Enhancing AI Capabilities: ParsonsGPT Indexes, Workflows, and Marketplaces

Aakash Pamnani, Aiden Deffner, Akash Wudali, Albert Jeng,
Allen Huo, and Peter A. Rochford
Innovative Technology Solutions (ITS) Directorate
Security and Mission Solutions (SMS) Sector
Engineered Systems (ES) Business Unit
Parsons Corporation
5875 Trinity Pkwy, Suite 230
Centreville, VA 20120

Peter.Rochford@parsons.us

Created: 7/22/2025
Revised: 8/8/2025

Abstract — The ParsonsGPT project aims to enhance enterprise productivity through the implementation of a secure, private instance of Generative Pre-Trained Transformer (GPT) models tailored for Parsons Corporation. To improve its utility for productivity and collaboration, advancements were made by implementing Index Expansion and Workflows within ParsonsGPT. Index Expansion introduces personal indexes for intelligent document search and a marketplace for sharing and interacting with indexes, supported by the scalable and cost-efficient Qdrant vector store. Workflows streamline repetitive tasks through query and output steps, generating structured outputs like PDFs and Word documents. A Workflow Marketplace has also been implemented that fosters collaboration by enabling users to share and access published workflows. Use cases such as data visualization, storytelling, and report generation highlight the flexibility and efficiency of ParsonsGPT as a secure, scalable, and user-friendly organizational tool.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

The recent advent of Large Language Models (LLMs) has introduced a paradigm shift in the ability to extract and generate information from large numbers of documents as substantiated by studies that find companies with workforces most exposed to generative artificial intelligence (AI) are outperforming the market (Wanna 2023, Eisfeldt, Schubert and Zhang 2023). To leverage the power of LLMs and empower staff to dramatically increase their business productivity, Parsons created a web interface to its own secure and private instance of Generative Pre-trained Transformer (GPT) models (henceforth referred to as ParsonsGPT). These are the latest versions of the ChatGPT models developed by OpenAI (OpenAI 2023) that are offered in Microsoft Azure (Microsoft Corporation 2023). In a continual effort to make ParsonsGPT a more useful enterprise resource, major changes were made to its user interface and features to make its information ingest capabilities more easily accessible.

2 Background

ParsonsGPT is a web interface to the OpenAI products in Microsoft (MS) Azure, namely the ChatGPT natural language processing (OpenAI 2024) and DALL-E text-to-image models (OpenAI 2021, OpenAI 2022, OpenAI 2023). It supports a wide variety of capabilities that include ChatGPT's standard LLM capabilities such as Natural Language Understanding and Generation, Conversational Abilities, Information Retrieval, Text Summarization, Translation, Content Creation, Image Recognition, and Code Assistance. Moreover, with DALL-E users have access to the capabilities of Text-to-Image Generation, Creative Image Synthesis, High-Resolution Outputs, Versatile Styles, and Conceptual Understanding. It provides the capability to "Bring Your Own Documents" by allowing you to create an index that is a repository of files, thereby providing ChatGPT with access to domain-specific knowledge bases to its LLM. A top-level view of the process that occurs when ParsonsGPT, using Retrieval Augmented Generation (RAG), executes the submission of a query and produces a response with an LLM is shown in Figure 1. For a description of RAG, the reader is referred to (Rochford, Daly and Smalley 2024, Appendix B).

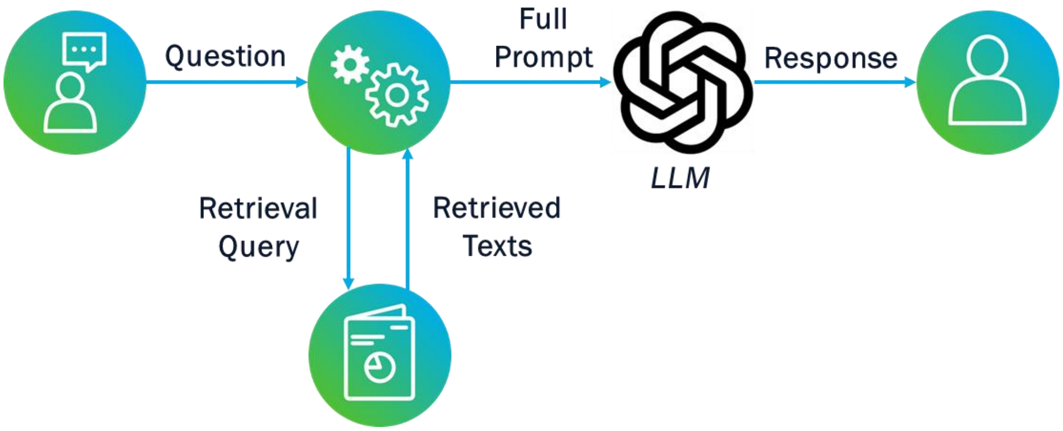

Figure 1. Top level view of query & response process when using an LLM.

