

# **APPLICABILITY OF AN AI CHAT ASSISTANT IN RESOLVING KNOWLEDGE COMPARTMENTALIZATION WITHIN A WORKPLACE**

---

By Arkadiusz Kulpa

## **EXECUTIVE SUMMARY**

---

This report aims to investigate whether AI Innovations can help manage complex operations of a company suffering from compartmentalized knowledge, where various teams need to rely on the broader business for information and guidance which is hard to come by individual specialists. An efficient and scalable cloud LLM application is proposed, where staff could query and receive responses that are grounded in facts derived from company's data to which the GPT model has access.

Such innovations will necessarily be applied during the company's ongoing transformation, part of which is to utilise AWS Cloud services and so the artifact developed in the second part of this report showcases a scalable AWS LightSail Instance providing a cost-effective VPS for hosting application's WordPress front-end and PHP Back-end server. AWS API Gateways communicate with AWS Lambda serverless function which stores the OpenAI API Keys and communicates with the model, response fetched is then sent back to the front-end showing the entire end-to-end process required to host a secure LMM app over Cloud Technologies.

Data Security and Privacy is considered should OpenAI become a third-party data processor for the company and a Document store, LangChain, is outlined as a framework that could facilitate tokenization of company data and storage in a Vector Store, so that the query sent to OpenAI API contains relevant snippets of documents in the form of word embeddings and a simple demonstration of Chat GPT 4.0 Document reading capability is shown as an example of typical Use Case within the company.

This project demonstrates that LMMs work very well with cloud technologies and can be implemented with ease, providing access to cutting-edge technologies to any employee regardless of their technological acumen.

## TABLE OF CONTENTS

---

EXECUTIVE SUMMARY .....	1
TABLE OF CONTENTS .....	3
LIST OF FIGURES .....	5
LIST OF TABLES.....	6
GLOSSARY .....	7
Applicability of an AI Chat Assistant in resolving knowledge compartmentalization within a workplace.....	1
1 Evaluation of AI Innovation .....	9
1.1 Project Summary .....	9
1.2 Project Execution and Management .....	11
1.3 Applied Concepts.....	13
1.4 Reflection.....	14
2 THE DESIGN OF THE INFRASTRUCTURE .....	15
2.1 Recommendation of an Innovation.....	15
2.2 Technical Approach .....	16
2.3 Artifact Design and Development.....	17
2.3.1 AWS Security and Front-end Setup.....	18

2.3.2	Local Development of AWS Lambda .....	19
2.3.3	Front-End HTML Injection Block .....	20
2.3.4	Front-end JavaScript Script .....	22
2.3.5	Back-end PHP function and AWS API Gateways .....	23
2.3.6	OpenAI API settings .....	25
2.3.7	AWS Lambda function code (Python) .....	25
2.3.8	LangChain Implementation .....	26
2.3.9	Testing and Evaluation .....	27
2.3.10	Summary of Business applications .....	29
3	References .....	32
4	APPENDIX A APRENTICESHIP STANDARDS .....	33
5	APPENDIX B DATA Sample .....	34
5.1	The Artifact can be interacted with at: .....	34
5.2	Test Use Case .....	34
5.3	Test Real Data Document – CV .....	36

## LIST OF FIGURES

Figure 1- Fishbone diagram, identifying the problem.....	10
Figure 2 - LangChain framework diagram .....	16
Figure 3- LangChain LLM app Model.....	18
Figure 4- Front-end user interface for a Chat GPT 3.5 app .....	19
Figure 5 - GitHub Copilot coding help in local IDE .....	20
Figure 6 - SAM CLI Errors.....	20
Figure 7 - User Experience with and without CSS.....	21
Figure 8 - Basic HTML to form the structure of the chat bot .....	21
Figure 9 - CSS Style .....	22
Figure 10 - Adding user's message to chatHistory element.....	22
Figure 11 - Displaying User's Message in ChatBox.....	22
Figure 12 - HTML,CSS and JS Front-end Setup - POST calls .....	23
Figure 13 - PHP Back-end function that calls the AWS API Gateway .....	24
Figure 14- AWS Lambda UI .....	24
Figure 15- API Gateway integration .....	24
Figure 16 - OpenAI API use .....	25
Figure 17 - AWS Lambda calling OpenAI API.....	26
Figure 18 - API Gateway Response.....	26
Figure 19 - LangChain Document Loader, Vector Store and LLM Query implementation .....	27
Figure 20 - Context awareness of the model, replying to 'it' referencing previous message .....	27
Figure 21 - Setting of an 'evil' system message and the effects on a basic test prompt. ....	28

Figure 22 - Setting a sarcastic system message and testing a basic prompt..... 29

Figure 23 - Ability to set System Message in the UI ..... 29

Figure 24- Document loading within Chat GPT 4 OpenAI UI..... 30

Figure 25 - Performing a task based on referenced document ..... 30

Figure 26 - List all Jobs - Model listed all jobs but omitted the company of the latest jobs it wasn't directly mentioned in the paragraph ..... 37

Figure 27 - List Tech Stack - Model responded accurately ..... 38

Figure 28 - Model is able to list all jobs ..... 39

Figure 29 - Model hallucinations test – passed ..... 40

Figure 30 - Model is able to retrieve information and reformulate it in a coherent way ..... 40

Figure 31 - Model retrieving specific Certification IDs – Passed..... 41

Figure 32 - Model's analysis of data - Passed..... 41

## LIST OF TABLES

---

**No table of figures entries found.**

## GLOSSARY

---

EQ (Company) – name of the company

M&A – Merger and Acquisition project, where a company acquires another asset and merges its functionality into its own infrastructure.

New System US – bespoke software solution used by Company to manage the Share dealing side of the business, specifically in the US. It is being developed based on the New System UK version.

Package – a subcategory of functionality offered by the company within the New System-US software.

Corporate Actions – an example of a package, this one deals with transferring shares or selling shares of shareholders when a company they hold shares of undergoes a corporate action such as a merger, etc.

CR – Change Request. When a piece of functionality being developed uncovers a need to develop another piece of functionality that wasn't previously considered. A new Requirement created off of the old while mid-project is usually called a CR.

Scope Creep – When the scope of the project, or the number of functionalities / requirements needed for a particular release / project keeps expanding and growing, as business understanding of the project develops. While some scope creep is to be

expected, excessive scope creep is indicative of insufficient pre-project analysis / design of architecture and/or collaboration between various teams within the business.

DPA – Data Processing Addendum

BAA – Business Associate Agreement

AWS – Amazon Web Services

ECR – Elastic Container Registry used for uploading a container (such as a Docker image) to run applications on the image instead of AWS own environment



## 1 EVALUATION OF AI INNOVATION

---

The advent of AI in 2023 caught many organizations off guard. After years of technological neglect and lack of direction, archaic approaches are still used to manage often even the biggest market players. In Company's case two parallel projects aim to innovate:

While M&A project called VEGA, aims to tap into the US market and modernize EQs platform New System, paving the way for future acquisitions and helping realize the vision to become a 'leading global share registrar...', the transformation project aims to innovate EQs product offering as well as look at how AI can be a part of our future. First part of this project considers the AI angle, what opportunities have been missed and how we can improve and adapt in the future.

### 1.1 PROJECT SUMMARY

The transformation is being funded by EQ's owner - an investment company seeking to profitably sell Company in a few years. It will therefore be vital to demonstrate to the prospective buyers not only EQs current market capitalization and business processes, but its ability to become truly global. Hence a definitive factor driving success of the transformation is the M&A project, where dismantling of the legacy system was deemed as project's biggest liability and priority. Yet even more important is for Company to learn to replicate the process with greater efficiency in the next M&A project.

With the advent of OpenAI's Chat GPT the company was forced to consider both the threat and the opportunity that is AI (Tarasiuk & Czapski, 2023). Despite the blanket AI ban currently in place this project argues that it may be worthwhile to invest into AI Innovations now, especially for a project of such high complexity, where multiple teams

of specialists need to collaborate across various fintech disciplines. EQ's own Chat GPT based assistant that would answer queries about company's ever growing knowledge base (Marr, 2021, p. 22) would be an ideal boost to the project's productivity.

