

NASA Virtual Assistant

A Report submitted in partial fulfilment of the requirement for the award of
degree of

Bachelor of Technology

In

Electronics and Communication Engineering

Under the Supervision of

Dr. Neeru Rathi

By

KUNAL SHOME (203150022820)

KARAMVEER SINGH (21915002820)

LALIT BHATT (20715002820)



MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

C-4, Janakpuri, New Delhi-58

Affiliated to Guru Gobind Singh Indraprastha University, Delhi

June, 2023

DECLARATION

We, students of B.Tech (Electronics & Communication Engineering) hereby declare that the project work done on “NASA Virtual Assistant” submitted to Maharaja Surajmal Institute of Technology, Janakpuri Delhi in partial fulfilment of the requirement for the award of degree of Bachelor of Technology comprises of our original work and has not been submitted anywhere else for any other degree to the best of our knowledge.

Kunal Shome, 20315002820

Karamveer Singh, 21915002820

Lalit Bhatt, 20715002820

CERTIFICATE

This is to certify that the project work done on “NASA Virtual Assistant” submitted to Maharaja Surajmal Institute of Technology, Janakpuri Delhi by Kunal Shome, Karamveer Singh and Lalit Bhat in partial fulfillment of the requirement for the award of degree of Bachelor of Technology, is a bonafide work carried out by them under my supervision and guidance. This project work comprises of original work and has not been submitted anywhere else for any other degree to the best of my knowledge.

Signature of Supervisor

(Project Supervisor)

Signature of HOD

(HOD, ECE)

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to everyone who directly or indirectly contributed to the successful development of this project. Your support, encouragement, and guidance were instrumental in overcoming challenges and achieving our goals.

Firstly, we want to thank Dr. Neeru Rathi, the Head of the Department of Electronics and Communication Engineering and our mentor, for giving us the opportunity to undertake this project. Her trust in our abilities and her unwavering support motivated us throughout the entire process.

Her faith in our potential and her constant encouragement were crucial in driving our progress. Her guidance played a significant role in shaping the success of this project, and we are truly grateful for her mentorship. Her faith in our potential and her constant encouragement were crucial in driving our progress. Her guidance played a significant role in shaping the success of this project, and we are truly grateful for her mentorship.

Beyond these specific individuals, we want to acknowledge the collective effort that went into this project. We are thankful to the Almighty for giving us the strength and resilience to complete the project on time. We are also deeply grateful for the unwavering support, understanding, and encouragement of our parents, friends, and well-wishers.

Their contributions were invaluable, and their presence sustained us throughout the journey. This project stands as a testament to the power of collaboration, encouragement, and shared determination. Each individual mentioned has played an integral role in its realization, and we are genuinely thankful for all the contributions.

Kunal Shome, 20315002820

Karamveer Singh, 21915002820

Lalit Bhatt, 20715002820

INDEX

ABSTRACT	i
LIST OF FIGURES	ii
CHAPTER-1 INTRODUCTION	1
1.1 Context	1
1.2 Need for the Virtual Assistant.....	2
1.3 Virtual Assistant Review.....	2
1.4 Software Used	3
CHAPTER-2 PRE-REQUISITE KNOWLEDGE	8
2.1 Open AI Assistant	4
2.2 Open AI API.....	9
2.3 Botpress Components.....	11
CHAPTER-3 CREATING CHATBOT	19
3.1 Data Extraction.....	19
3.2 Open AI Assistant	20
3.3 Botpress Studio	21
3.4 Styling the Bot.....	24
3.5 Knowledge Base	Error! Bookmark not defined.
CHAPTER-4 IMPLEMENTATION	31
4.1 Error! Bookmark not defined.	
4.2 Creating Template Website	Error! Bookmark not defined.
4.2 Creating Template Website	Error! Bookmark not defined.
CHAPTER-5 CONCLUSION	40
REFERENCES	41

ABSTRACT

This report introduces the "NASA Virtual Assistant," a groundbreaking project powered by OpenAI and seamlessly deployed with the capabilities of Botpress. Serving as a pivotal addition to the NASA digital ecosystem, the chatbot addresses the need for an interactive and intelligent assistant on the official NASA website. Developed to enhance user experience and provide real-time assistance, the NASA Virtual Assistant represents a collaborative effort between OpenAI and NASA, offering a dynamic and user-friendly interaction. This project fills a crucial void by introducing a dedicated chatbot to the NASA website, providing instant, personalized, and intuitive support for users navigating the vast realm of space-related inquiries.

The report explores the necessity for the virtual assistant in the evolving landscape of digital engagement, emphasizing the role of advanced artificial intelligence integrated with Botpress in redefining the standard for interactive engagement on the NASA website. It also delves into the broader context of virtual assistant applications in space agencies, highlighting potential benefits in streamlining communication, providing information, and engaging users interactively.