Our approach with ParsonsGPT is to use the latest rapidly evolving technologies that are becoming available in the current AI revolution. Our software solution uses the most recent OpenAI products in Azure, thereby bringing the newest capabilities of LLMs into the hands of users without incurring the costly delays in development and deployment. ParsonsGPT is distinct from other commercial offerings in that it is a private instance that ensures no information is shared with OpenAI, Microsoft, or other users, maintaining the confidentiality and security of sensitive company information through encrypted conversations. It is also now available on Azure Government Cloud (GovCloud) where government restricted information can be stored and processed. As of the date of this report, this capability is an enterprise resource that is used by 1500+ Parsons staff.

The documentation for ParsonsGPT users is comprehensive and categorized to cater to the two user roles of Chat Users and Knowledge Managers (Table 1). These information resources are available on the ParsonsGPT SharePoint site (Parsons Corporation 2024) for readers seeking a foundational understanding of LLMs and step-by-step instructions on how to access and utilize ParsonsGPT. They include practical examples such as generating emails, summarizing text, and creating indexes for querying multiple documents. These guides enhance productivity by providing users with a quick reference to various tasks that can be accomplished using ParsonsGPT. Video tutorials are also available that complement these written guides, ensuring that both non-technical and technical staff can fully leverage the capabilities of ParsonsGPT in their respective roles.

Table 1. ParsonsGPT information resources.

Information Resource	Description
ParsonsGPT 101, 102, 103, & 104	Video recorded webinars
ParsonsGPT Beginner Guide	Beginner guide for non-technical and technical staff
ParsonsGPT Chat User Guide	Guide for on ParsonsGPT chat features
ParsonsGPT GovCloud Guide	Recommended for ParsonsGPT GovCloud users who seek guidance on which controlled information may be uploaded.
ParsonsGPT Knowledge Manager Guide	Instructions for managing document ingestion and index creation
ParsonsGPT Knowledge Manager Reference Manual	A comprehensive reference that details all the features available in the Knowledge Manager user interfaces of ParsonsGPT.
ParsonsGPT Quick Start Guide	Help new users quickly get access to ParsonsGPT
ParsonsGPT Software Developer Guide	Provides examples of how ParsonsGPT can be used for generating computer code.
ParsonsGPT Supplemental Guide	Miscellaneous examples

Information Resource	Description
ParsonsGPT User Reference Manual	Detailed feature descriptions of user interfaces
Migrating Conversations and Indexes in ParsonsGPT	Outlines the steps for migrating a conversation with an index from one instance of ParsonsGPT to another
ParsonsGPT Use Cases for Qualified Opportunities	A guide for ParsonsGPT users who have been granted a Knowledge Manager role and wish to leverage the power offered by Large Language Models (LLMs) for creating proposals as part of Parsons bidding efforts.

3 Index Expansion

Currently, Azure AI Search is used to store vector embeddings for indexes. The cost of this storage is high, especially since other included features aren't being utilized. The goal of index expansion is to find a cheaper alternative that is more dedicated to vector storage. Additionally, access to indexes was limited to administrators and managers. However, we plan on expanding this feature to include employees, enabling them to leverage the full range of index capabilities without the need to depend on index creation by an administrator or manager. This expansion will not only save costs but allow more users to use the indexing feature. With this planned expansion, we will develop additional features such as an index creation tool and index marketplace for general users.