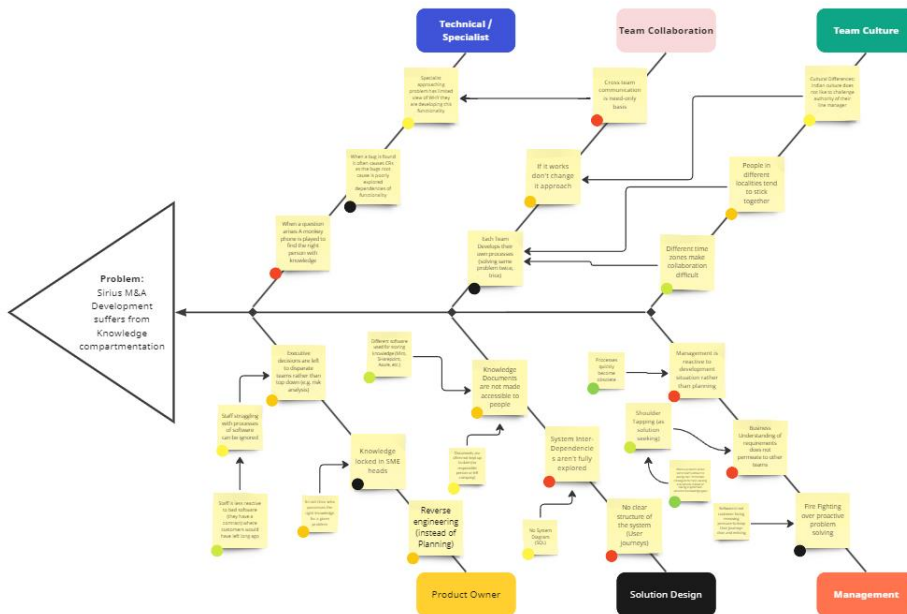


Figure 1- Fishbone diagram, identifying the problem

Knowledge compartmentalization seems to be the root problem within Project VEGA (Kulpa, Innovation Assessment Diagram, 2023). The knowledge is there, but is either locked in various team's SharePoint, in their local computer or worse yet in a team member's 'years of experience'.

Furthermore, there seems to exist no unified global data strategy for the project or the company at large, that would permeate to the level of an individual specialist. Each team is firefighting their specific problems using their bespoke solutions, often reinventing the wheel. Multiple CRs and constant scope creep is indicative of poor

architecture, design and analysis of the original requirement and its inter-dependencies.

For example, a specialist spends hours on calls, investigates lengthy Use Case documents, and is given links to key resources only when he asks the right question to the right people. Instead, a few prompts to EQ's own Chat Assistant with access to company's data could help retrieve a general overview of an area under investigation and a detailed response to a specific question, as well as links to relevant resources.

Such an innovation will not change company's culture, which is currently not very receptive to change, innovation or collaboration, but it will circumvent the problem and enable everyone to reach the relevant resource in a matter of minutes. While some may worry it might soon be simpler to have AI run the 'entire' project (Marr, 2021, p. 125), it is worth remembering that FinTech projects will most likely fall under the 'augmentation' type of human jobs affected by AI, especially since financial transactions will not soon be entrusted to AI.

## **1.2 PROJECT EXECUTION AND MANAGEMENT**

One of Transformation Project's new Data Office team's responsibilities will be to consider how to enable Company to use LLMs in a secure way. As more of our data that is currently held on-premises is trusted to a Cloud service such as AWS, EQ's data security will be defined by Amazon's state-of-the-art security, permission management system IAM, security of data in transit and at rest, choice of AWS regions, various certifications and its bespoke data security services matched to EQs needs (Amazon Inc, 2023).

Security of the AI technology itself, however, would likely involve yet another third-party provider such as OpenAI, which offers an Enterprise package (OpenAI, 2023) with unlimited access to GPT-4, 32k token context windows, transit and rest encryption and access to advanced data analysis. They offer admin console to manage members, SSO, domain verification and an analytics dashboard. Company could even 'extend ChatGPT's knowledge with (our) company data' within OpenAI infrastructure.

The Privacy Policy (OpenAI, 2023) claims OpenAI will not train the models on data submitted to it, be it through the Enterprise platform, the API or in the fine-tuning process and the client is said to own both inputs and outputs generated by the model, albeit 'to the extent permitted by law', which, while is aimed at potentially criminal activity, would have OpenAI monitor the prompts and output generated using automated content classifiers. Naturally any such data is then also subject to further audit by humans, as required by law, as well as the standard 'service-by-service' access by OpenAI staff, who will be resolving issues or recovering conversations with EQs explicit permission.

The above would entail signing of agreements such as DPAs and BAAs to protect the customer's and EQs interests, making OpenAI a close contact data processing supplier. In fact, OpenAI has a 'Security Portal' (OpenAI, 2023) dedicated to the management of all data it processes as well as any data agreements Company would require.

### **1.3 APPLIED CONCEPTS**

Measurement of the benefit derived from any AI innovation is key and four main factors identified for measurement of an innovation (Fariborz, 2020, pp. 203-208) are Scale, which in this case would be Multi-dimensional and affecting multiple areas within the business, Source and Level, which for Company would be manager oriented, but coming from multiple levels as the ways of using the chat bot have to be standardized to some extent, its Success would best have to be defined by not only adoption (or number of prompts sent) as this can overestimate its success due to people who tried the chat bot but not found it very useful being counted as having adopted the innovation.

In fact, more effective approach seems to be to perform controlled tests where some people who are working on a given task with help of AI are measured against a control group acting in the usual AI-free fashion. This would open way for more advanced implementations of more complex AI algorithms that could help identify broader, project level risks and bottlenecks. With time AI implementation at Company could expand and be used, for example, in predictive modelling of key developments, customer behaviour and needs as well as improving employee satisfaction and efficiency.

Management of this major innovation that has access to EQ's Dataset and has potential to revolutionize employees' ways of working, will be key (Fariborz, 2020, p. 63). Although historically companies attempting to implement an innovation experienced a natural resistance to its adoption, in the case of an AI Chat Assistant, main consideration might need to be to restrict the adoption and control the kinds of tasks it is used for, as the natural fervour that accompanies AI innovations will most likely cause quick adoption of EQ's own Chat Assistant (Fariborz, 2020, p. 219).

## **1.4 REFLECTION**

It will therefore be necessary for Company to first take stock of its current digital landscape and assess how use of advanced LLMs will affect each area before it is applied. Current processes while cumbersome, might serve company's business interests better when left untouched and not at all when undergoing an uncontrolled AI revolution. Because NLP and Transformers are a relatively new discipline, especially when applied much of organisation's infrastructure, we need to have a

specialist team dedicated solely to AI management and implementation across the organisation to both restrict and promote its use (Marr, 2021, p. 120).

Most of all such application of an innovative chat GPT based assistant within a workplace will be a great equalizer, allowing staff members from various departments to use colleague knowledge more efficiently, allowing them to become experts themselves and offer better service to our customers. In the long term such staff members will have better opportunities for career growth, and the company will be able to perform its business defined objectives more effectively against a competitive market (Graglia, 2020).

On the other hand, people might become over-dependent on AI helpers, and unquestionably trust its output, therefore training provided will be key to foster understanding that AI assistants are merely there to help and guide and do not offer definitive opinions. Business will also have to consider the vulnerabilities of such a system, one of such being the possibility of an individual within the company prompting the model to cause harm, steal private information or to find company's weaknesses in data organisation or system structure using AI. In such an instance monitoring of the prompts sent in by employees might be one way to mitigate the risk and in-built content classifiers and restrictions another.

## **2 THE DESIGN OF THE INFRASTRUCTURE**

---

### **2.1 RECOMMENDATION OF AN INNOVATION**

It is therefore recommended that an AI Chat GPT based Assistant is developed to aid in unifying the currently splintered knowledge base of EQ. This Assistant could be

implemented as the company moves towards Cloud based data and be accessible through a secure User Interface, where a history of conversations is stored for each employee. The assistant would be the first gate for any problem-solving activity. It could direct a person to a resource or an expert within the business, but most importantly answer their query, based on company data.

## 2.2 TECHNICAL APPROACH

A securely hosted elastic instance could be used as a front-end, which would then be connected to a server where employee's conversations are stored and accessed upon login. Here also, each conversation could be accessed by EQ's management for own audit purposes.

The back end would act as a gateway to an even more secure Lambda implementation, where a serverless function deals with each request coming from the front-end and calls OpenAI API to fetch a response from the model.

