

# **Configuration Manual**

MSc Research Project Artificial Intelligence for Business

> Disha Agrawal Student ID: 22212159

School of Computing National College of Ireland

Supervisor:

Victor del Rosal

### National College of Ireland

### **MSc Project Submission Sheet**



### **School of Computing**

Student Name:	Disha Agrawal		
Student ID:	22212159		
Programme:	M.Sc. AI for Business	Year:	
Module:	MSc Research Practicum/Internship part	t 2	
Lecturer: Submission Due Date:			
Project Title	Evoluting Advanced AI Canabilities for Dro	viact Ma	nagement

Project Title: Exploring Advanced AI Capabilities for Project Management

Word Count: ......9.

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

Dicha Agra

Date:

.....15/09/2024.....

.....

### PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple	
copies)	
Attach a Moodle submission receipt of the online project	
submission, to each project (including multiple copies).	
You must ensure that you retain a HARD COPY of the project, both	
for your own reference and in case a project is lost or mislaid. It is not	
sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

# **Configuration Manual**

Disha Agrawal Student ID: 22212159

# 1 Requirements guide

This manual lists down all the steps in detail that must be followed in order to replicate the development of Project Management Virtual Assistant chatbot, ProManage. This document will discuss both the hardware and software requirements for the system to run successfully.

### 1.1 Hardware requirements

The chatbot is developed using a cloud server via Google Colaboratory. However, to replicate and run the code locally, system would require 12.7 GB RAM and 107 GB Disk space.

### **1.2 Software requirements**

Operating system: Windows 10 or 11, Linux, or macOS Python environment setup: Python 3.7 or higher. Packages and Libraries-OpenAI API to facilitate interaction with the LLM model (GPT-3.5) Gradio: For building the frontend of chatbot Pandas and Numpy: for data modelling Json: For reading Json project data files Google Colab Notebook: For writing the code and testing the chatbot HuggingFace Spaces: For hosting the chatbot interface Carrd.co: To design the webpage on which chatbot will be linked Google Drive: For storing all the required files, datasets, code, and more

### **1.3** Network requirements

Stable internet with a minimum speed of 10Mbps.

## 2 Development Methodology



## 3 Acquiring API key

Step 1: Visit the OpenAI website (<u>https://www.openai.com/</u>) and sign up for an account if you don't already have, or login with an existing account.

Step 2: After account verification, navigate to the API dashboard and click on "API keys" and click on "Create New secret key". If you are not able to generate any key, that means that you need to add balance to your account. Navigate to the profile icon on the top right corner of your screen, and go to Your Profile and then Billing. Add payment details and purchase credits (minimum amount \$5).

Step 3: Navigate back to the API dashboard and create a new secret key. This will be then added to your dashboard. Copy and save the key for later use.

P Personal 0 / Default pro	oject ≎				Playground	Dash
DASHBOARD	API keys					
🙁 Assistants						
😴 Fine-tuning	(i) Project API keys have replaced user API keys.					
{≡} Batches	We recommend using project based API keys for mo	re granular control over you	ur resources. Learn more			
Storage						
d) Usage	As an owner of this project, you can view and manage all API	keys in this project.				
API keys	Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may al leaked publicly.					also au
	View usage per API key on the Usage page.					
	NAME	SECRET KEY	CREATED	LAST USED ()	CREA	TED BY
	Secret key	skqL4q	22 Jun 2024	11 Aug 2024	Disha	Agraw



### **4** Dataset generation

Step 1: Navigate to Google Colab notebooks (<u>https://colab.research.google.com/</u>).

Step 2: Create an account or use an existing account for login. Then click on new notebook.

Step 3: Consider a hypothetical project. In this instance, we take Website redesign project. Step 4: Import Json library and define the dataset. Create a dictionary that shows project details, such as project overview, task timelines, team members, Budget information, client,

and so on.

Step 5: Populate the Python dictionary with random values and click on run.

Step 6: Save the dataset as a Json file in your drive and name the folder by the project name.

Step 7: Repeat the above steps for 3 other projects, namely, Marketing Campaign, Mobile App Development and Software Development.