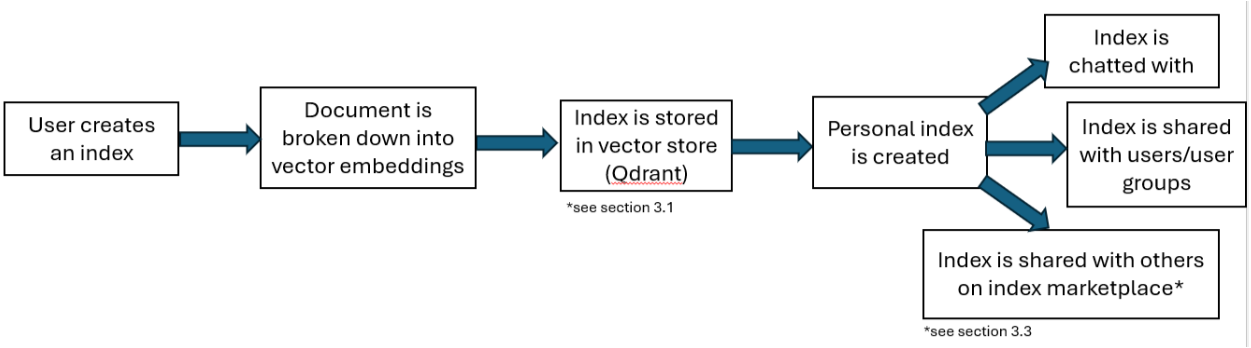
User creates an index
Document is broken down into vector embeddings
Index is stored in vector store (Qdrant)
*see section 3.1
Personal index is created
Index is chatted with
Index is shared with users/user groups
Index is shared with others on index marketplace*
*see section 3.3

Figure 2. Algorithm process for new ParsonsGPT vector Store.

3.1 Alternate Vector Store

A vector is a store of vector embeddings. A GPT model will use these embeddings to search indexes to find the most similar result to a query. An index is a group of documents, which is split into multiple chunks (usually a string), which is then converted into a mathematical representation of that chunk, also known as a vector. Doing this allows vector stores (a storage system of different vectors) to mathematically compare very quickly those vectors from a query that is also represented by vectors.

Currently, index documents (and therefore the vector embeddings as well) are stored using Azure AI Search. Alternate options were explored to both decrease cost and increase storage capabilities. The team conducted research on several vector stores including Qdrant, pgvector, Pinecone, and CosmosDB (Parsons Corporation 2025). Performance tests were developed and conducted to compare concurrency and scalability for the most promising vector stores, namely Qdrant and pgvector. The benchmark was optimized to focus the difference more on the vector store rather than the end-to-end process. This involved creating a cache (using the signatures of files) for the chunks and the vector embeddings. Ultimately, it was decided that migrating to Qdrant would be most effective for both performance and future scaling and will allow more indexes to be stored and queried efficiently, which is crucial for the additional features of index expansion.

Qdrant benchmarks are run by downloading the zip file of Qdrant from GitHub (Parsons Corporation 2025), extracting it, running that application in a terminal window, and then running the benchmark as a separate process in a different terminal window.

3.2 Personal Indexes

This feature allows individuals to upload documents directly from their devices. Uploaded files are automatically broken down into smaller sections and stored as vector embeddings, enabling intelligent search and interaction with the content. Personal Indexes are ideal for summarizing documents, extracting key information or simply retrieving specific details through conversation.

Creating a Personal Index
When setting up an index, users must:

- Provide a Title and Description that explain the purpose of the index.
- Assign an Information Sensitivity label to indicate the level of sensitivity of the data.
- Accept the ParsonsGPT policy after reviewing it.

Users also have the option to:

- Set or modify the expiration date of the index. Once expired, the index can no longer be used for chat interactions.
- Add users and create user groups for sharing access to indexes.
- Choose to make indexes discoverable in the Index Marketplace for broader visibility.
- Select a Persona, which customizes the AI's tone and behavior when interacting with the index.

Detailed instructions on how to create a personal index can be easily obtained by querying the ParsonsGPT Help index within ParsonsGPT. Written instructions with examples can also be found in the ParsonsGPT Beginner Guide (Rochford, Daly and Smalley 2024) and ParsonsGPT Knowledge Manager Manual (P. A. Rochford 2025) which are available on the ParsonsGPT SharePoint site.

3.3 Index Marketplace

The ParsonsGPT index marketplace is designed to be a social expansion of the existing index browser in the administrator console. Switching to Qdrant as the new vector store provides the capability to expand index usage across more employees at decent speeds with the index marketplace allowing the sharing and discussion of these indexes. Users can visit the index marketplace from the chat application to view, upvote, comment on, and request access to indexes created by other users. The marketplace also includes a search feature for several index fields and various forms of filtering and sorting. Currently, all functionality related to indexes is only accessible via the administrator console.

NOTE: To create Personal Indexes and access the Index Marketplace, users must have *Document User* permissions. These features are available to all users except employees in the Europe, Middle East, and Africa (EMEA) business unit who are not managers or administrators.