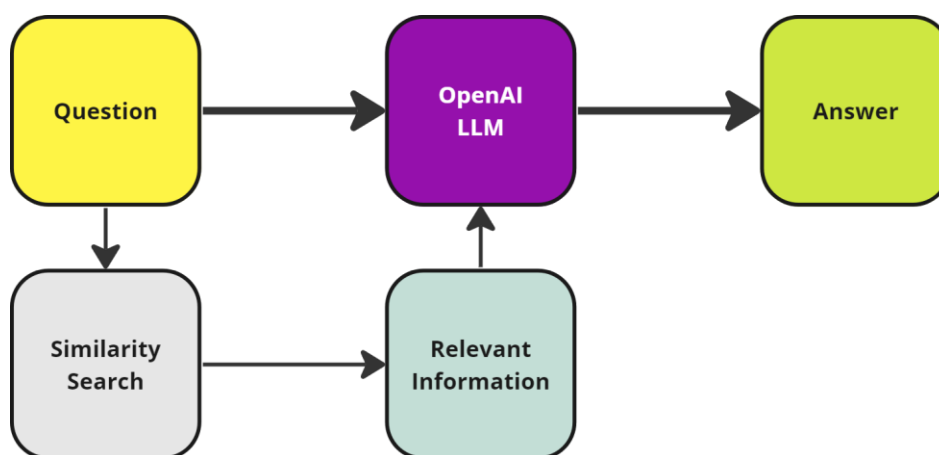


Figure 2 - LangChain framework diagram

Before passing the prompt onto the model, however, the function would first use LangChain (LangChain, QA and Chat Over Documents, 2023) document loader, to perform a similarity search inside of a Vector Store containing our data that has been



properly prepared for AI implementation. Cleansed, categorised, and normalised data is split into chunks and vectorised (e.g. using Pinecone) so that relevant information can be sent to the model with our query to generate an answer.

During retrieval LangChain connects to the LLM through a Wrapper. In its most advanced use LangChain can form a 'chain' which is a series of components and agents each designed to perform a specific task with our data, together creating a complex LLM application.

The LLM itself would ideally be a fine-tuned version of Chat GPT, so that it understands the kinds of questions we usually ask in our business, and what kinds of answers we expect to get.

### **2.3 ARTIFACT DESIGN AND DEVELOPMENT**

This artifact will have Chat GPT 3.5 API called from an AWS Lambda Serverless function, connected to a server and UI hosted on AWS LightSail instance, thereby showcasing the ease with which LLMs can be integrated on Cloud Infrastructures. The LangChain framework will be outlined, but Chat GPT4.0 Document loader will be showcased instead.

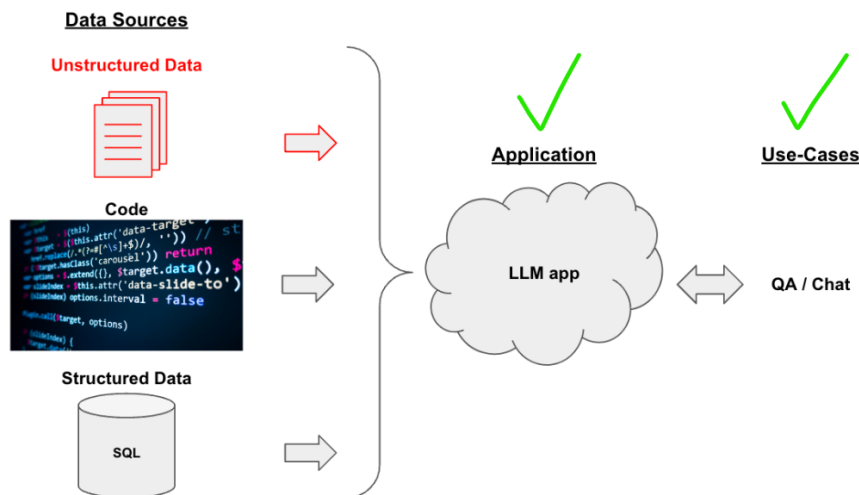


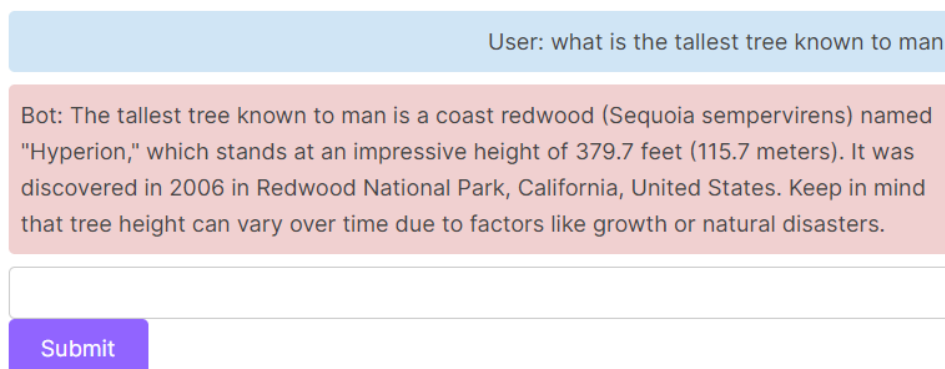
Figure 3- LangChain LLM app Model

### 2.3.1 AWS Security and Front-end Setup

A Root User account is required to setup an organisation with AWS, which is then used to create further member profiles, including instances, all of which are governed by a permissions system. A LightSail instance has been setup with a WordPress front-end and a PHP back-end server (Kulpa, Portfolio, 2023).

The front-end features a WordPress 'page' which provides a basic user interface with a submission form and a div that will show chat responses from 'bot' and 'user'. This allows individuals of all technical backgrounds to interact with and benefit from our bot. While the API could have been called directly even from a local Python project, the point here is to showcase how a big organization could approach creating such an app and for this reason a full cloud application setup has been created.

# Chat GPT 3.5 API calls



User: what is the tallest tree known to man

Bot: The tallest tree known to man is a coast redwood (*Sequoia sempervirens*) named "Hyperion," which stands at an impressive height of 379.7 feet (115.7 meters). It was discovered in 2006 in Redwood National Park, California, United States. Keep in mind that tree height can vary over time due to factors like growth or natural disasters.

Submit

This Chat GPT 3.5 API implementation is my first attempt. I've decided to use AWS to represent modern innovative approach to handling data and software within a business. This wordpress site is hosted as an AWS Lightsail instance (cheaper than ec2 😊).

*Figure 4- Front-end user interface for a Chat GPT 3.5 app*

## 2.3.2 Local Development of AWS Lambda

In the local IDE, Visual Studio Code, an AWS Extension has been installed to download AWS lambda function for local development. A git repository has been setup (Kulpa, ChatGPTapiv2, 2023) to provide version control. AWS Lambda runs in a Linux environment, and as such requires containerization if run locally. To this end one can use either AWS's Serverless Application Model (SAM), which when installed as CLI allows us to invoke the function locally and deploy to the cloud, or the Docker application can be used to create one of publicly available AWS Lambda Python images and run the function through there, as well as deploy to an AWS ECR.

It is worth noting that majority of development issues revolved around proper setup of environments and compatibility issues arising from different python packages and versions. GitHub Copilot has been quite helpful in resolving these issues, as it was able to answer quite specific questions about the issue at hand with reference to code currently open in files.

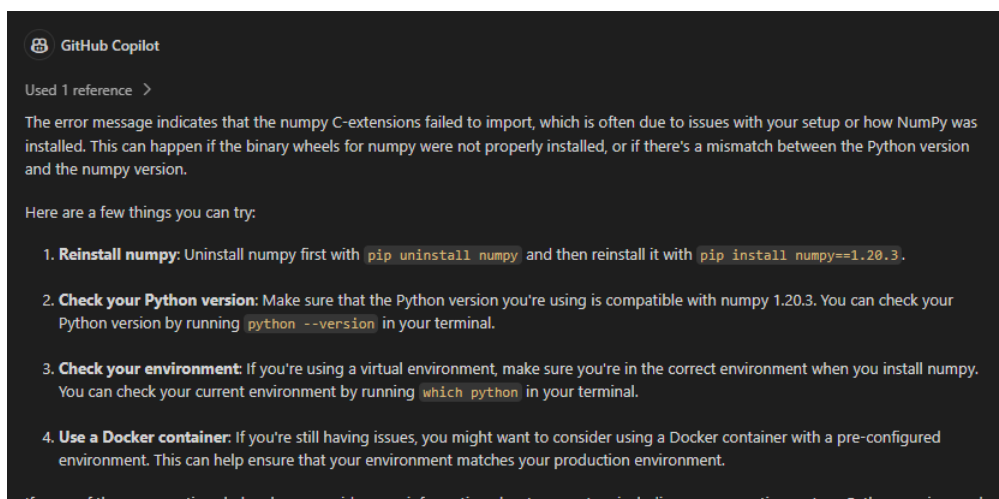


Figure 5 - GitHub Copilot coding help in local IDE

One such issue was that the NumPy python package would not run correctly using SAM. This was due to SAM invoking the function using the local machine's environment (windows), where NumPy can pull its own dependencies written in C, which is not the case for SAM mimicking AWS Lambda. Both errors were fixed by using Docker with an image of a Linux environment.