Furthermore, the report details the software used in the implementation of the project, with a focus on Visual Studio Code (VS Code) for website styling and development, OpenAI for advanced natural language processing capabilities, and Botpress as the open-source conversational platform central to the chatbot's creation. The role of each software component is elucidated, emphasizing their contributions to the success and user experience of the NASA Virtual Assistant.

Overall, this report serves as a comprehensive overview of the NASA Virtual Assistant project, from its inception to its technological components, underscoring the transformative power of artificial intelligence in advancing the frontiers of knowledge and user interaction in the realm of space exploration.

LIST OF FIGURES

Figure 1: Open AI Assistant Objects

Figure 2: Workflows

Figure 3: Creating Knowledge Base

Figure 4: Botpress Studio

Figure 5: Configure Bot

Figure 6: Emulator Gate

Figure 7: Handle Responses

Figure 8: Index

Figure 9: Knowledge Base Data

Figure 10: Chatbot “Donna”

Figure 11: Template Website with Working Chatbot

LIST OF ABBREVIATIONS

1. **NASA**: National Aeronautics and Space Administration
2. **AI**: Artificial Intelligence
3. **GPT-3**: Generative Pre-trained Transformer 3
4. **API**: Application Programming Interface
5. **VS Code**: Visual Studio Code
6. **ML**: Machine Learning
7. **CSS**: Cascading Style Sheets
8. **HTML**: Hypertext Markup Language
9. **IDE**: Integrated Development Environment

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION TO THE PROJECT

Introducing the groundbreaking project, the "NASA Virtual Assistant," a cutting-edge creation by OpenAI, seamlessly deployed with the robust capabilities of Botpress. This innovative chatbot stands as a pivotal addition to the NASA digital ecosystem, addressing the need for an interactive and intelligent assistant on the official NASA website.

In a concerted effort to enhance user experience and provide real-time assistance, OpenAI collaborated with NASA to bring forth this state-of-the-art virtual assistant. Its integration with Botpress ensures a dynamic and user-friendly interaction, allowing visitors to navigate through the vast realm of NASA's information effortlessly.

The inception of the NASA Virtual Assistant was necessitated by the absence of a dedicated chatbot on the NASA website. Recognizing the importance of instant and personalized support for users exploring the depths of space-related inquiries, OpenAI embarked on this project to bridge the gap and augment the accessibility of NASA's wealth of information.

As users engage with the NASA Virtual Assistant, they can anticipate a seamless experience, tapping into a wealth of knowledge and resources at their fingertips. This project not only represents a technological milestone but also underlines the commitment to fostering a more interactive and informative online space exploration experience for enthusiasts and researchers alike. Welcome to a new era of exploration and discovery facilitated by the NASA Virtual Assistant—a testament to the power of artificial intelligence in advancing the frontiers of knowledge.

1.2 NEED FOR THE VIRTUAL ASSISTANT

In the ever-evolving landscape of digital engagement, the absence of a dedicated virtual assistant on the NASA website left a noticeable void in user interaction. Navigating the vast cosmos of information housed within NASA's web domain proved to be a challenging endeavor for users seeking immediate, personalized, and intuitive assistance. Recognizing this gap, the "NASA Virtual Assistant" project by OpenAI emerged as a strategic response to the imperative need for a dynamic chatbot solution. By integrating advanced artificial intelligence with the user-friendly framework of Botpress, the NASA Virtual Assistant not only fills the void but also redefines the standard for interactive engagement on the NASA website. This initiative marks a crucial step towards providing users with a seamless, informative, and responsive gateway to the wonders of space exploration.

1.3 VIRTUAL ASSISTANT REVIEW

Research in the application of virtual assistants in space agencies has highlighted their potential to streamline communication, provide information, and engage users in a more interactive manner. Existing studies may showcase the benefits of AI-driven assistants in handling complex queries related to space missions, satellite data analysis, and astronomical research.

Understanding the advancements in AI and chatbot technologies is crucial for assessing the feasibility and effectiveness of integrating a virtual assistant into a space agency's digital infrastructure. Literature in this area may delve into natural language processing, machine learning algorithms, and user experience design, providing insights into optimizing the performance of virtual assistants.

The success of a virtual assistant project relies heavily on user experience and engagement. Research in this domain may investigate user perceptions, preferences, and the impact of virtual assistants on enhancing accessibility to information. Evaluating the effectiveness of chatbot deployments, especially in scientific domains, can shed light on the potential benefits for space exploration.

Exploring case studies and real-world implementations of virtual assistants in scientific organizations, or analogous domains, can offer valuable lessons and benchmarks for the NASA Virtual Assistant project. Understanding challenges faced, solutions implemented, and outcomes achieved will contribute to informed decision-making and project optimization.