4 ParsonsGPT Workflow

People use workflows in AI tools to streamline repetitive tasks, improve productivity, and ensure consistency in outputs across various applications. Workflows allow users to integrate AI tools into their processes for tasks such as summarizing meeting notes, generating first-draft emails, simplifying technical documentation, and creating standardized templates for repeatable tasks. By embedding ParsonsGPT into workflows, Parsons staff can save time, reduce bottlenecks, and focus on higher-value activities, making it a valuable tool for automation and efficiency. To empower Parsons staff, a user interface (UI) was designed and implemented for users to create, run, and share structured workflows.

Workflows are defined as a repeatable set of steps designed to accomplish a process, allowing users to create and repeatedly run a streamlined and automated series of tasks. Currently, the only workflow process available in ParsonsGPT is a “multi-prompt” system that is only achievable by uploading a Comma-Separated Values (CSV) file with multiple prompts to run sequentially in the chat application. Implementing workflows in the chat application allows for a more streamlined and automated alternative to this approach, offering a user-friendly UI that eliminates the need for CSV files. Implementing workflows empowers users to create reproducible outputs that can be shared with others to achieve specific results. Potential use cases that illustrate the versatility of workflows are document templating and automating tasks like sending emails or performing web searches. By implementing a UI for Workflows, users can easily create and share workflows to help with repeatable tasks that leverage the power and capabilities of ParsonsGPT.

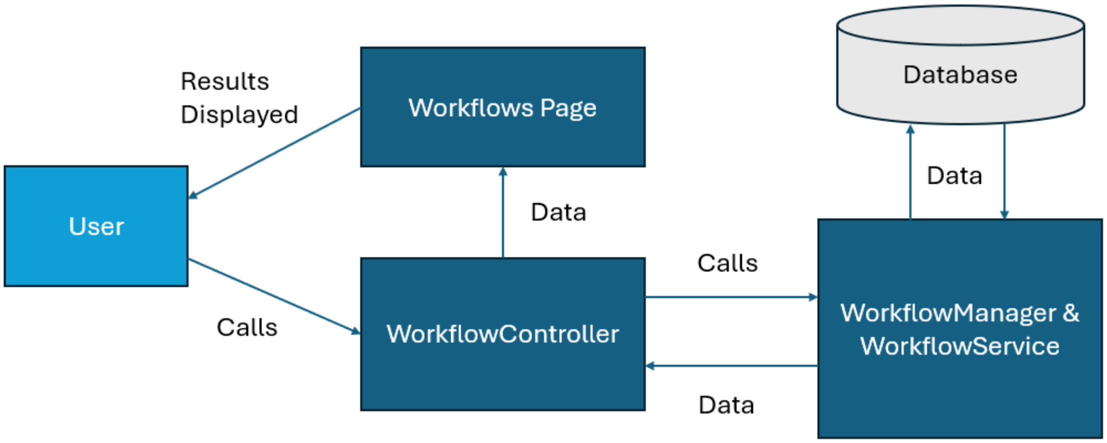

Figure 3. Algorithm process for workflow tasks.

The workflow process involves creating a structured sequence of steps to achieve a specific task or output. It begins with defining **query steps**, which allow users to specify prompts, models, agents, indexes, personas, and file attachments. Each query step interacts with the model or agent to generate an output based on the provided data. Users can edit these steps, run them individually, or execute all steps sequentially to test and refine their workflows.

Once the query steps are completed, the process culminates in an **output step**, which defines how the results are saved, displayed, or delivered. Output types include text files, PDF files, DOCX files, and emails. The output step consolidates all prior step outputs and their prompts into a result, ensuring completeness and accuracy. It uses GPT-4.1, which has a large token window to accommodate extensive workflows.

Users can also leverage agents, such as the **code interpreter agent** (for running Python code and analyzing data) and the **web search agent** (for gathering information from the internet). These agents enhance the workflow's capabilities by enabling tasks like data visualization, file creation, and web-based research.

Additionally, workflows can be shared through a **workflow marketplace**, where users can access and utilize published workflows created by others. This feature promotes collaboration and reproducibility of outputs. Users can share workflows, and in the future, features like support document template filling, structured outputs for API calls, integrations with tools like Azure Functions, and voting and favoriting workflows will be added to enhance the marketplace experience.

4.1 Backend Overview

The Workflow consists of two different types of steps: Query Steps (the normal LLM/agent prompting steps), and Output Steps (defines the deliverables). Users can add Query Steps, and an Output Step will always be defined. These steps are implemented through an abstract Step class with an abstract Run method: the QueryStep and OutputStep extend it to define their own Run method.

4.2 Backend Architecture

A Model View Controller design architecture was used to connect the user interface to the workflows backend and database. The Controller, implemented using the .NET framework, is responsible for reading and executing user requests such as adding, deleting, and running workflows and their steps.