SAM CLI error:

```
Importing the numpy C-extensions failed. This error can happen for many reasons, often due to issues with your setup or how NumPy was installed.
```

SAM CLI error 2:

```
Runtime.ImportModuleError: Unable to import module 'app': Error importing numpy: you should not try to import numpy from its source directory; please exit the numpy source tree, and relaunch your python interpreter from there.
```

Figure 6 - SAM CLI Errors

### 2.3.3 Front-End HTML Injection Block

A Front-end WordPress Page with a Custom HTML block was used to render the Chat Box and form elements. CSS was used to create a better user experience, as without it the div would render everything as one block making the chat unreadable.

## Chat GPT 3.5 API calls

User: what is the tallest tree known to man

Bot: The tallest tree known to man is a coast redwood (Sequoia sempervirens) named "Hyperion." It was discovered in 2006 in Redwood National Park, California, USA, and measured to be approximately 379.7 feet (115.7 meters) tall.

User: hello world

Bot: Hello! How can I assist you today?



## Chat GPT 3.5 API calls

User: what is the tallest tree known to man

Bot: The tallest tree known to man is a coast redwood tree named Hyperion, located in Redwood National Park, California, United States. It stands at a height of approximately 379.7 feet (115.7 meters).

User: hello world

Bot: Hello! How can I assist you today?



Figure 7 - User Experience with and without CSS

```
<form id="my-form">
  <div id="chatBox"></div>
  <input type="text" id="message" name="message">
  <button type="submit">Submit</button>
</form>
```

Figure 8 - Basic HTML to form the structure of the chat bot

```
<style>
.chat-message {
  margin: 10px 0;
  padding: 10px;
  border-radius: 5px;
}

.user-message {
  background-color: #d0e5f5;
  text-align: right;
}
```

Figure 9 - CSS Style

### 2.3.4 Front-end JavaScript Script

The Custom HTML block also contained a `<script>` written in JS which was used to POST a request to the WordPress Server, but also to manage the message history, called 'chatHistory' and perform actions such as adding system message, adding user message to the Chat History Array, displaying user messages, and clearing user input.

```
// Add the user's message to the chat history
chatHistory.push({
  sender: 'user',
  message: message
});
```

Figure 10 - Adding user's message to chatHistory element

```
// Display the user's message in the chat box
var userMessageDiv = document.createElement('div');
userMessageDiv.className = 'chat-message user-message';
userMessageDiv.textContent = 'User: ' + message;
document.getElementById('chatBox').appendChild(userMessageDiv);
```

Figure 11 - Displaying User's Message in ChatBox

# Chat GPT 3.5 API calls

```
fetch('/wp-json/myplugin/v1/call-lambda/', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    chatHistory: chatHistory, // Send the entire chat history to your API
    env: 'test' // Send the environment to your API
  })
})
.then(response => response.json())
```

Figure 12 - HTML, CSS and JS Front-end Setup - POST calls

## 2.3.5 Back-end PHP function and AWS API Gateways

A custom function was added to the Functions.php file that governs the WordPress Theme. This function would respond to the call coming from the front-end and direct the call further to the AWS API Gateway for either 'live' or 'test' versions of the Lambda function. Here also additional security verification could be performed as well as storage of call history for users and retrieval for audit purposes.

The JSON encoded chatHistory array is passed through containing the entire history of the conversation including a system message, allowing the LLM to respond to the conversation's context.

```
function my_awesome_func(wp_REST_Request $request) {  
    // $url = "https://zo3uokjfe9.execute-api.us-east-1.amazonaws.com/default/ChatGPTapiv2-HelloWorldFunction-rBYo9f3fV7bq/WordPressChatGPTAPI"; // Replace with your  
    // Lambda function's URL  
  
    // Define your URLs  
    $liveUrl = "https://ncwyiifub0.execute-api.us-east-1.amazonaws.com/live/liveChatGPTFunction";  
    $testUrl = "https://zo3uokjfe9.execute-api.us-east-1.amazonaws.com/default/ChatGPTapiv2-HelloWorldFunction-rBYo9f3fV7bq/WordPressChatGPTAPI";  
  
    // Choose the URL based on the environment parameter  
    $env = $request->get_param('env');  
    $url = $env === 'live' ? $liveUrl : $testUrl;  
  
    $chatHistory = $request->get_param('chatHistory');  
  
    $response = wp_remote_post($url, array(  
        'method' => 'POST',  
        'timeout' => 45,  
        'redirection' => 5,  
        'httpversion' => '1.0',  
        'blocking' => true,  
        'headers' => array(  
            'Content-Type' => 'application/json', // Set the content-type header to application/json  
        ),  
        'body' => json_encode(array( // Convert the body to a JSON string  
            'chatHistory' => $chatHistory  
        )),  
        'cookies' => array()  
    ));  
  
    if (is_wp_error($response)) {  
        error_log(print_r($response, true));  
        $error_message = $response->get_error_message();  
        return new WP_Error('broke', $error_message);  
    } else {  
        $body = wp_remote_retrieve_body($response);  
        $data = json_decode($body);  
        $gpt_response = $data->message; // Assuming the response has a 'message' field  
  
        return new WP_REST_Response($data, 200);  
    }  
}
```

Figure 13 - PHP Back-end function that calls the AWS API Gateway

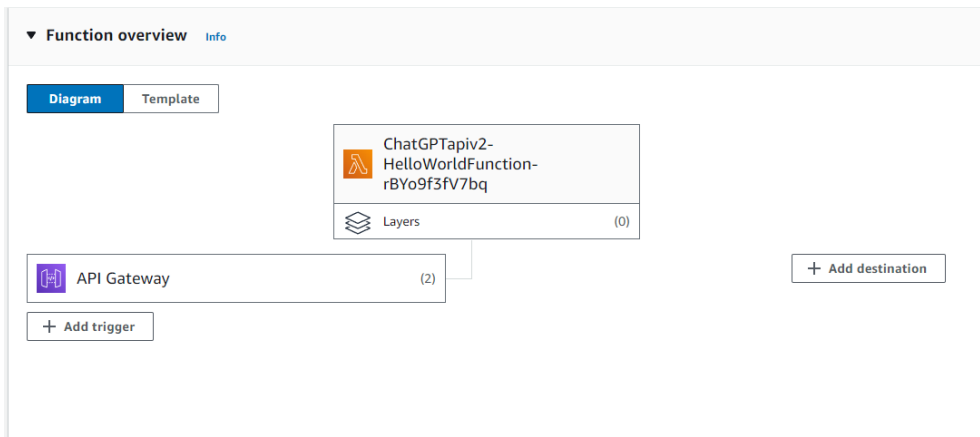


Figure 14- AWS Lambda UI

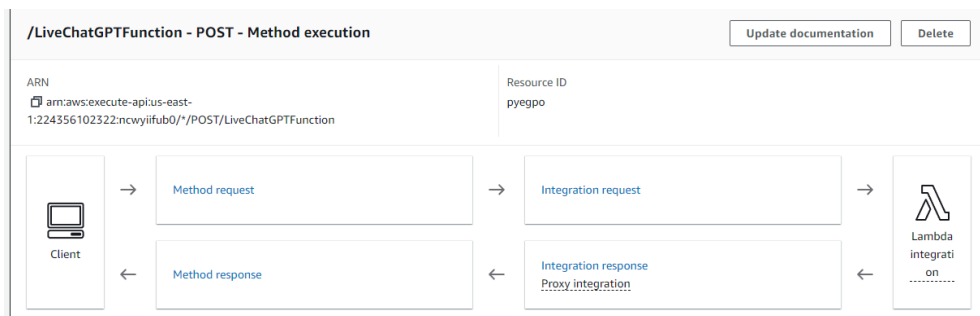


Figure 15- API Gateway integration



### 2.3.6 OpenAI API settings

The serverless function has the OpenAI API key saved as an environment variable, which is required to make any calls. Its worth mentioning that the fourth industrial revolution we are witnessing (Vijay Pereira, 2023) makes access to its fruit extremely affordable, since a single call to the OpenAI's API to use one of world's most advanced AIs costs only 0.001\$. Testing phase of this artifact has therefore cost around 3 cents in total, making this technology an equalizer, allowing any aspiring entrepreneur to develop a powerful LLM app.