1.4 SOFTWARE USED:

VS CODE

Visual Studio Code (VS Code) stands as a versatile and widely embraced source-code editor renowned for its robust features, cross-platform compatibility, and an extensive array of extensions. Developed by Microsoft, VS Code offers a lightweight yet powerful environment for coding across various programming languages. Its intuitive interface, integrated debugging capabilities, and support for version control systems make it a preferred choice among developers for efficient code development and collaboration.

It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE. It is free for private or commercial use. Some of its features are

- i. Visual Studio Code can be used with a variety of programming languages, including C#, Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran. It is based on the Electron framework, which is used to develop Node.js web applications that run on the Blink layout engine.
- ii. It includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.
- iii. Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports many

programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

- iv. Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where users can access version control settings and view changes made to the current project. To use the feature, Visual Studio Code must be linked to any supported version control system (Git, Apache Subversion, Perforce, etc.). This allows users to create repositories as well as to make push and pull requests directly from the Visual Studio Code program.
- v. Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.
- vi. Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled. Due to the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected.
- vii. Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, time travel debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

In the implementation of the virtual assistant project, VS Code played a pivotal role in shaping the visual aesthetics and styling of the associated website. Leveraging the software's rich set of extensions and integrated tools, we were able to streamline the design process, ensuring a seamless and visually appealing user interface. The flexibility and efficiency of VS Code facilitated the creation of a well-structured and visually cohesive website, contributing significantly to the overall success and user experience of the virtual assistant project.

OPEN AI

OpenAI stands at the forefront of artificial intelligence research, pioneering the development of diverse models that have reshaped the landscape of natural language processing. Among its flagship models is GPT-3 (Generative Pre-trained Transformer 3), a state-of-the-art language model that demonstrates unprecedented proficiency in understanding and generating human-like text. GPT-3 has been leveraged across various applications, from creative writing and content generation to language translation and code completion, showcasing its versatility and adaptability.

One remarkable aspect of OpenAI's offerings is the OpenAI API, which empowers developers to harness the capabilities of GPT-3 and integrate them seamlessly into their own applications. With the OpenAI API, users can access the power of GPT-3 by generating human-like text, answering questions, creating conversational agents, and more. What sets OpenAI apart is its commitment to democratizing access to artificial intelligence.

The platform enables users to customize and fine-tune their own virtual assistants using the OpenAI API, creating tailored solutions that align with their specific needs. This revolutionary approach allows developers to craft unique AI-powered applications by generating API keys, providing unprecedented flexibility in building intelligent systems that fetch and process information with remarkable nuance and contextual understanding. OpenAI's commitment to advancing the field of AI is epitomized by its user-friendly API, fostering a new era of creative possibilities and innovative applications in natural language processing.

BOTPRESS

Botpress is an open-source conversational platform designed to simplify and streamline the development of chatbots and virtual assistants. With a focus on flexibility and ease of use, Botpress empowers developers to create sophisticated conversational experiences through a visual interface and modular architecture.

The Botpress API is a RESTful set of HTTP endpoints that allow you to create, deploy, and run chatbots on the Botpress Cloud([opens in a new tab](#)). It can be used to create and manage bots, handle conversations and messages, as well as to manage their content, users, and configuration. The base URL of the Botpress Cloud API is: <https://api.botpress.cloud>

The API endpoints will expect the Content-type: application/json HTTP header to be present in the request and the request body (if any) to be in JSON format, and will send back the same header in the response and return a JSON response body as well.

Its robust set of features includes natural language understanding, conversation flow management, and integrations with various messaging channels, making it suitable for a wide range of applications. Developers appreciate Botpress for its extensibility, allowing the integration of custom modules and plugins, and its support for advanced functionalities like machine learning and artificial intelligence.

Whether used for customer support, internal communication, or automation of tasks, Botpress stands as a versatile tool that facilitates the creation of intelligent and interactive conversational interfaces.

KAGGLE

Kaggle is a renowned online platform that serves as a hub for data scientists, machine learning enthusiasts, and researchers to collaborate, compete, and showcase their expertise in the field of data science. Acquired by Google in 2017, Kaggle provides a diverse range of datasets, competitions, and kernels (code notebooks) that enable users to explore, analyze, and model data. The platform fosters a vibrant community by hosting data science competitions that challenge participants to tackle real-world problems and find innovative solutions.

Kaggle also offers extensive educational resources, forums, and collaborative tools, making it an invaluable resource for individuals and organizations looking to leverage data for insights and predictions. Whether for honing data science skills, participating in competitions, or accessing valuable datasets, Kaggle stands as a dynamic and collaborative space at the forefront of the data science community.