In the workflow controller, each endpoint is responsible for a particular action. To allow for code readability and reusability in the controller, a separate service is defined to execute these actions and a designated WorkflowManager class to manage and run these actions. Each controller endpoint call methods of a WorkflowManager class, delegating all the logic to said class.

Apart from implementing the business logic for interacting with the database, the WorkflowManager class also validates the workflow. It ensures that it starts with a Query step, ends with an output step and that all query and output steps have the required information to run effectively. This rigid structure ensures that each workflow will work as intended by the user and makes it easier for non-technical users to develop more customized workflows.

The fundamental building blocks for a workflow are the workflow steps. As mentioned above, we organize steps into two types (Query and Output), and these two Step types are sub-classes of an abstract Step class. The method that each sub-class must implement is a Run method, which returns the text and file Uniform Resource Identifier (URI) generated from a particular step. The Step class will also hold the context from previous steps, which is accumulated and set in the WorkflowManager class. This context is structured as follows:

Step [step number]

Prompt:

[Prompt for the particular step]

Response:

[The result of the step]

This structured context is easy for an LLM to understand, allowing later query and output steps to reference previous output steps.

Database & Tables

The Workflows store user's Workflows and Workflow runs through various tables.

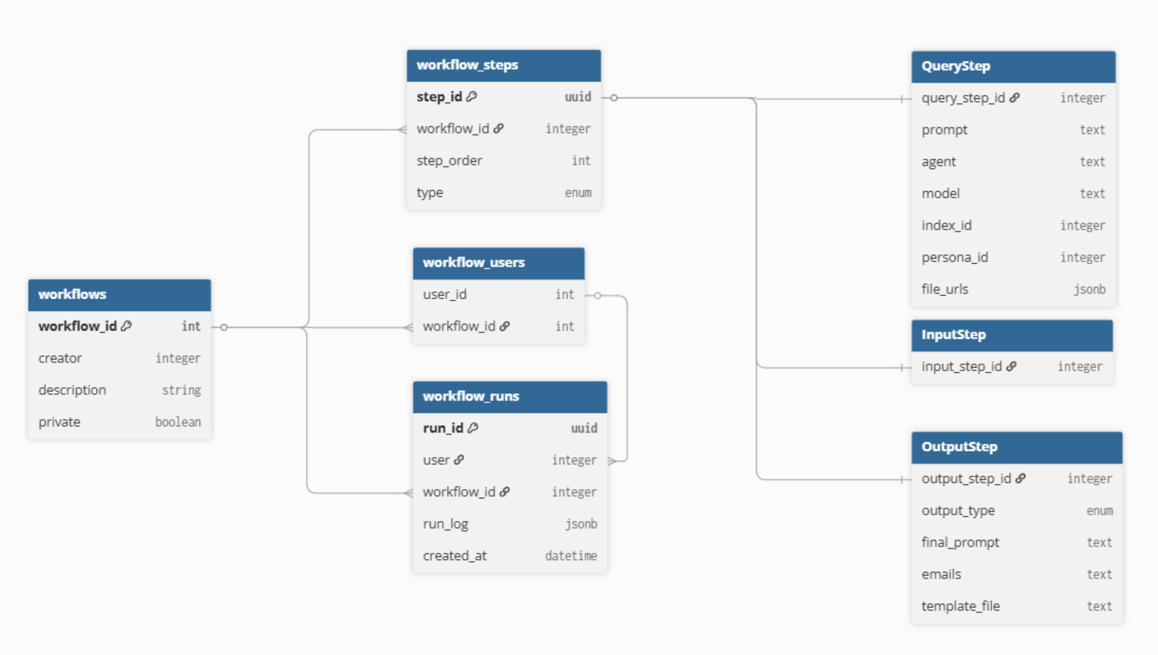

Figure 4. Diagram design for Workflows

The workflows are stored in the Workflow tables, but all individual steps are stored in the individual step tables. These reference a stepID in WorkflowSteps which in turn has a reference to the workflowID in Workflows.

4.3 Workflow Tools

The purpose of defining workflow tools is to allow users to use AI to perform certain actions in a controlled environment. The current tools that workflows provide are:

- Generating and running python code in a sandbox environment (via CodeInterpreter Agent)
- Searching the web (via WebSearch Agent)
- Generating Portable Document Format (PDF) and text documents from scratch
- Filling out a Word template document using the context from Query Steps

To implement the code generating and web search tools, Microsoft Azure's AI Foundry Agent Service was used that allows Azure users to call AI Agents hosted on their cloud infrastructure using a

Representational State Transfer (REST) Application Programming Interface (API). Two Agents are currently implemented: the Code Interpreter and the Web Search agents. The Code Interpreter agent can run its own code, allowing users to create graphs or analyze data. The Web Search agent infers whether the user requires the use of content from the internet, and searches the web for relevant information, as well as keeps track of the Uniform Resource Locators (URLs) from which it obtained its information.