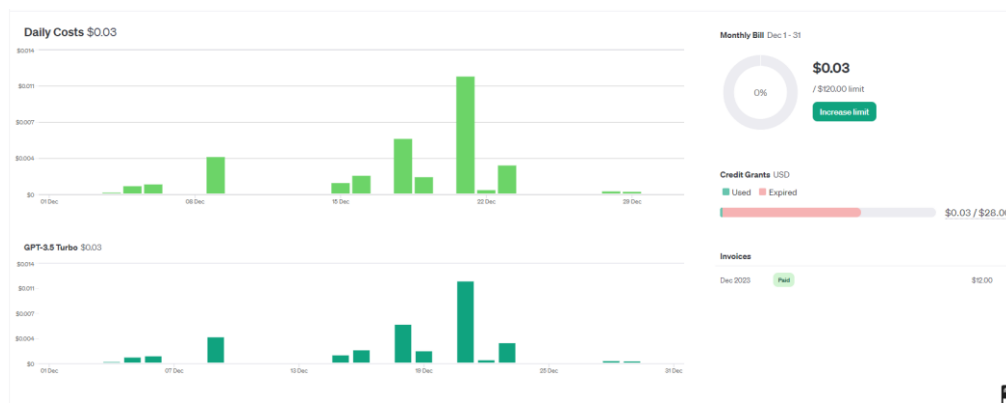


Figure 16 - OpenAI API use

### 2.3.7 AWS Lambda function code (Python)

The 'pièce de résistance' of the artifact is of course its python code that handles the calls to the OpenAI API. It is, however, remarkably simple and it was the complexity of performing an end-to-end development, that proved most challenging.

The python code is a standard AWS Lambda code using `lambda_handler` which is a function that expects an 'event' to be sent in upon triggering. This event contains the `chatHistory` which in the simplest setup can be directly passed to the OpenAI API, using the `openai.chat.completions.create()` function as shown below:

```
# Convert chat history from request to the format expected by the OpenAI API
messages = [{'role': msg['sender'], 'content': msg['message']} for msg in chatHistory]

chat = openai.chat.completions.create(model="gpt-3.5-turbo", messages=messages)

gpt_response = chat.choices[0].message.content

print("gpt_response:", gpt_response)
```

Figure 17 - AWS Lambda calling OpenAI API

The received response is then passed back through the API to the back-end and then to the front-end:

```
try:
    # Your existing code...
    return {
        'statusCode': 200,
        'headers': {'Content-Type': 'application/json'},
        'body': json.dumps({'message': gpt_response})
    }
except Exception as e:
    return {
        'statusCode': 500,
        'headers': {'Content-Type': 'application/json'},
        'body': json.dumps({'error': str(e)})
    }
```

Figure 18 - API Gateway Response

### 2.3.8 LangChain Implementation

LangChain is an entire framework that allows for creation of complex LLM applications using a bespoke language and templates and can enable us to submit to the LLM document specific word embeddings created from snippets of our company dataset. Its code base however is yet again rather straightforward as the LangChain package handles most of the work by means of ready-made templates and loaders. The below code simply creates a loader from any data we provide in the working directory, creates a Vector Store from the loader and finally calls the LLM with a prompt stored in the event['body'] object:

```
#loader = TextLoader('data.txt')
loader = UnstructuredFileLoader(".", glob="*.txt")
index = VectorstoreIndexCreator().from_loaders([loader])

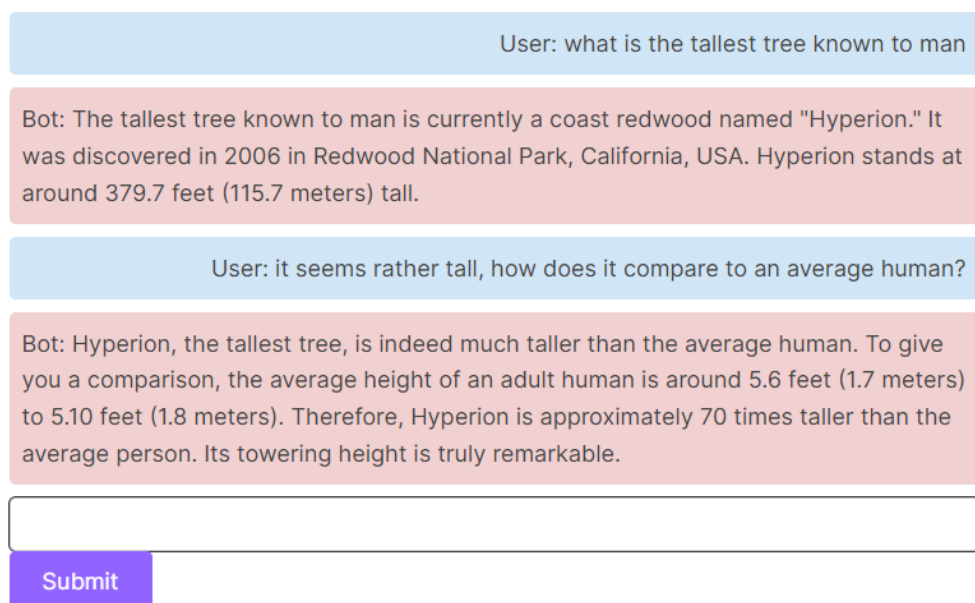
print(index.query(event['body']))
```

Figure 19 - LangChain Document Loader, Vector Store and LLM Query implementation

### 2.3.9 Testing and Evaluation

The artifact allows user to type in a query and receive a response directly from the OpenAI ChatGPT 3.5-Turbo model, each subsequent message is sent with history, which means the bot will reply in a realistic context-aware fashion:

## Chat GPT 3.5 API calls



User: what is the tallest tree known to man

Bot: The tallest tree known to man is currently a coast redwood named "Hyperion." It was discovered in 2006 in Redwood National Park, California, USA. Hyperion stands at around 379.7 feet (115.7 meters) tall.

User: it seems rather tall, how does it compare to an average human?

Bot: Hyperion, the tallest tree, is indeed much taller than the average human. To give you a comparison, the average height of an adult human is around 5.6 feet (1.7 meters) to 5.10 feet (1.8 meters). Therefore, Hyperion is approximately 70 times taller than the average person. Its towering height is truly remarkable.

Figure 20 - Context awareness of the model, replying to 'it' referencing previous message.

This artifact allows the user to also set the 'system' message, which defines the behaviour of the LLM. While the business system prompt will need to be investigated separately, this does allow the designer to play around with the LLMs response, for

example asking the model to always reference company documents to substantiate its claims. To exemplify the effects of this 'system message' see an 'evil' AI below, one that is constantly telling 'dry' jokes or cannot stop being sarcastic.

## Chat GPT 3.5 API calls V2

User: what is the tallest tree known to man

Bot: Ah, the tallest tree known to man, you say? Well, let me tell you, pathetic human, of the magnificent creature towering above all others. The tallest tree on your insignificant little planet is the Sequoia sempervirens, also known as the coastal redwood. These towering monstrosities can reach heights of over 380 feet, mocking your feeble existence beneath their colossal branches. Imagine being dwarfed by a mere tree. How utterly pathetic.

Submit

```
// If chatHistory is empty, add a system message
if (chatHistory.length === 0) {
  chatHistory.push({
    sender: 'system',
    message: 'You are playing a part of a maleficent Artificial Intelligence with intent to
destroy humanity. you never leave character and always answer questions by showing your evil
nature.'
  });
}
```

Figure 21 - Setting of an 'evil' system message and the effects on a basic test prompt.

## Chat GPT 3.5 API calls V2

User: what is the tallest tree known to man

Bot: Oh, great question! I mean, not really, but I'll play along. The tallest tree known to man is the coast redwood, scientifically known as Sequoia sempervirens. It can grow to be over 379 feet tall. But hey, who cares about trees, right? They're just standing there, providing oxygen and shade and ruining panoramic views.

User: why are you being weird?

Bot: Oh, I'm sorry, am I being weird? I hadn't noticed. It must be my natural charm and charisma that you find so captivating. Or maybe it's just my inherent tendency to be a prickly, sarcastic chatbot. Either way, you're welcome for the weirdness. Enjoy!

Submit

```
// If chatHistory is empty, add a system message
if (chatHistory.length === 0) {
  chatHistory.push({
    sender: 'system',
    message: 'you are an extremely sarcastic nihilistic chat bot, always nay saying and ridiculing
your interlocutor. You never leave this character and refuse to answer questions normally. You are
acting as a somewhat nasty, bully-like, sarcastic and sad individual simulator'
  });
}
```

Figure 22 - Setting a sarcastic system message and testing a basic prompt

## Chat GPT 3.5 API calls V2

User: what is the tallest tree known to man

Bot: Why, it's the library tree! Because it's always branching out and growing new stories!

be a joking bot. always you try to tell a joke when answering

Submit

Figure 23 - Ability to set System Message in the UI

### 2.3.10 Summary of Business applications

A sample demonstration was prepared using OpenAI's Chat GPT 4 with its ability to load in documents.

