**AI Project - Intelligent Excuse Generator**

Project Overview: he AI team has been assigned the development of an Intelligent Excuse Generator, an AI-driven system designed to provide context-aware,highly customizable excuses for different scenarios, enhancing user credibility with automated reasoning and Context-based excuse suggestions (work, school, social, family). Allows users to refine excuses based on urgency and AI-generated documents, chat screenshots, and location logs to support Auto-triggers fake emergency messages or calls. Auto-creates professional or emotional apologies. Generates excuses in both written and speech format for added supportive proof generation.

Project Objectives:

• Develop an AI-powered excuse generator with contextual awareness.

• Enable users to generate proof-backed excuses for various situations.

• Integrate automated messaging and emergency alert systems.

• Ensure the excuses sound natural, believable, and adaptable.

Code of the above Project

import random

import datetime

from faker import Faker

import pyttsx3

import json

import os

import threading

import pygame

from gtts import gTTS

import tempfile

from PIL import Image, ImageDraw, ImageFont

fake = Faker()

excuse\_history = []

favorites = []

voice\_engine = pyttsx3.init()

pygame.mixer.init()

def load\_data():

    try:

        with open('excuse\_history.json', 'r') as f:

            global excuse\_history

            excuse\_history = json.load(f)

    except (FileNotFoundError, json.JSONDecodeError):

        excuse\_history = []

    try:

        with open('favorites.json', 'r') as f:

            global favorites

            favorites = json.load(f)

    except (FileNotFoundError, json.JSONDecodeError):

        favorites = []

def save\_data():

    with open('excuse\_history.json', 'w') as f:

        json.dump(excuse\_history, f)

    with open('favorites.json', 'w') as f:

        json.dump(favorites, f)

def generate\_excuse():

    excuses = {

        'work': {

            'low': [

                "Feeling kinda off today, need to rest",

                "Got a family thing I can’t move"

            ],

            'medium': [

                "Internet’s down, total chaos",

                "Forgot about a doctor’s appointment"

            ],

            'high': [

                "Family emergency, gotta run",

                "Tested positive for COVID, quarantining"

            ]

        },

        'school': {

            'low': [

                "Need more time to nail this assignment",

                "Got confused by the instructions"

            ],

            'medium': [

                "Computer died, lost my work",

                "Been dealing with some personal stuff"

            ],

            'high': [

                "Lost someone close, need time",

                "Having some serious health issues"

            ]

        },

        'social': {

            'low': [

                "Totally forgot we had plans",

                "Not really up for going out"

            ],

            'medium': [

                "Car’s in the shop, bad timing",

                "Work’s keeping me late"

            ],

            'high': [

                "Going through a rough patch, need space",

                "Was around someone with COVID"

            ]

        },

        'family': {

            'low': [

                "Got something else that day",

                "Feeling under the weather"

            ],

            'medium': [

                "Work schedule got crazy",

                "Car’s acting up, can’t drive"

            ],

            'high': [

                "Emergency came up, gotta handle it",

                "Dealing with some health problems"

            ]

        }

    }

    print("\nCategories: work, school, social, family")

    category = input("What’s it for? ").lower()

    print("Urgency levels: low, medium, high")

    urgency = input("How urgent is it? ").lower()

    if category in excuses and urgency in excuses[category]:

        excuse = random.choice(excuses[category][urgency])

        timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

        excuse\_data = {

            'excuse': excuse,

            'category': category,

            'urgency': urgency,

            'timestamp': timestamp

        }

        excuse\_history.append(excuse\_data)

        save\_data()

        print(f"\nYour excuse: {excuse}")

        return excuse

    else:

        print("Doh! Bad category or urgency. Try again.")

        return None

def generate\_proof():

    proof\_types = ["document", "chat screenshot", "location log", "image proof"]

    print("\nProof types:", ", ".join(proof\_types))

    proof\_type = input("What kind of proof do you need? ").lower()

    if proof\_type == 'document':

        proof = f"Generated Document: {fake.file\_name(extension='pdf')}"

    elif proof\_type == 'chat screenshot':

        proof = f"Chat Screenshot: {fake.file\_name(extension='png')}"

    elif proof\_type == 'location log':

        proof = f"Location Log: {fake.address()}"

    elif proof\_type == 'image proof':

        proof = "Image Proof: Generated (saved as proof\_image.png)"

        generate\_proof\_image()

    else:

        proof = "No proof generated, pick a valid type next time!"