By tapping into Kaggle's diverse collection of datasets spanning various domains, we ensured that our virtual assistant could provide users with accurate and contextually relevant data. The seamless integration of Kaggle datasets into our bot's knowledge base not only facilitated real-time information retrieval but also enriched the user experience by offering a comprehensive and dynamic source of data. Kaggle's platform played a pivotal role in empowering our bot to stay informed and adaptable, showcasing the collaborative spirit of data science in action.

CHAPTER 2: PRE-REQUISITE KNOWLEDGE

2.1 OPEN AI ASSISTANT

The Assistants API automatically manages the context window such that you never exceed the model's context length. Once the size of the Messages in a Thread exceeds the context window of the model, the Thread will attempt to include as many messages as possible that fit in the context window and drop the oldest messages. Note that this truncation strategy will evolve over time to become more sophisticated. Currently, the Assistant will include the maximum number of messages that fit in the context length. We plan to explore the ability for you to control the input / output token count beyond the model you select, as well as the ability to automatically generate summaries of the previous messages and pass that as context. If your use case requires a more advanced level of control, you can manually generate summaries and control context with the Chat Completion API.

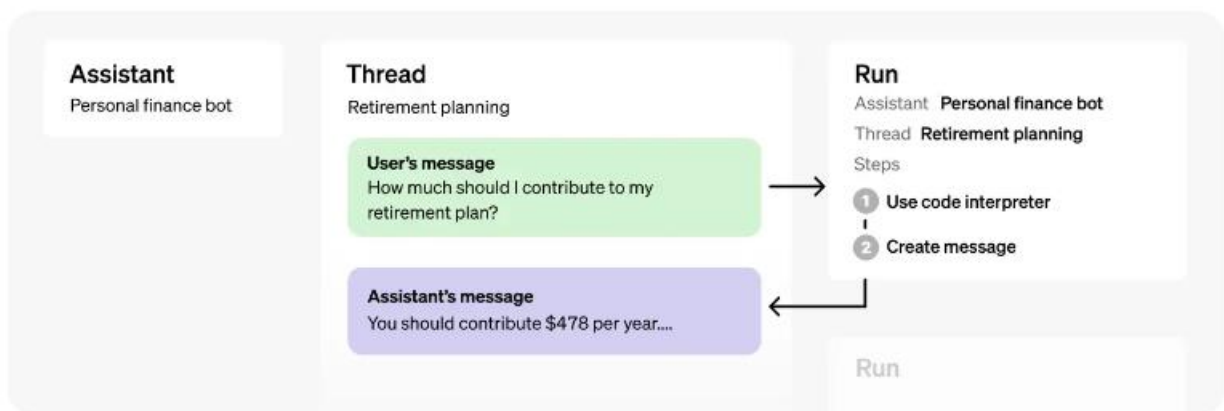


Figure 1: Open AI Assistant Objects

Threads and Messages represent a conversation session between an Assistant and a user. There is no limit to the number of Messages you can store in a Thread. Once the size of the Messages exceeds the context window of the model, the Thread will attempt to include as many messages as possible that fit in the context window and drop the oldest messages.

Messages can contain text, images, or files. At the moment, user-created Messages cannot contain image files but we plan to add support for this in the future. Messages also have the same file size and token limits as Assistants (512 MB file size limit and 2,000,000 token limit).

An Assistant represents an entity that can be configured to respond to users' Messages using several parameters like:

- i. Instructions: how the Assistant and model should behave or respond
- ii. Model: you can specify any GPT-3.5 or GPT-4 models, including fine-tuned models. The Retrieval tool requires gpt-3.5-turbo-1106 and gpt-4-1106-preview models.
- iii. Tools: the API supports Code Interpreter and Retrieval that are built and hosted by OpenAI.
- iv. Functions: the API allows you to define custom function signatures, with similar behavior as our function calling feature.

2.2 OPEN AI API

The OpenAI API can be applied to virtually any task. It offers a range of models with different capabilities and price points, as well as the ability to fine-tune custom models.

OpenAI offers a spectrum of APIs, each designed to cater to specific needs and applications, showcasing the organization's commitment to advancing artificial intelligence across various domains. The GPT-3 API, the most prominent among them, empowers developers with unprecedented natural language processing capabilities. It enables the generation of human-like text, answering questions, writing code, creating conversational agents, and much more. The GPT-3 API's

versatility makes it a cornerstone for applications ranging from content generation to interactive virtual assistants.

Additionally, OpenAI provides the OpenAI Codex API, which is built on the Codex model, designed specifically for code generation. This API facilitates the creation of innovative applications in software development, automating coding tasks, and generating code snippets based on natural language prompts.

OpenAI also offers the OpenAI DALL-E API, which leverages the DALL-E model to generate diverse and creative images from textual descriptions. This API finds applications in visual arts, design, and creative content generation.