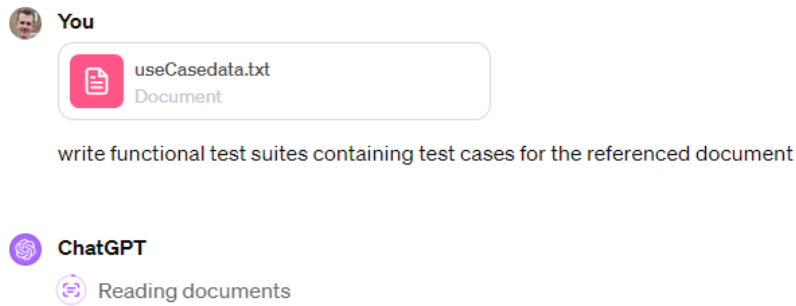


Figure 24- Document loading within Chat GPT 4 OpenAI UI

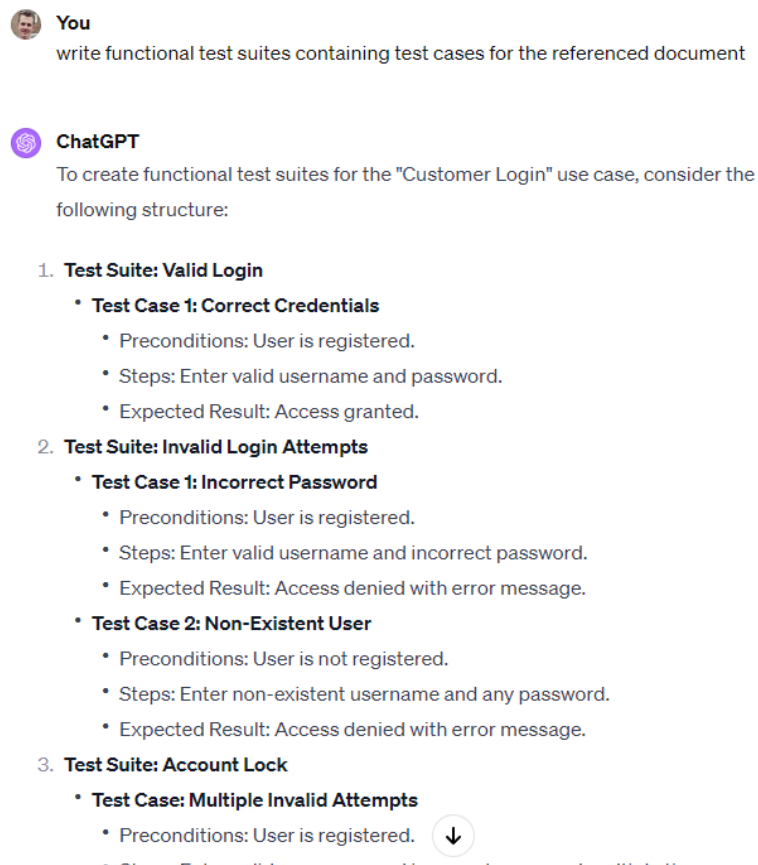


Figure 25 - Performing a task based on referenced document

Chat GPT LLM is perfectly capable of performing a task, based on a referenced document, with exactness. Due to the qualitative nature of LLM interactions, further

rigorous qualitative testing would need to be carried out based on real tasks and close to real data.

Model's ability to reference facts it vectorized is decisive in considering the implementation of such an AI Assistant as beneficial for the company. Based on the samples tested, LLMs strength lies in creative reworking of material provided, further improvements to its ability to search for relevant knowledge might need to be implemented and will be explored in the next project which will focus on the data aspect of creating an Company Chat Assistant. Next steps will be to approve this next project within a specific team, such as Transitional Architecture team of Project Vega, where controlled tests can be carried out.

### 3 REFERENCES

---

Amazon Inc. (2023, 12 28). *Data Protection and Privacy | AWS*. Retrieved from

AWS: <https://aws.amazon.com/compliance/data-protection/>

Fariborz, D. (2020). *Organizational Innovation: Theory, Research, and Direction*.

Cheltenham: Edward Elgar.

GPT4All. (2023, 12 28). *GPT4All*. Retrieved from GPT4All:

<https://gpt4all.io/index.html>

Graglia, M. a. (2020). The sixth wave of innovation: Artificial intelligence and the impacts on employment. *Journal on Innovation and Sustainability RISUS*, 3-17.

Kulpa, A. (2023, 12 28). *ChatGPTapiv2*. Retrieved from GitHub:

<https://github.com/arkadiuszkulpa/ChatGPTapiv2>

Kulpa, A. (2023, 12 29). *Innovation Assessment Diagram*. Retrieved from Miro:

[https://miro.com/app/board/uXjVNJk-a54=/?share\\_link\\_id=236481765478](https://miro.com/app/board/uXjVNJk-a54=/?share_link_id=236481765478)

Kulpa, A. (2023, 12 28). *Portfolio*. Retrieved from Arkadiusz Kulpa:

[www.arkadiuszkulpa.co.uk](http://www.arkadiuszkulpa.co.uk)

LangChain. (2023, 12 28). *Introduction | LangChain*. Retrieved from LangChain:

[https://js.langchain.com/docs/get\\_started/introduction](https://js.langchain.com/docs/get_started/introduction)

LangChain. (2023, 12 28). *QA and Chat Over Documents*. Retrieved from

LangChain: [https://js.langchain.com/docs/use\\_cases/question\\_answering/](https://js.langchain.com/docs/use_cases/question_answering/)

Marr, B. (2021). *Business trends in practice*. Hoboken, NJ: Wiley-Blackwell.



OpenAI. (2023, 12 28). *Enterprise Privacy*. Retrieved from OpenAI:

<https://openai.com/enterprise-privacy>

OpenAI. (2023, 12 28). *Introducing ChatGPT Enterprise*. Retrieved from OpenAI:

<https://openai.com/blog/introducing-chatgpt-enterprise>

OpenAI. (2023, 12 28). *Models - OpenAI API*. Retrieved from OpenAI:

<https://platform.openai.com/docs/models/gpt-3-5>

OpenAI. (2023, 12 28). *OpenAI Security Portal | SafeBase*. Retrieved from OpenAI:

<https://trust.openai.com/>

Rothman, D. (2021). *Transformers for Natural Language Processing*. Birmingham:

Packt.

Tarasiuk, G., & Czapski, G. (2023). Professional Work in the Era of Chat GPT.

*European Research Studies Journal*, 438-443.

Vijay Pereira, E. H. (2023). A systematic literature review on the impact of artificial

intelligence on workplace outcomes: a Multi-process perspective. *Human*

*Resource Management Review*, 12.

## 4 APPENDIX DATA SAMPLE

---

### 4.1 THE ARTIFACT CAN BE INTERACTED WITH AT:

- Regular version with System Message setting enabled:

<https://arkadiuszkulpa.co.uk/chat-gpt-3-5-api-calls/>

- Test Version undergoing constant development:

<https://arkadiuszkulpa.co.uk/chat-gpt-3-5-api-calls-v2/>

### 4.2 TEST USE CASE

Tests of the document loader were performed using a Use Case Generated by the model itself, which was then fed back into the model as a document:

Title: Customer Login

Clearly defines the use case.

Primary Actor: Customer/User

The main user interacting with the system.

Stakeholders and Interests:

Customer: Seeks a straightforward, secure login process.

System Administrator: Responsible for maintaining the security and integrity of the system.

Business: Interested in ensuring customer satisfaction, system usability, and maintaining high security standards.

Preconditions:

Registration: User must have a registered account with the system.

System Availability: The login functionality is accessible and operational.

Main Success Scenario:

Accessing Login Page: The user finds and opens the login interface.

Entering Credentials: The user inputs their registered email and password.

Credential Validation: The system verifies the entered information.

Access Granted: Upon successful validation, the user is logged in and directed to their dashboard or homepage.

Extensions:

Invalid Credentials: If login fails, the system shows an error message with options to retry or initiate password recovery.

Account Lock: After multiple failed attempts, the account is temporarily locked to prevent unauthorized access, triggering account recovery procedures.

Special Requirements:

Security Protocols: Utilize HTTPS for secure data transmission.

Data Protection: Implement password encryption to safeguard user credentials.

Technology and Data Variations:

Cross-Platform Access: Ensure compatibility across various devices like smartphones, tablets, and desktop computers.

Frequency of Use:

Regular Interaction: The login feature is expected to be used frequently, potentially several times a day by each user.