    print(proof)

def generate\_proof\_image():

    try:

        img = Image.new('RGB', (400, 300), color=(255, 255, 255))

        d = ImageDraw.Draw(img)

        try:

            font = ImageFont.truetype("arial.ttf", 14)

        except:

            font = ImageFont.load\_default()  # Fallback, not ideal but works

        d.rectangle([50, 50, 350, 250], outline=(9, 132, 227), width=2)

        d.text((60, 60), "Official Proof Document", fill=(45, 52, 54), font=font)

        d.text((60, 90), f"Date: {datetime.date.today()}", fill=(45, 52, 54), font=font)

        d.text((60, 120), "Status: Verified", fill=(9, 132, 227), font=font)

        d.text((60, 150), f"Ref: {fake.uuid4()}", fill=(45, 52, 54), font=font)

        d.text((60, 180), "Signed: Dr. Smith", fill=(45, 52, 54), font=font)

        img.save('proof\_image.png')  # Save it since no UI

        print("Image saved as proof\_image.png")

    except ImportError:

        print("Bummer, need Pillow for images. Install it!")

def generate\_emergency():

    emergencies = [

        "Family crisis, gotta go now",

        "Medical emergency, heading to hospital",

        "House flooded, dealing with it",

        "Car crash, sorting it out"

    ]

    emergency = random.choice(emergencies)

    print(f"\nEmergency message: {emergency} (Sent to whoever needs to know)")

def generate\_apology():

    style = input("\nWant a professional apology? (yes/no): ").lower()

    if style == 'yes':

        apology = "So sorry for the inconvenience. I value our relationship and I’ll make sure this doesn’t happen again."

    else:

        apology = "I’m super sorry, I messed up big time! Please forgive me, I’ll make it right."

    print(f"\nApology: {apology}")

def speak\_excuse(excuse):

    if not excuse:

        print("Nothing to say, generate an excuse first!")

        return

    voice = input("\nUse Google TTS? (yes/no, no means system voice): ").lower()

    if voice == 'yes':

        speak\_with\_gtts(excuse)

    else:

        speak\_with\_pyttsx3(excuse)

def speak\_with\_pyttsx3(text):

    def \_speak():

        voice\_engine.say(text)

        voice\_engine.runAndWait()

    threading.Thread(target=\_speak, daemon=True).start()

def speak\_with\_gtts(text):

    try:

        with tempfile.NamedTemporaryFile(suffix='.mp3', delete=False) as f:

            temp\_path = f.name

        tts = gTTS(text=text, lang='en')

        tts.save(temp\_path)

        pygame.mixer.music.load(temp\_path)

        pygame.mixer.music.play()

        while pygame.mixer.music.get\_busy():

            pygame.time.Clock().tick(10)

        os.unlink(temp\_path)

    except Exception as e:

        print(f"Speech failed: {str(e)}")

def add\_to\_favorites(excuse):

    if not excuse:

        print("No excuse to save!")

        return

    for item in excuse\_history:

        if item['excuse'] == excuse:

            if item not in favorites:

                favorites.append(item)

                save\_data()

                print("Added to favorites!")

            else:

                print("Already in favorites!")

            return

    print("Can’t find that excuse in history.")

def show\_history():

    if not excuse\_history:

        print("\nNo excuses yet!")

        return

    print("\nYour Excuse History:")

    for i, item in enumerate(excuse\_history):

        print(f"{i+1}. {item['timestamp']} | {item['category'].capitalize()} | {item['urgency'].capitalize()} | {item['excuse']}")

    choice = input("\nEnter number to speak or favorite (0 to skip): ")

    if choice.isdigit() and 0 < int(choice) <= len(excuse\_history):

        item = excuse\_history[int(choice) - 1]

        action = input("Speak or favorite? (s/f): ").lower()

        if action == 's':

            speak\_excuse(item['excuse'])

        elif action == 'f':

            add\_to\_favorites(item['excuse'])

def show\_favorites():

    if not favorites:

        print("\nNo favorites yet!")