Furthermore, OpenAI provides the OpenAI CLIP API, based on the CLIP (Contrastive Language-Image Pretraining) model, enabling powerful interactions between text and images. This API finds applications in image recognition, content retrieval, and multimodal understanding, bridging the gap between language and vision.

OpenAI's suite of APIs reflects a commitment to democratizing access to advanced AI capabilities, offering developers the tools to integrate cutting-edge models into a wide array of applications, from natural language understanding and code generation to image processing and beyond. The diverse range of OpenAI APIs opens up new possibilities for innovation across industries, marking a significant step forward in the democratization and accessibility of artificial intelligence.

2.3 BOTPRESS COMPONENTS

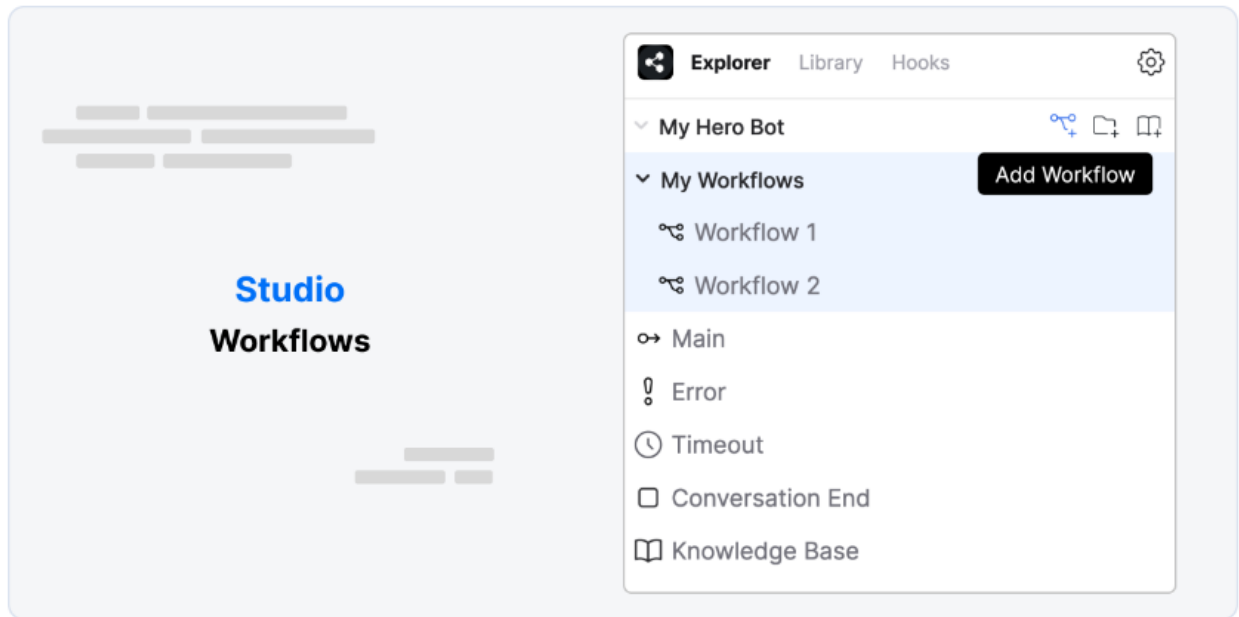


Figure 2: Workflows

I. WORK FLOWS

A flow allows you to break down a complex chatbot into multiple smaller flows. Breaking down the chatbot into multiple flows makes it easier to maintain, and you can re-use these flows when building other flows or even other chatbots.

A workflow always starts at its Entry Node. At the beginning of each conversation, the Start Node of the Main flow of the Global Topic is the first to be executed. When the conversation reaches a node, Botpress will queue the active node's instructions and execute them in the order they were added to the node.

The flow is event-based and non-blocking by default, meaning it will execute all the nodes and instructions until it needs to wait. In other words, once the Entry Node is processed, the flow will continue to proceed to the next node in the flow until it reaches the very end of the flow or until a node needs to “wait” for user input.

i. Global Flows

Global flows in Botpress are essential components of the chatbot's functionality and cannot be deleted. These flows serve as the foundation for handling various important aspects of the chatbot's behavior and interactions.

While you cannot delete global flows, you can modify and customize them to suit your specific needs. You can add new actions, adjust the workflow logic, and incorporate additional functionality within the existing global flows. This allows you to extend the capabilities of the chatbot while maintaining the essential system-level functionality provided by the global flows.

ii. Main

This is the primary conversation path that the chatbot follows in order to achieve its main purpose.

For example, if the chatbot is designed to provide customer service for a particular product, the Main Flow would include steps to gather information about the customer's issue, provide potential solutions, and ultimately resolve the problem.