0	import json
	# Define the dataset
	data = {
	"project name": "Website Redesign",
	"project manager": "Alice Johnson".
	"start date": "2024-06-01".
	"end date": "2024-12-31".
	"tacks" (
	("task id" 1 "task name", "Tnitial Planning" "status", "completed" "assigned to", "Alice Johnson" "due date", "2024-06-10" "risk level
	["task id": 2"task name": "Denuirament (athening" "status": "complatad" "ascigned to": "Deb Smith" "due data": "Denuirament (athening" "status": "complatad" "accigned to": "Deb Smith" "due data": "Denuirament (athening") "status": "complatad" "accigned to": "Deb Smith" "due data": "Denuirament (athening) "status": "complatad" "accigned to": "Deb Smith" "due data": "Def Smithering" "status": "complatad" "accigned to": "Deb Smithering": "def Smithering": "status": "complatad" "accigned to": "Deb Smithering": "def Smithering": "status": "complatad" "accigned to": "Deb Smithering: "def Smithering": "status": "complatad" "accigned to": "Deb Smithering: "def Smithering: "status": "status: "status": "status: "sta
	( tash_id*, 2) tash_indire . Requirement decine ing ) steadus . Compared ) assignet_(0, 000 smith) ude_date . 2024-00 zeto ) restance ( f"task id*, 2) "task norma". "Doeing Machune" "tetatus". "in prograss" "assigned tast. "Corol White" "due date. "2024-00 zeto" "in a law id "
	("task_id", 4, "task_nome", "Douglon Encound", "tastus", "popular", "popular", "popular", "douglond Encound", "task_nome", "popular Encound", "task_nome", "popular Encound", "task_nome", "popular, "tastus", "popular", "popular", "popular, "task_nome", "t
	("task_id", t, task_indime. Develop Hontend", "status , pending", assigne_ut. Dave brown, ude_ute. 2024-06-20, "risk_level", "
	{ task_idi; 5, task_hame; beverup backenu, status; pending, assigned_to; eve green, uue_date; 2024-08-30, risk_lever; http://doc.org///like/abs/signed/abs/signe
	{ task_id: 6, task_name : integration, status : pending, assigned to : Frank Blue , due_date : 2024-09-15, risk_level : medium
	task id : /, task name : Testing, status : pending, assigned_to : Grace Black , due date : 2024-10-30 , risk_level : medium ,
	{"task_id": 8, 'task_name": 'Launch", 'status": "pending", 'assigned_to': 'Alice Jonnson', 'due_date": '2024-11-15', 'risk_level': 'low', 'r
	{"task_id": 9, "task_name": "Post-launch Support", "status": "pending", "assigned_to": "Bob Smith", "due_date": "2024-12-31", "risk_level":
	μ · · · · · · · · · · · · · · ·
	"resources": [
	{"name": "Alice Johnson", "role": "Project Manager", "allocated_hours": 40, "workload": "moderate", "tasks": [1, 8]},
	{"name": "Bob Smith", "role": "Business Analyst", "allocated_hours": 70, "workload": "high", "tasks": [2, 9]},
	{"name", "Carol White" "role", "Designer" "allocated hours", 50 "workload", "moderate" "tasks", [3]}

Fig 3: Dataset generation

# 5 Writing the code

Step 1: Navigate to Google Colab notebooks (<u>https://colab.research.google.com/</u>).

Step 2: Create an account or use an existing account for login. Then click on new notebook. Step 3: Mount your Google Drive to access files and datasets stored there.



Fig 4: Drive Mount

Step 4: Import Json, Install OpenAI, and import OpenAI.



Fig 5: Installing necessary libraries in Python

Step 5: Load the Json file you saved earlier in the drive. Step 6: Load your API key (Generated with OpenAI).



Step 6: Define a function for the chatbot to work with user queries.



Fig 7: Function creation

Step 7: Setup Gradio library for building the chatbot interface.



Step 8: Run all the cells to execute the program. Click on the URL generated to interact with the chatbot.

### 6 Hosting the Server on HuggingFace Spaces

Step 1: Navigate to <u>https://huggingface.co/</u>. Set up your account and click on "New", and then on Space, to create a new space.