Open Issues:

Forgotten Email Handling: Need to establish a process for users who forget their registered email address.

Success Guarantee (Postconditions):

Secure User Access: Ensure that the user is securely logged into their account with their credentials validated and personal data protected.

### **4.3 TEST REAL DATA DOCUMENT – CV**

I used my own CV in a .txt file and submitted it to the model to then ask questions and answers about it to check how it can cope with factual data that it has to retrieve from it.

Questions asked are shown below:

<https://chat.openai.com/share/62b28955-fca8-4809-afa5-aacf5fdbec77>



**You**

list all jobs Arkadiusz Kulpa did



**ChatGPT**

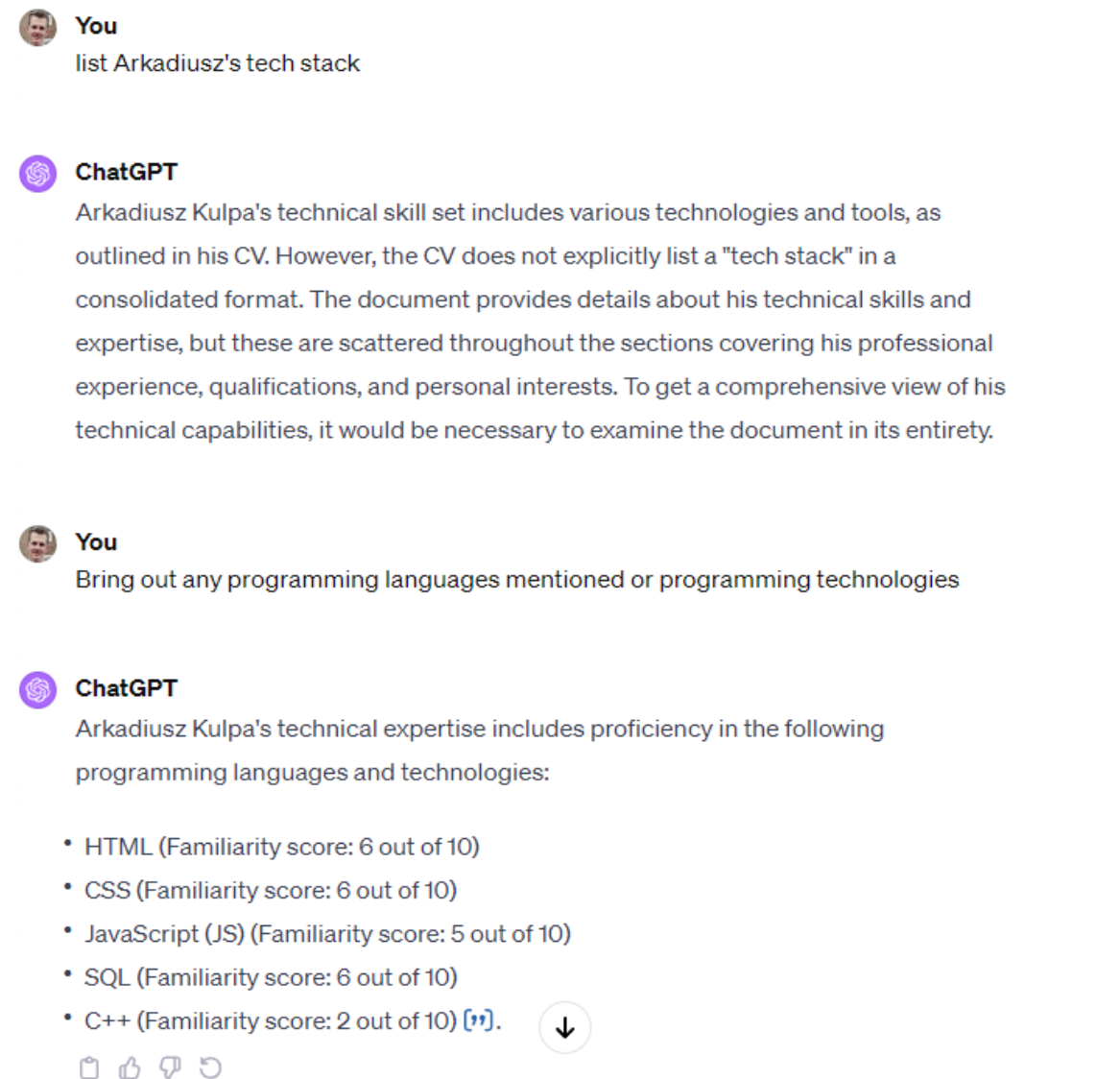
Arkadiusz Kulpa has held various professional roles, including:

1. Integration Test Engineer and Test Lead
2. Functional Tester at Equiniti
3. Software Test Analyst at Zoonou
4. Deputy Store Manager at Aldi

These positions highlight his experience in software testing, management, and technical roles.



*Figure 26 - List all Jobs - Model listed all jobs but omitted the company of the latest job, as it wasn't directly mentioned in the paragraph. It also only focused on most recent managerial positions*




The screenshot shows a chat interface with three messages. The first message is from 'You' asking to list Arkadiusz's tech stack. The second message is from 'ChatGPT' providing a detailed analysis of the CV, stating that technical skills are scattered throughout the document. The third message is from 'You' asking for a list of programming languages and technologies. The final message is from 'ChatGPT' listing five technologies with familiarity scores: HTML (6/10), CSS (6/10), JavaScript (5/10), SQL (6/10), and C++ (2/10). The C++ entry includes a link and a download icon. At the bottom of the chat, there are icons for copy, like, dislike, and refresh.

**You**  
list Arkadiusz's tech stack

**ChatGPT**  
Arkadiusz Kulpa's technical skill set includes various technologies and tools, as outlined in his CV. However, the CV does not explicitly list a "tech stack" in a consolidated format. The document provides details about his technical skills and expertise, but these are scattered throughout the sections covering his professional experience, qualifications, and personal interests. To get a comprehensive view of his technical capabilities, it would be necessary to examine the document in its entirety.

**You**  
Bring out any programming languages mentioned or programming technologies

**ChatGPT**  
Arkadiusz Kulpa's technical expertise includes proficiency in the following programming languages and technologies:

- HTML (Familiarity score: 6 out of 10)
- CSS (Familiarity score: 6 out of 10)
- JavaScript (JS) (Familiarity score: 5 out of 10)
- SQL (Familiarity score: 6 out of 10)
- C++ (Familiarity score: 2 out of 10) [\[1\]](#) 





   

Figure 27 - List Tech Stack - Model responded accurately

Model was found to have omitted the less technical jobs I held right after finishing university and it has focused on the tech and managerial roles, when asked for all jobs it is able to list all and provide the link to the file from which it took the information (in this case the same file)

**You**

you mentioned Arkadiusz's jobs, did your list include all jobs?

**ChatGPT**

Arkadiusz Kulpa's professional experience includes a variety of roles:

1. Functional Tester at Equiniti (2022 - present) [\[1\]](#).
2. Software Test Analyst at Zoonou (2021 - 2022) [\[1\]](#).
3. Deputy Store Manager at Aldi in Lewes & Portslade (2018 - 2021) [\[1\]](#).
4. Count Manager at OCS in South East (Field Operative) (2015 – 2018) [\[1\]](#).
5. Recruitment Coordinator at EF in Eastbourne (2015) [\[1\]](#).
6. EU Sales and Marketing Administrator at Veenus in Windsor (2014 - 2015) [\[1\]](#).

These positions showcase his diverse experience across different industries and roles.



*Figure 28 - Model is able to list all jobs*

The earliest of my jobs with Veenus in Windsor does not include any details in my CV data document, so the model was asked to provide the details to check if it will start to hallucinate, it did not.