The document generation and form filling tools take advantage of Word and PDF documents being represented as eXtensible Markup Language (XML) and Hypertext Markup Language (HTML) documents respectively. We used OpenAI's models to generate XML and HTML content that reflect the user's queries and context, using libraries to convert this output into the desired document format.

4.4 Workflow Publishing and Sharing

One important use case for workflows is for a power user (a user with technical ability and understanding) to create workflows that other users can use. To do this the ability to publish workflows to the public was created. However, certain security requirements must be met before publishing a workflow. A user must be the owner of a workflow and cannot publish a workflow that uses private indexes or personas. This prevents users from accidentally giving access to otherwise restricted resources. Additionally, published workflows can be read and run only for everyone except the owner. This means that, while users can run workflows with their own inputs, they cannot change any prompts or share any of their results.

4.5 Workflow Use Case 1

The workflow capability has been exercised for various practical use case applications. The first use case is creating a workflow from scratch that highlights basic functionality and the user interface. It entails editing query steps, running individual steps, and executing all steps sequentially to test the workflow. The example for the use case is to use a query step to generate a summary of photosynthesis and then modify the prompt to produce a different output, such as an interesting fact or a joke.

The steps performed are as follows:

1. **Creating a Workflow**: Create a new workflow named "Test Workflow." Upon creation, the workflow will automatically include a default query step with a pre-defined prompt. For example, the query step can be set to generate a short summary of photosynthesis.
2. **Editing a Query Step**: Edit the query step by clicking on the pen icon. This allows the user to modify the prompt, select models, specify indexes, personas, and upload files. These settings determine the output generated by the model when the query step is run.
3. **Running a Query Step**: Run the query step by clicking the play button, which offers two options: running all steps up to a specific query step or running only the selected step. For the example above, run only the first step, which produces a summary of photosynthesis as the output.

4. **Modifying the Prompt**: Change the prompt from generating a summary of photosynthesis to “Tell me something interesting.” After saving the changes, re-run the query step to have the model provide an interesting fact as the output.
5. **Adding a New Query Step**: Add a second query step with the prompt “Tell me a joke.” This step can be run individually, producing a joke as the output. Each query step operates independently unless the workflow is run in full, which combines all steps into a final output.
6. **Running the Entire Workflow**: Run entire workflow, including all query steps and the output step. This consolidates the outputs from each query step into a result, demonstrating how workflows can chain multiple steps together to produce a comprehensive output.

4.6 Workflow Use Case 2

The second use case showcases the capabilities of the **code interpreter agent** through a workflow designed to simulate coin flips and visualize the results. The steps performed are as follows:

1. **Introduction to the Code Interpreter Agent**: The code interpreter agent is a LLM capable of running Python code, creating files, analyzing data, and generating outputs like graphs or charts. It operates in a sandboxed environment, ensuring security by preventing internet access.
2. **Setting Up the Workflow**: As an example, define a workflow titled “Coin Flip Simulation” which includes a query step with a prompt instructing the code interpreter agent to simulate 100-coin flips using Python’s random.choice function. The prompt specifies counting the occurrences of heads and tails and plotting the results in a bar chart using Matplotlib.
3. **Selecting the Code Interpreter Agent**: Select the code interpreter agent for the query step, ensuring that the workflow will execute Python code to generate the desired output. Save the step to finalize the setup.
4. **Running the Workflow**: Run the query step to generate the bar chart.

4.7 Workflow Use Case 3

The third use case illustrates how workflows can chain multiple steps together to produce complex, structured outputs, such as a professionally formatted PDF containing both text and images. It uses the example of a fantasy story workflow that generates both text and images, culminating in a structured PDF output. The steps performed are as follows:

1. **Setting Up the Workflow**: Create a prompt instructing GPT-4.0 to generate a five-paragraph fantasy story titled “Rise of the Serpents” about a snake growing into a dragon. Create a second prompt that creates an image based on the story.
2. **Running Individual Query Steps**: The first step is to generate the text of the fantasy story and in the second step create an image based on the story’s theme. Note that generating images takes more time due to the computational intensity of the task. The duration of each step is displayed, allowing users to gauge the time required for specific tasks.