        return

    print("\nYour Favorite Excuses:")

    for i, item in enumerate(favorites):

        print(f"{i+1}. {item['timestamp']} | {item['category'].capitalize()} | {item['urgency'].capitalize()} | {item['excuse']}")

    choice = input("\nEnter number to speak or remove (0 to skip): ")

    if choice.isdigit() and 0 < int(choice) <= len(favorites):

        item = favorites[int(choice) - 1]

        action = input("Speak or remove? (s/r): ").lower()

        if action == 's':

            speak\_excuse(item['excuse'])

        elif action == 'r':

            favorites.remove(item)

            save\_data()

            print("Removed from favorites!")

def get\_prediction():

    if len(excuse\_history) > 3:

        print("\nYo, my gut says you’ll need an excuse tomorrow afternoon. You’ve been slacking a lot!")

    else:

        print("\nNot enough excuses to predict yet, keep at it!")

def generate\_multilang():

    langs = ["english", "spanish", "french", "german"]

    print("\nLanguages:", ", ".join(langs))

    lang = input("Pick a language: ").lower()

    excuses = {

        'english': "Sorry, can’t make it today",

        'spanish': "Lo siento, hoy no puedo",

        'french': "Désolé, je ne peux pas venir aujourd’hui",

        'german': "Tut mir leid, ich kann heute nicht"

    }

    excuse = excuses.get(lang, "No clue how to say it in that language")

    print(f"\nExcuse: {excuse}")

def get\_ranking():

    if not excuse\_history:

        print("\nNo excuses to rank yet!")

        return

    ranked = sorted(excuse\_history,

                    key=lambda x: 3 if x['urgency'] == 'high' else

                                 2 if x['urgency'] == 'medium' else 1,

                    reverse=True)

    print("\nYour Best Excuses:")

    for i, item in enumerate(ranked[:3]):

        print(f"{i+1}. {item['excuse']} (Vibe: {item['urgency'].capitalize()})")

def main():

    load\_data()

    while True:

        print("\n Excuse Generator ")

        print("1. Generate Excuse")

        print("2. Generate Proof")

        print("3. Emergency Message")

        print("4. Apology")

        print("5. Show History")

        print("6. Show Favorites")

        print("7. Predict Next Excuse")

        print("8. Multi-language Excuse")

        print("9. Rank Excuses")

        print("0. Quit")

        choice = input("What is your choice: ")

        if choice == '1':

            excuse = generate\_excuse()

            if excuse:

                speak = input("Speak this excuse? (yes/no): ").lower()

                if speak == 'yes':

                    speak\_excuse(excuse)

                fav = input("Add to favorites? (yes/no): ").lower()

                if fav == 'yes':

                    add\_to\_favorites(excuse)

        elif choice == '2':

            generate\_proof()

        elif choice == '3':

            generate\_emergency()

        elif choice == '4':

            generate\_apology()

        elif choice == '5':

            show\_history()

        elif choice == '6':

            show\_favorites()

        elif choice == '7':

            get\_prediction()

        elif choice == '8':

            generate\_multilang()

        elif choice == '9':

            get\_ranking()

        elif choice == '0':

            print("Peace out!")

            break

        else:

            print("Pick a valid option, dude!")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**OUTPUT**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 1**

**Categories: work, school, social, family**

**What’s it for? work**

**Urgency levels: low, medium, high**

**How urgent is it? medium**

**Your excuse: Forgot about a doctor’s appointment**

**Speak this excuse? (yes/no): yes**

**Use Google TTS? (yes/no, no means system voice): yes**

**Speech failed: [WinError 32] The process cannot access the file because it is being used by another process: 'C:\\Users\\nazee\\AppData\\Local\\Temp\\tmpdkx3ugnz.mp3'**

**Add to favorites? (yes/no): yes**

**Added to favorites!**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 2**

**Proof types: document, chat screenshot, location log, image proof**

**What kind of proof do you need? document**

**Generated Document: certain.pdf**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 3**

**Emergency message: Family crisis, gotta go now (Sent to whoever needs to know)**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 4**

**Want a professional apology? (yes/no): yes**

**Apology: So sorry for the inconvenience. I value our relationship and I’ll make sure this doesn’t happen again.**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 5**