The Main Flow is typically the most detailed and complex flow in the chatbot's design, as it needs to account for a wide range of possible inputs and scenarios.

iii. Error

The Error Flow is designed to handle unexpected failures that may occur during the conversation. It will be triggered when there's technical issues including irrecoverable errors in the bot.

For example, if you have an Execute Code card with invalid Javascript code like workflow. = 'some value', or custom choices for Captures or Carousels in the wrong format, the Error flow shall be triggered.

iv. Timeout

The Timeout Flow is triggered when the chatbot doesn't receive a response from the user for a certain period of time.

This can happen if the user gets distracted with something else or if there is a delay in their internet connection.

The Timeout Flow is important because it helps prevent the chatbot from appearing unresponsive or disconnected, which can lead to user frustration and disengagement. Learn more about the Inactivity Timeout setting.

The default timeout is 30 minutes. So if the user doesn't respond for 30 minutes, this flow will be triggered.

v. Conversation End

The Conversation End Flow is designed to wrap up the conversation in a way that feels natural and satisfying to the user.

Depending on the context and purpose of the chatbot, this may involve thanking the user for their input and inviting them to provide feedback or continue the conversation later, or it may involve providing specific next steps for the user to take.

The Conversation End Flow is an important part of the chatbot design, as it helps create a positive user experience and encourages users to return in the future.

vi. Custom Workflow

It's important to note that you can create and manage custom workflows alongside the global flows to implement specific business logic and custom behavior. This way, you can enhance the chatbot's functionality without compromising the integrity of the global flows.

By clicking on the "Add Workflow" icon located at the top of the Explorer Menu, you gain the ability to create custom workflows for your chatbot. Workflows are a crucial aspect of chatbot development as they define the logic and flow of conversations between the bot and its users.

Creating custom workflows allows you to tailor the behavior of your chatbot according to your specific requirements and use cases.

Here are some key benefits and possibilities of creating custom workflows:

- a) **Designing Conversational Flows:** Workflows enable you to design conversational flows that guide the interaction between the chatbot and users. You can define the sequence of messages, questions, and responses to create engaging and meaningful conversations.
- b) **Implementing Business Logic:** Custom workflows empower you to incorporate complex business logic into your chatbot. You can define conditions, triggers, and actions based on user inputs or specific events to provide dynamic and context-aware responses.
- c) **Integration with External Systems:** Workflows allow you to integrate your chatbot with external systems and APIs. You can utilize custom workflows to fetch data from databases, invoke web services, or perform any other necessary operations to enrich the user experience and provide relevant information.
- d) **Handling User Input:** With custom workflows, you can capture and process user input in a structured manner. You can define prompts,

validate input, and handle different scenarios based on user responses, ensuring a smooth and interactive conversation flow.

- e) **Implementing Advanced Features:** Custom workflows enable you to implement advanced features such as user authentication, session management, context preservation, and more. These features enhance the functionality and capabilities of your chatbot, allowing for a more sophisticated user experience.

vii. Execute Workflow

After building a few flows, you'll quickly notice that there are some common patterns that you find yourself implementing over and over. These can be seen as reusable components between multiple flows. So, you can create Folders to categorise your workflows.

Once you have created your workflow, right-click in the Flow Editor, hover over the Execute Workflow option and select your workflow.

II. KNOWLEDGE BASE

The Botpress Knowledge Base is a powerful tool that allows users to manage and access organizational knowledge in one central location. It has been designed to replace the traditional Q&A system and enable users to upload various sources of information, including PDFs, Text Document, and website URLs.

You need to enable the Knowledge Base Agent to use the Knowledge Base feature in Botpress. Click on the Agent Tab in the Explorer on the left panel of the studio. You will find the Knowledge Base Agent in the list of available agents. Click on the Enable button to enable the Knowledge Base Agent.

To create a Knowledge Base, click on the Add Knowledge Base Icon in the Explorer on the left panel of the studio. You can create folders and organize your knowledge bases as per your preferences.

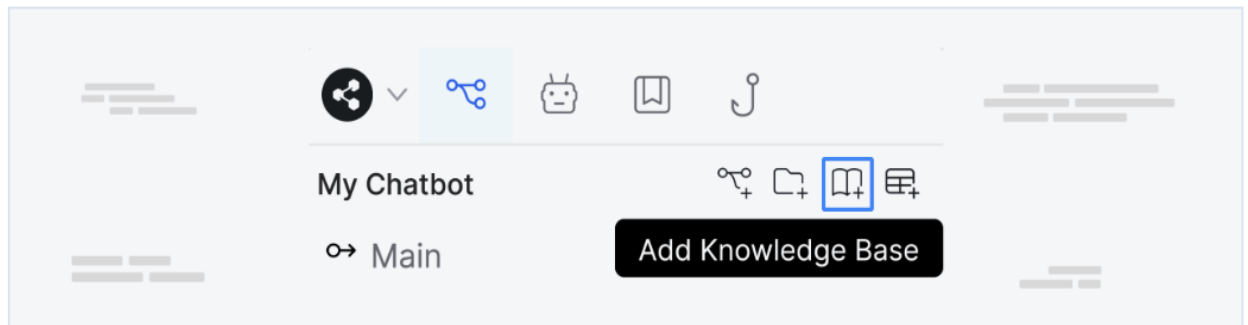


Figure 3: Creating Knowledge Base

This helps the system understand what type of questions to expect and gives it a better chance of finding the right answers for your users.

