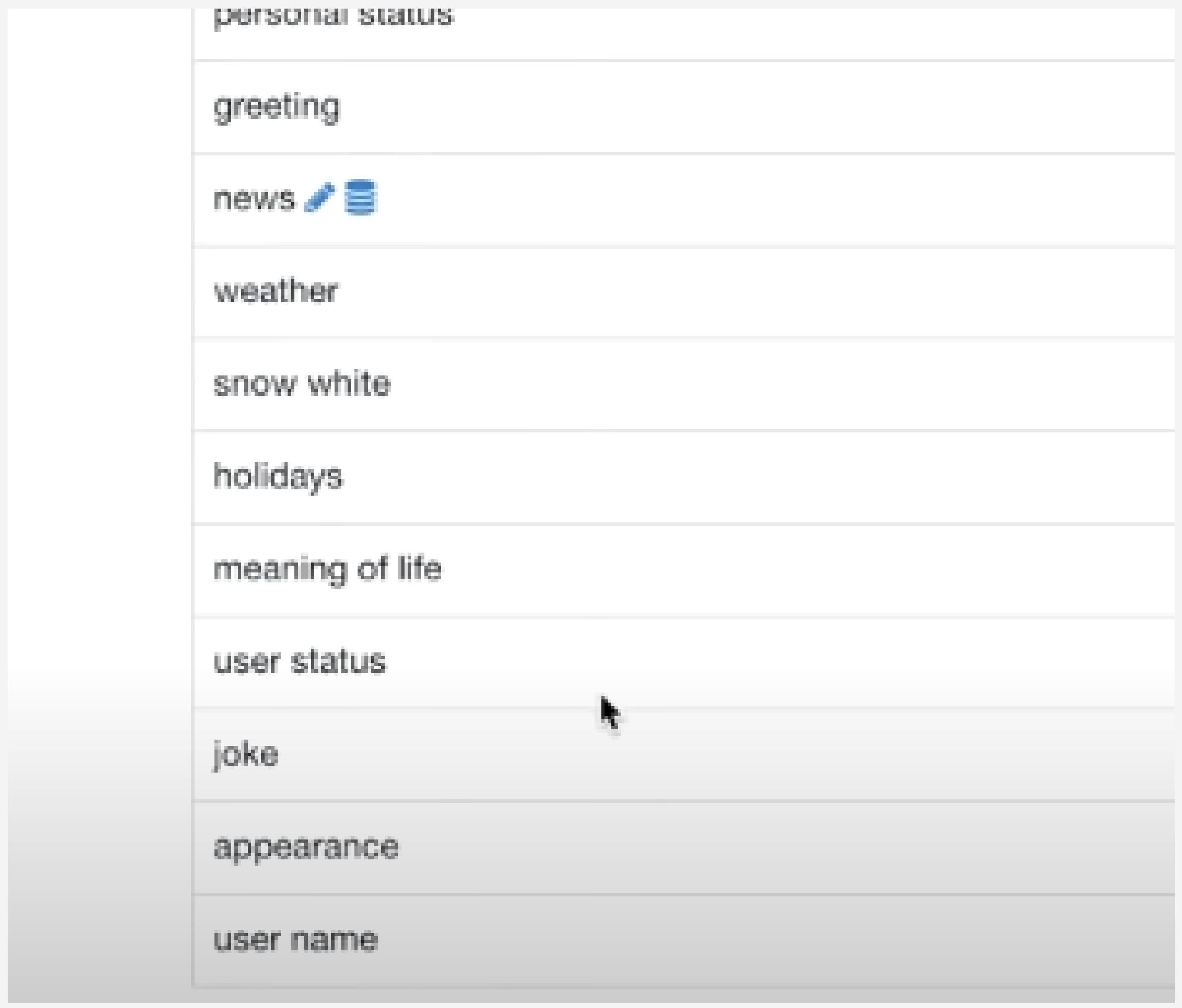


AI IMPLEMENTATION

MODELS:

1. FACIAL RECOGNITION
2. WIT TRAINING

For facial recognition we use OpenCV. For wit training we choose wit.ai where we can train our cases. Things that our py can understand ->



personal status
greeting
news
weather
snow white
holidays
meaning of life
user status
joke
appearance
user name

CODE:

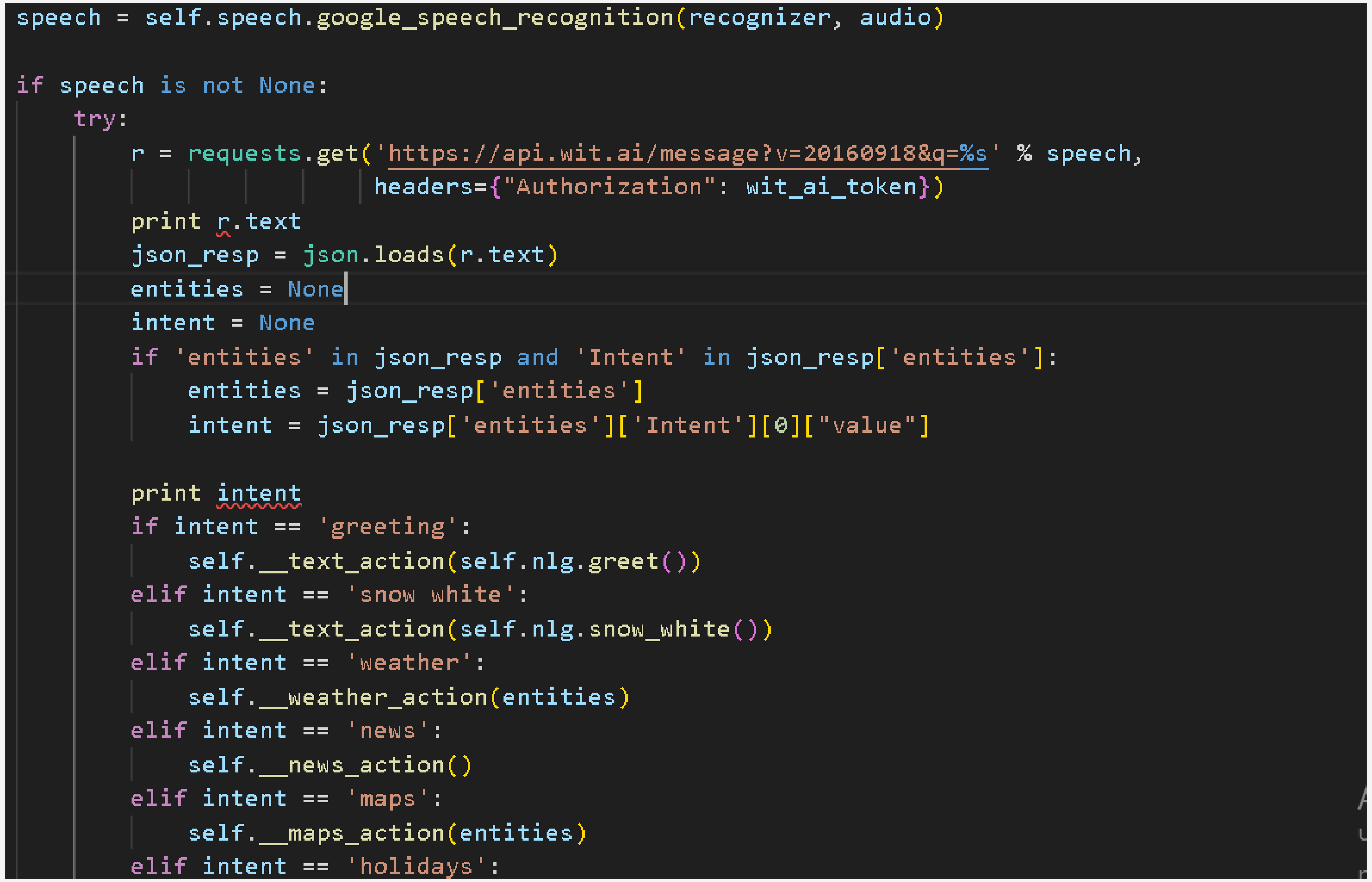
Our Python script represents the core logic of a conversational bot designed for a smart mirror. Here's a brief overview of its key components:

- Initialization:
  - Imports necessary modules and libraries, including those for speech recognition, natural language generation, knowledge retrieval, and vision.
- Bot Class:
  - Initializes various components such as NLG (Natural Language Generation), Speech, Knowledge, and Vision.
  - Implements the main loop, waiting for a launch phrase and then deciding on actions based on user input.
- Action Decisions:
  - Utilizes the Wit.ai API for natural language understanding (NLU) to determine user intent.
  - Decides on actions based on recognized intents, such as greetings, weather inquiries, news requests, and more.
- Weather Actions:
  - Retrieves weather information using a knowledge module and responds with synthesized text.
  - Handles different timeframes (current, hourly, daily) and specific weather types (e.g., today, tomorrow, 3-day forecast).
- News Actions:
  - Fetches news headlines and sends them to the smart mirror display.
  - Generates synthesized text for news-related responses.

This code segment performs the following tasks:

1. Speech Recognition:
   - Converts received audio data into text using Google's Speech Recognition API.
2. Natural Language Understanding (Wit.ai):
   - Sends the recognized speech text to Wit.ai for intent and entity extraction.
3. Intent-Based Actions:
   - Determines user intent from Wit.ai response.
   - Executes specific actions based on recognized intent, such as weather retrieval, news fetch, etc.
4. Default Response:
   - Provides a default response if no matching intent is found.

In essence, it interprets user speech, understands intent, and takes corresponding actions based on the recognized commands.



```
speech = self.speech.google_speech_recognition(recognizer, audio)

if speech is not None:
    try:
        r = requests.get('https://api.wit.ai/message?v=20160918&q=%s' % speech,
                         headers={"Authorization": wit_ai_token})
        print r.text
        json_resp = json.loads(r.text)
        entities = None
        intent = None
        if 'entities' in json_resp and 'Intent' in json_resp['entities']:
            entities = json_resp['entities']
            intent = json_resp['entities']['Intent'][0]["value"]

        print intent
        if intent == 'greeting':
            self.__text_action(self.nlg.greet())
        elif intent == 'snow white':
            self.__text_action(self.nlg.snow_white())
        elif intent == 'weather':
            self.__weather_action(entities)
        elif intent == 'news':
            self.__news_action()
        elif intent == 'maps':
            self.__maps_action(entities)
        elif intent == 'holidays':
```

CODE:

- Maps Actions:
  - Determines actions related to maps, such as displaying a map of a specified location.
  - Retrieves map URLs and sends them to the smart mirror display.
- Holidays Actions:
  - Retrieves information about holidays and displays the next upcoming holiday on the smart mirror.
  - Generates synthesized text for holiday-related responses.
- Speech Synthesis and Display:
  - Uses the Speech class to handle speech synthesis, sending synthesized text to the smart mirror for display.
  - Interacts with the smart mirror display through HTTP requests.
- Vision Recognition:
  - Uses the Vision module to recognize faces and trigger actions based on the presence of a face.
- User Interaction:

  Handles user interactions through speech recognition and responds accordingly.

This script demonstrates the integration of various components to create a smart mirror with conversational capabilities. The bot responds to voice commands, providing weather updates, news, maps, and other relevant information.

CONCLUSION

In summary, the "Magic Mirror with AI" project is a fusion of cutting-edge technologies that transforms a traditional mirror into a smart and intelligent interface. It not only reflects the user's image but also engages in interactive conversations, provides useful information, and offers a unique and futuristic user experience.