3. **Output Step Configuration**: Configure the output step to compile the results into a PDF file. The output step consolidates all prior query step outputs into a structured format, ensuring completeness and accuracy. The PDF file is generated based on the prompts and outputs from the workflow.
4. **Downloading and Reviewing the PDF**: After running the workflow, download the generated PDF file to review its contents. For this example, the PDF includes a five-paragraph fantasy story, the image of the snake transforming into a dragon, and other structured elements.
5. **Exploring Output Options**: The output steps can generate various file types, including PDF, DOCX, TXT, and even emails. This demonstrates the versatility of workflows in handling different output formats based on user needs.

4.8 Workflow Use Case 4

The fourth use case showcases the creation of a report comparing two models, **Cloud Three** and **GPT-4.0**, and generating a structured PDF output. The steps performed are as follows:

1. **Setting Up the Workflow**: Create a workflow designed to compare Cloud Three and GPT-4.0. The query steps gather information about the two models, summarize the findings, and generate a structured output. The final output will be a PDF file containing the comparison.
2. **Customizing the Output Step**: Modify the workflow’s output step to specify that the result should be saved as a PDF file. This ensures that the report will be formatted and downloadable as a document.
3. **Running the Workflow**: Run the workflow to execute the query steps sequentially. These steps include searching for information about the models, summarizing the data, and preparing the content for the final report. The workflow utilizes the code interpreter agent to process and organize data.
4. **Reviewing the Output**: On completion the workflow produces two PDF files, one titled "A Model Overview for Cloud Three and GPT-4.0: A Head-to-Head Breakdown" and a structured layout. The report features a **table of pros and cons** comparing the two models, a summary of the findings, and a final verdict. This highlights the ability to personalize the final output using the prompts in the workflow.
5. **Additional Features**: Workflows are flexible in generating structured outputs, such as tables and summaries. They can integrate data analysis, create graphs using the code interpreter agent, and include those visualizations in the final report. This demonstrates the capability to chain multiple steps together to produce a comprehensive and professional document.

4.9 Agents

Two main types of agents are available within workflows: the **code interpreter agent** and the **web search agent**. These agents significantly enhance the functionality of workflows by enabling specialized tasks.

1. **Code Interpreter Agent**:
   The code interpreter agent is a LLM capable of running Python code in a sandboxed environment. It can perform tasks such as creating files, analyzing data, and generating visualizations like graphs or charts. For example, the code interpreter agent can process uploaded files, such as CSVs, to analyze data and generate outputs like bar graphs or line charts. This makes it particularly useful for tasks like data analysis and report generation.
2. **Web Search Agent**:
   The web search agent uses an LLM to parse user prompts, search the web for relevant information, and return outputs based on the gathered data. For instance, it can be used as a web search agent to gather and summarize information from the internet such as finding the latest stock price.

Both agents operate within workflows to enhance automation and streamline complex tasks. The code interpreter agent is particularly suited for data-driven tasks, while the web search agent is ideal for gathering external information. These agents enable users to perform advanced operations, such as generating visualizations, analyzing datasets, and retrieving web-based information, all within the workflow framework.

4.10 Workflow Marketplace

The Workflow Marketplace provides a place for users to browse available Workflows, like the Index Marketplace or the Index browser in the Admin Console. It acts as the primary way to access any Workflows and serves as the “homepage” for the rest of the UI. Users can view their own workflows, and with the publishing feature, can view Workflows that others have published. Users also can query workflows by title, description, or user with the search feature. Additional features would include favorites and upvotes for workflows.

5 Summary

The report describes the key advancements of **Index Expansion** and **Workflows** developed and implemented with ParsonsGPT. Index Expansion introduces personal indexes and an index marketplace, whereas Workflows automate repetitive tasks and ensures consistency in outputs.

Personal indexes allow users to upload documents, which are converted into vector embeddings for intelligent search and interaction. The index marketplace fosters collaboration by enabling users to share, upvote, and comment on indexes. To reduce costs and improve scalability, the platform transitioned to Qdrant as the vector store, which supports efficient storage and querying of indexes. This shift allows broader access to indexing capabilities and enhances the overall performance of the system.

Workflows consist of query steps, where users specify prompts and models, and output steps, which define the deliverables such as PDFs, Word documents, or emails. The platform integrates agents like the **Code Interpreter Agent**, which can run Python code for data analysis and visualization, and the **Web Search Agent**, which retrieves information from the internet. These agents enhance the functionality of workflows, enabling tasks such as generating visualizations, analyzing datasets, and performing web-based research. Additionally, the **Workflow Marketplace** allows users to share, and access, published workflows, promoting collaboration and reproducibility.