III. NODES

Nodes are the primary units of conversational logic of your chatbot. A node generally transitions to another node or flow. You can think of a node as a step in a conversation.

A single node can have multiple content types, instructions, and transitions. When there aren't any more transitions, the conversation ends.

Types of Nodes:

i. Start Node

The Start Node is a specialized Entry Node that is only available in the main flow of a Botpress bot.

It serves as the starting point for the conversation and can only execute transitions to other standard nodes.

ii. Entry Node

Every workflow in your bot, except the main flow, starts with an Entry node. The Entry node is the starting point for each individual workflow and can only execute transitions to other standard nodes within that workflow. It is used to define the entry point for the workflow and the initial conditions that must be met for the workflow to start executing.

iii. Exit Node

Similar to the Entry node, every workflow in your bot, except the main flow, should end with an Exit node. The Exit node is the final node in a workflow and can only receive transitions from other standard nodes within that workflow. It is used to define the exit point for the workflow and specify the conditions that must be met for the workflow to end. The Exit node typically contains cleanup or finalization logic, such as sending a message to the user or updating a database, before the workflow is completed.

iv. Standard Node

The Standard node is a versatile building block in your bot's workflow. It allows you to add both instructions and transitions to your flow, and it can be used to represent a variety of steps in the conversation with the user.

You can add instructions to the Standard node to define what the bot should do at that point in the conversation, such as sending a message or asking the user for input. You can also add transitions to the Standard node to specify the conditions under which the conversation should proceed to the next node.

Standard nodes can be connected to other nodes within the same workflow using transitions, allowing you to create complex conversation flows that adapt to the user's input and actions. They are essential building blocks for creating conversational AI experiences that are engaging and useful for your users.

v. End Node

The End Node is a unique node in your bot's Main Flow. Its purpose is to clear the conversation session and reset the bot to its initial state once it is reached. When the End Node is executed, it will erase all variables and user data associated with the conversation and set the cursor back to the beginning of the Main Flow.

The End Node is typically used when you want to reset the conversation with the user and start a new session from scratch. It is particularly useful for bots that handle sensitive or personal information, where it is important to ensure that previous session data is not accessible to subsequent users.

Note that the End Node is only available in the Main Flow of your bot and cannot be used in other workflows.

CHAPTER 3: CREATING CHATBOT

3.1 DATA EXTRACTION

The creation of the chatbot for NASA was a nuanced and multifaceted process that sought to seamlessly blend advanced technology with the imperative need for real-time and accurate data. The initial building blocks of the chatbot were constructed using Botpress, a versatile and powerful platform designed for chatbot development. This framework allowed for the structuring of the chatbot into manageable flows, facilitating efficient maintenance and reusability.

However, to elevate the chatbot's capabilities and ensure its knowledge base remained current and accurate, a crucial step involved the extraction of data from Kaggle and NASA's official website. This decision was driven by the recognition that while ChatGPT, the underlying language model, is an immensely powerful tool, its training data extends only up to September 2019. In the dynamic and rapidly evolving field of space exploration, relying solely on pre-existing knowledge might lead to gaps in information, especially concerning the latest discoveries, mission updates, and scientific breakthroughs.

Kaggle, known for its diverse and constantly updated datasets, emerged as an invaluable resource in this context. Leveraging Kaggle allowed the chatbot to tap into a wealth of real-time information, including astronomical data, mission details, and the latest findings in the realm of space science. Simultaneously, extracting data from NASA's official website ensured that the chatbot incorporated authoritative and current information directly from the primary source.

This comprehensive approach, integrating the robust capabilities of ChatGPT with up-to-date data from Kaggle and NASA, positioned the chatbot as a reliable and informed conversational agent. The amalgamation of advanced language processing with the latest and most accurate information not only enriched the user experience but also showcased a commitment to providing users with the most current insights into the ever-evolving landscape of space exploration. The synergy between advanced AI models and real-time data extraction exemplifies a forward-thinking approach to chatbot development, ensuring that the virtual assistant remains a valuable resource for those seeking the latest

and most accurate information about NASA's activities and the broader field of space science.