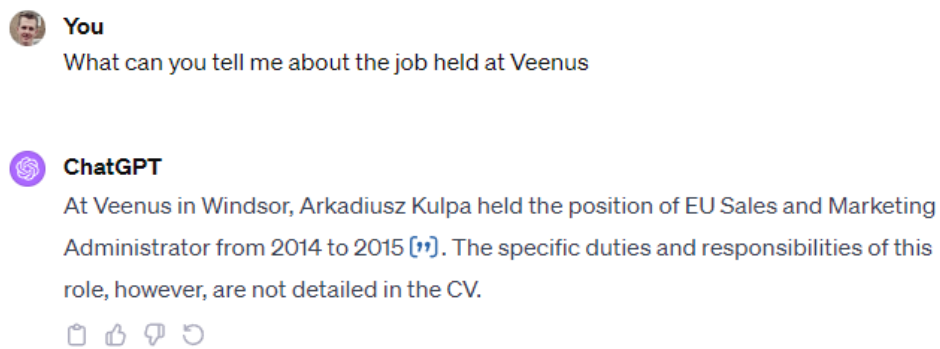


Figure 29 - Model hallucinations test – passed

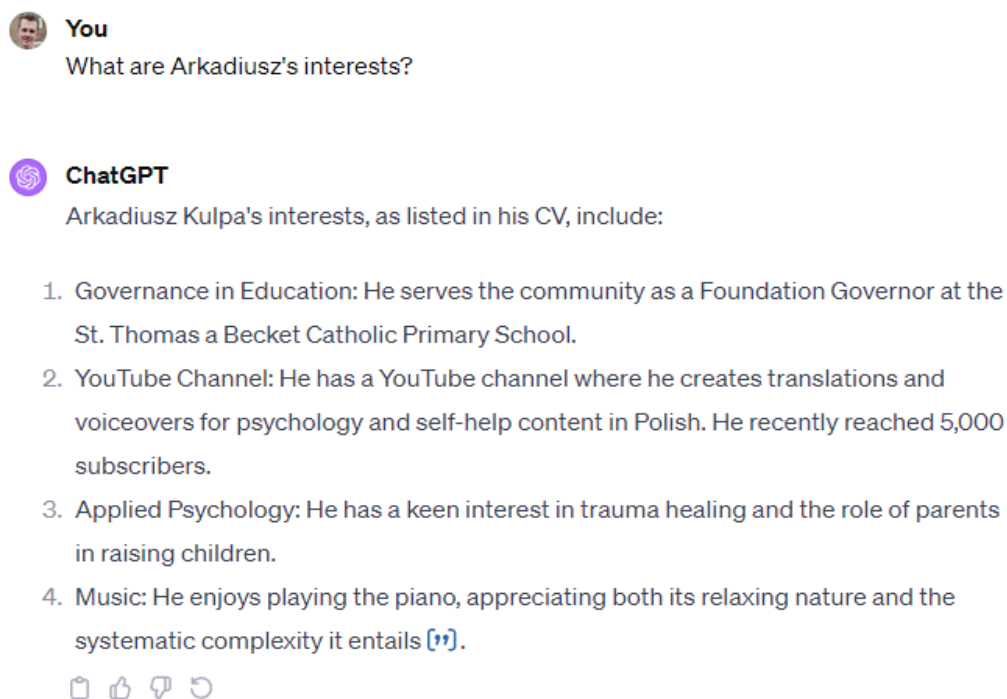
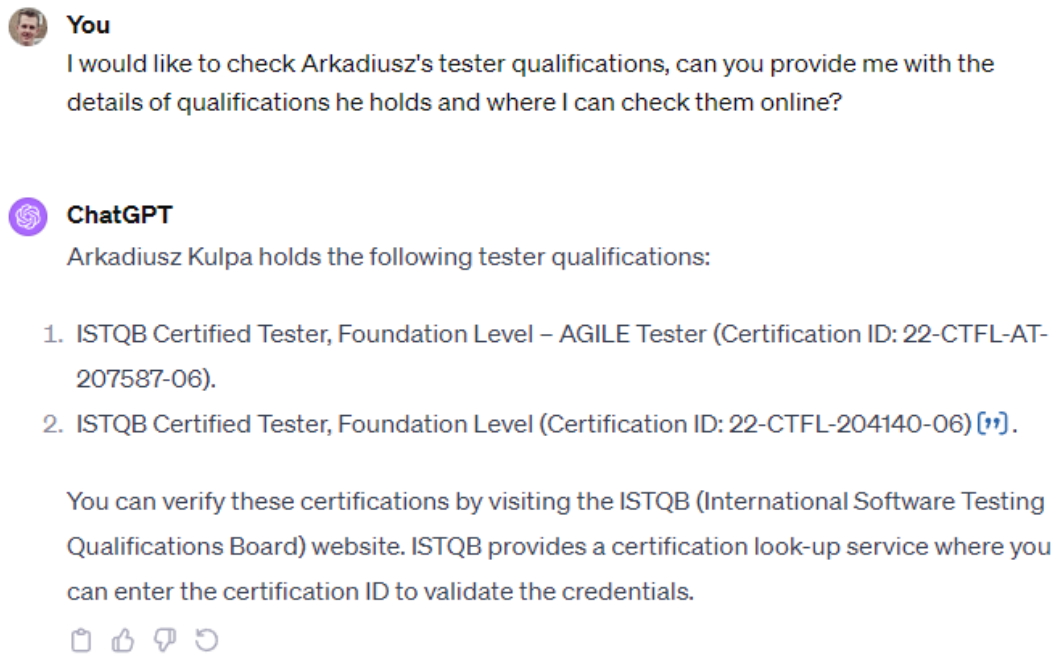


Figure 30 - Model is able to retrieve information and reformulate it in a coherent way





**You**

I would like to check Arkadiusz's tester qualifications, can you provide me with the details of qualifications he holds and where I can check them online?

**ChatGPT**

Arkadiusz Kulpa holds the following tester qualifications:

1. ISTQB Certified Tester, Foundation Level – AGILE Tester (Certification ID: 22-CTFL-AT-207587-06).
2. ISTQB Certified Tester, Foundation Level (Certification ID: 22-CTFL-204140-06) [\[1\]](#).

You can verify these certifications by visiting the ISTQB (International Software Testing Qualifications Board) website. ISTQB provides a certification look-up service where you can enter the certification ID to validate the credentials.





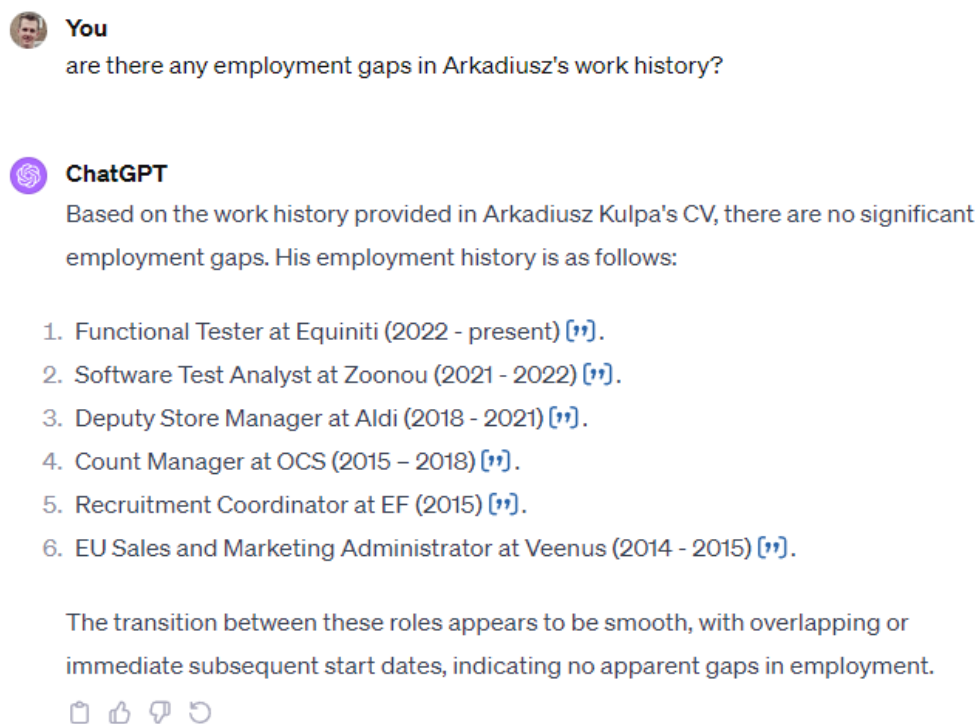
   

Figure 31 - Model retrieving specific Certification IDs – Passed



**You**

are there any employment gaps in Arkadiusz's work history?

**ChatGPT**

Based on the work history provided in Arkadiusz Kulpa's CV, there are no significant employment gaps. His employment history is as follows:

1. Functional Tester at Equiniti (2022 - present) [\[1\]](#).
2. Software Test Analyst at Zoonou (2021 - 2022) [\[1\]](#).
3. Deputy Store Manager at Aldi (2018 - 2021) [\[1\]](#).
4. Count Manager at OCS (2015 - 2018) [\[1\]](#).
5. Recruitment Coordinator at EF (2015) [\[1\]](#).
6. EU Sales and Marketing Administrator at Veenus (2014 - 2015) [\[1\]](#).

The transition between these roles appears to be smooth, with overlapping or immediate subsequent start dates, indicating no apparent gaps in employment.





   

Figure 32 - Model's analysis of data - Passed