+ New	Following 0 ~		🗹 Trending 🛛 last 7 days
Model Upload a model and get an API endpoint.	Aodels Datasets Spaces Papers Collections s Likes	Community Posts	All Models Datasets Space
<b>Dataset</b> Easily host your data.	Follow your favorite AI creators	© <u>Refresh List</u> ×	& black-forest-labs/FLUX Text-to-Image • Updated 2 d
Space Create an interactive ML demo.	ı <b>ri-pl</b> · Integrate AI in game dev	Follow	& black-forest-labs/FLUX
Collection	deCopet · Research on Music generation	Follow	V Text-to-Image • Updated 2 da
Organizations	Ū		FLUX,1[Sch
+ Create New			
Resources			FLUX.1[de
Hub guide			Standard (Minic CDM V 2 (
Transformers doc			Openbilib/HINICPH-V-2_6      Text Generation + Undated 1 di
@ Forum			o lext deneration - oputted 1 di
A Tasks			
	Fig 9: HuggingFace dashboard		

Step 2: Fill all details on the next page and select the Free plan (CPU Basic) to start with. Select Gradio as the SDK, and chatbot as template. Finally, click on "Create Space".

paces   🕒 Dishaa/hello 🖆 🔯 like 0 🕒 Running 📑	🖗 Арр	)≣ File
👳 Chatbot		
ting started with your Gradio Space.		
ting started with your Gradio Space. × r Space is online and ready to be updated. rt by cloning this Space repository using git:		
tting started with your Gradio Space.       ×         r Space is online and ready to be updated.       ×         rt by cloning this Space repository using git:       •         HTTPS       SSH       •         Use an access token as git password/credential       •		
ting started with your Gradio Space.       ×         r Space is online and ready to be updated.       *         rt by cloning this Space repository using git:       •         HTTPS       SSH       •         Use an access token as git password/credential       •         When prompted for a password, use an access token with related to the form your settings: https://huggingface.co/settingtit clone https://huggingface.co/space/Dishaa/hello		

Fig 10: Chatbot interface (Gradio)

Step 2: Navigate to "Files" in the top tool bar, just beside "App". You will see 2 pre-existing files named .gitattributes and README.md. Create a new file named app.py and enter the following code.

```
raw 📋 Copy download link 🙂 history 🙂 blame 🖉 edit 🔟 delete 🤍 No virus
    import subprocess
     import sys
     def install(package):
        subprocess.check_call([sys.executable, "-m", "pip", "install", package])
     install("openai==0.28")
 9 import json
     from pathlib import Path
    import gradio as gr
12 import os
13 import shutil
     import json
15 import openai
    api_key = os.getenv("api_key")
    ♯ Load the OpenAI API key
    openai.api_key = api_key
     project_data = {}
     def ask_chatbot(query, chat_history, project_data):
        messages.append({"role": "user", "content": message})
        response = ""
        for message in client.chat_completion(
           messages,
            max_tokens=max_tokens,
           stream=True,
            temperature=temperature,
            top_p=top_p,
        ):
            token = message.choices[0].delta.content
            response += token
            yield response
    .....
   For information on how to customize the ChatInterface, peruse the gradio docs: https://www.gradio.app/docs/o
    .....
    demo = gr.ChatInterface(
       respond.
        additional_inputs=[
            gr.Textbox(value="You are a friendly Chatbot.", label="System message"),
            gr.Slider(minimum=1, maximum=2048, value=512, step=1, label="Max new tokens"),
           gr.Slider(minimum=0.1, maximum=4.0, value=0.7, step=0.1, label="Temperature"),
           gr.Slider(
                minimum_0 1
   if __name__ == "__main__":
       demo.launch()
```

#### Fig 11: Code for app.py

### 7 Creating Webpage

Step 1: Navigate to <u>https://carrd.co/</u> and create an account or use an existing account to login. Step 2: Click on 'Choose a Starting Point" and select a theme for your website. Chose from any of the free templates available. Step 3: Customize the template according to the project's requirements. Add relevant content and button "Get Started" to link the chatbot. Add the link obtained from HuggingFace into the URL <u>https://huggingface.co/spaces/Dishaa/project\_manager</u>.



Fig 12: Carrd.co Dashboard



Fig 13: Design Template

Step 3: Click on "Done" and then "Publish Site".



ProManage understands your unique project needs by processing your emails, chats, and meetings. It



# 8 Conclusion

By following the above-mentioned steps, one can recreate the chatbot ProManage.