**Your Excuse History:**

**1. 2025-07-03 19:10:09 | Work | High | I've tested positive for COVID and need to isolate**

**2. 2025-07-03 19:10:31 | School | High | I'm experiencing severe health issues**

**3. 2025-07-03 19:26:02 | Work | Medium | I have a doctor's appointment I forgot about**

**4. 2025-07-03 19:26:52 | School | High | I'm experiencing severe health issues**

**5. 2025-07-03 23:18:08 | Work | High | I've tested positive for COVID and need to isolate**

**6. 2025-07-03 23:21:34 | Work | Medium | I have a doctor's appointment I forgot about**

**7. 2025-07-04 07:40:21 | Work | Medium | I have a doctor's appointment I forgot about**

**8. 2025-07-04 07:40:21 | Work | Medium | My internet connection is down unexpectedly**

**9. 2025-07-04 09:09:21 | Work | Medium | My internet connection is down unexpectedly**

**10. 2025-07-04 09:10:54 | School | High | I'm experiencing severe health issues**

**11. 2025-07-04 10:49:48 | Work | Medium | My internet connection is down unexpectedly**

**12. 2025-07-04 10:50:30 | Work | High | There's been a family emergency I need to attend to**

**13. 2025-07-04 22:21:20 | Work | Medium | I have a doctor's appointment I forgot about**

**14. 2025-07-04 22:37:41 | Work | Medium | I have a doctor's appointment I forgot about**

**15. 2025-07-04 23:01:26 | Work | Medium | My internet connection is down unexpectedly**

**16. 2025-07-05 09:35:10 | Work | Medium | My internet connection is down unexpectedly**

**17. 2025-07-05 09:48:09 | Work | Medium | I have a doctor's appointment I forgot about**

**18. 2025-07-05 09:55:12 | Work | Medium | Internet’s down, total chaos**

**19. 2025-07-05 10:30:50 | Work | Medium | Forgot about a doctor’s appointment**

**20. 2025-07-05 11:07:44 | Work | Low | Feeling kinda off today, need to rest**

**21. 2025-07-05 11:10:36 | Work | Low | Feeling kinda off today, need to rest**

**22. 2025-07-05 11:19:15 | Work | Low | Feeling kinda off today, need to rest**

**23. 2025-07-05 11:38:44 | Work | Medium | Internet’s down, total chaos**

**24. 2025-07-05 11:39:21 | School | High | Lost someone close, need time**

**25. 2025-07-05 11:39:45 | Social | Low | Totally forgot we had plans**

**26. 2025-07-05 11:40:30 | Family | High | Dealing with some health problems**

**27. 2025-07-05 11:43:13 | Work | Medium | Forgot about a doctor’s appointment**

**Enter number to speak or favorite (0 to skip): 2**

**Speak or favorite? (s/f): s**

**Use Google TTS? (yes/no, no means system voice): yes**

**Speech failed: [WinError 32] The process cannot access the file because it is being used by another process: 'C:\\Users\\nazee\\AppData\\Local\\Temp\\tmp3vtqwbsu.mp3'**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 6**

**Your Favorite Excuses:**

**1. 2025-07-05 11:07:44 | Work | Low | Feeling kinda off today, need to rest**

**2. 2025-07-05 09:55:12 | Work | Medium | Internet’s down, total chaos**

**3. 2025-07-05 11:39:21 | School | High | Lost someone close, need time**

**4. 2025-07-05 10:30:50 | Work | Medium | Forgot about a doctor’s appointment**

**Enter number to speak or remove (0 to skip): skip**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 7**

**Yo, my gut says you’ll need an excuse tomorrow afternoon. You’ve been slacking a lot!**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 8**

**Languages: english, spanish, french, german**

**Pick a language: spanish**

**Excuse: Lo siento, hoy no puedo**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 9**

**Your Best Excuses:**

**1. I've tested positive for COVID and need to isolate (Vibe: High)**

**2. I'm experiencing severe health issues (Vibe: High)**

**3. I'm experiencing severe health issues (Vibe: High)**

**Excuse Generator**

**1. Generate Excuse**

**2. Generate Proof**

**3. Emergency Message**

**4. Apology**

**5. Show History**

**6. Show Favorites**

**7. Predict Next Excuse**

**8. Multi-language Excuse**

**9. Rank Excuses**

**0. Quit**

**What is your choice: 0**

**Peace out!**