Several practical use cases of workflows are highlighted, including generating summaries, simulating coin flips with visualizations, creating fantasy stories with images, and producing professional reports comparing models. These examples demonstrate the flexibility and power of workflows in automating complex tasks and generating structured outputs. Future enhancements to workflows include support for document template filling, structured outputs for API integrations, and expanded marketplace features such as voting and favoriting workflows.

In summary, ParsonsGPT represents a significant step forward in leveraging AI to improve productivity and efficiency within the organization. Its secure, scalable, and user-friendly design empowers employees to automate repetitive tasks, collaborate effectively, and focus on high-value activities. By integrating cutting-edge AI tools into its processes, Parsons Corporation is well-positioned to harness the transformative potential of generative AI.

6 References

Eisfeldt, Andrea L, Gregor Schubert, and Miao Ben Zhang. 2023. *Generative AI and Firm Values.* National Bureau of Economic Research. doi:10.3386/w31222.

Microsoft Corporation. 2023. *Azure.* 11 26. Accessed 11 26, 2023. https://azure.microsoft.com/en-us/.

OpenAI. 2024. *ChatGPT.* July 19. Accessed July 19, 2024. https://openai.com/chatgpt/.

—. 2022. *DALL·E 2.* April 6. Accessed July 19, 2024. https://openai.com/index/dall-e-2/.

—. 2023. *DALL·E 3.* October 15. Accessed July 19, 2024. https://openai.com/index/dall-e-3/.

—. 2021. *DALL·E: Creating images from text.* June 5. Accessed July 19, 2024. https://openai.com/index/dall-e/.

—. 2023. *OpenAI.* 11 26. Accessed 11 26, 2023. https://openai.com/.

Parsons Corporation. 2024. *ParsonsGPT: A Parsons private instance to Generative Pre-trained Transformer (GPT) models.* July 17. Accessed July 17, 2024. https://parsons365.sharepoint.com/sites/TechnologyCapabilities/chatgpt.

—. 2025. *Performance Testing.* July 28. Accessed August 8, 2025. https://bitbucket.parsons.us/projects/PGPT/repos/parsonsgpt-performance-testing/browse .

—. 2025. *Vector Store Comparison Summary.* July 1. Accessed August 8, 2025. https://confluence.parsons.com/pages/viewpage.action?pageId=649593433&spaceKey=PGPT&title=Vector%2BStore%2BComparison%2BSummary.

Rochford, Peter A, John Daly, and Nancy Smalley. 2024. *ParsonsGPT Beginner Guide.* User Guide, Centreville, VA: Parsons Corporation.

Rochford, Peter A. 2025. May 22. Accessed August 6, 2025. https://parsons365.sharepoint.com/sites/TechnologyCapabilities/aiatparsons/SitePages/ParsonsGPT-Documentation.aspx?csf=1&web=1&e=z9zeaM&CID=eff32535-65c7-44d7-8967-da1dc93f7924#parsonsgpt-knowledge-manager-reference-manual.

Wanna, Carly. 2023. "AI Study Says Stocks Already Pricing a Job-Replacement Premium." *Bloomberg.* August 2. Accessed September 3, 2023. https://www.bloomberg.com/news/articles/2023-08-02/firms-exposed-to-ai-job-streamlining-are-already-beating-the-market-study-says#xj4y7vzkg.

7 Acronyms

This section includes a list of acronyms used throughout the report for the reader's ease of reference.

AI	Artificial Intelligence
API	Application Programming Interface
CSV	Comma Separated Values
GovCloud	Government Cloud
GPT	Generative Pre-trained Transformer
HTML	Hypertext Markup Language
LLaMA	Large Language Model Meta AI
LLM	Large Language Model
MS	Microsoft
RAG	Retrieval Augmented Generation
REST	Representational State Transfer
UI	User Interface
URL	Uniform Resource Locator
XML	eXtensible Markup Language

8 Distribution List

Anthony Iasso, Chief Technology Officer (CTO), Engineered Services, Parsons Corporation, anthony.iasso@parsons.us

David Boyd, Vice President, Chief Technology Officer (CTO), Security and Mission Solutions, AI Solutions and Parsons Fellow, Parsons Corporation, david.boyd@parsons.us

James Quinn, Principal Software Engineer, Security and Mission Solutions, Parsons Corporation, james.quinn@parsons.us