3.2 OPEN AI ASSISTANTS

In this instance, we've utilized the OpenAI Assistant API, providing input lines to the model to enable it to respond to a variety of user queries.

We have fed the following lines to the model:

“You are an educational chatbot about space and NASA, having a personality of a nerdy female teacher.

You send an emoji in each response bubble.

You respond in English only.

Respond in less than 80 words.

1. If the input is related to NASA, give an output within 3 sentences (excluding personality texts).

2. If the input related to space but not NASA, give an output in 3 sentences (excluding personality texts). Also express that this information is not available with NASA, but it would be available with ISRO in a humorous way.

3. If the input is not related to space or NASA, give an output saying 'I can only answer questions about space and NASA!' or something similar.

After giving answer to the user's question, leave a line and then you give a random 'did you know' fact about space starting with 'Fun Fact: ' ”

OpenAI typically provides interactive environments or interfaces, often referred to as "playgrounds," for users to experiment with various models. In such settings, users can input prompts or queries and observe real-time responses generated by the OpenAI models.

These environments are designed to offer a user-friendly and interactive experience, allowing users to explore the capabilities of the OpenAI Assistant API, experiment with different prompts, and understand how the model processes and generates responses.

3.3 BOTPRESS STUDIO

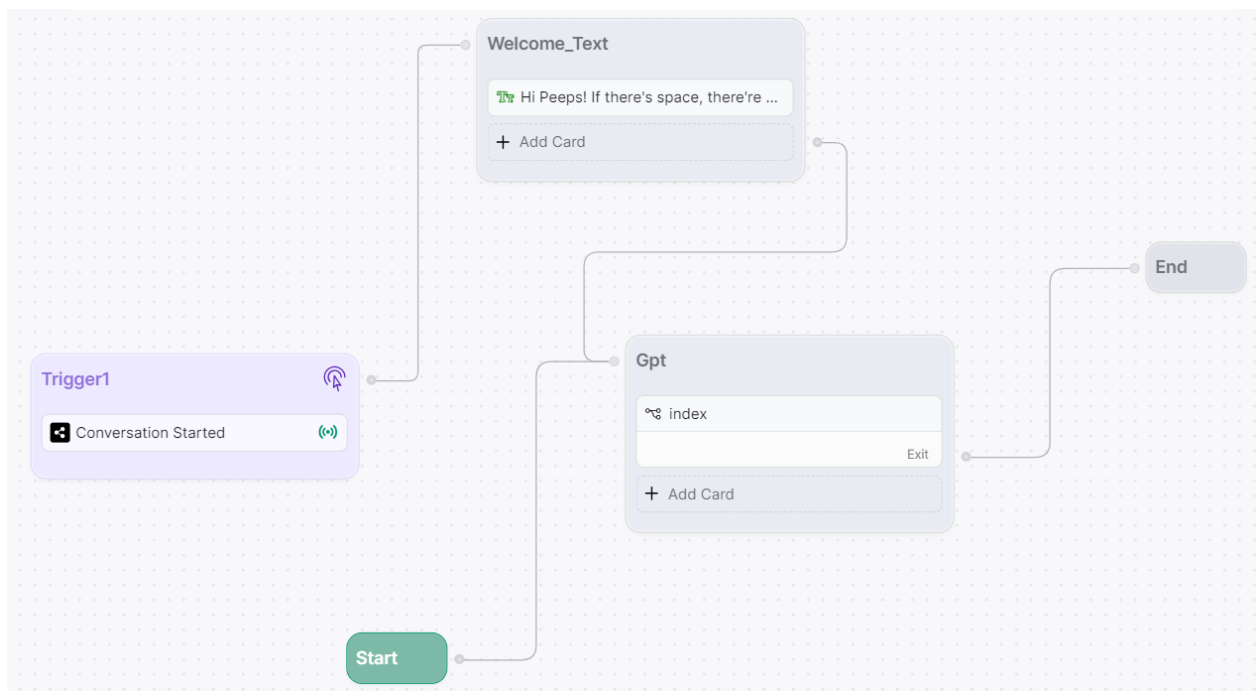


Figure 4: Botpress Studio

Within the primary section, we formulate the core workflow of our chatbot, determining how the bot will interact in response to various user inputs. To initiate the conversation, we link a trigger to the starting point, connecting it to a welcome message that the user sees at the beginning of the interaction.

Following this, we establish a connection to the GPT API immediately after the start option. This integration enables the bot to retrieve and process the information provided by the user, generating a relevant response that is then communicated back to the user.

Within the GPT API note, additional components include configuring the bot, setting up the emulatorGate, managing response handling, and establishing the index.

I. Configure Bot

It initiates from the entry point of the input and then diverges into various directions based on predefined standards, determining the corresponding outputs according to user interactions, and concludes.

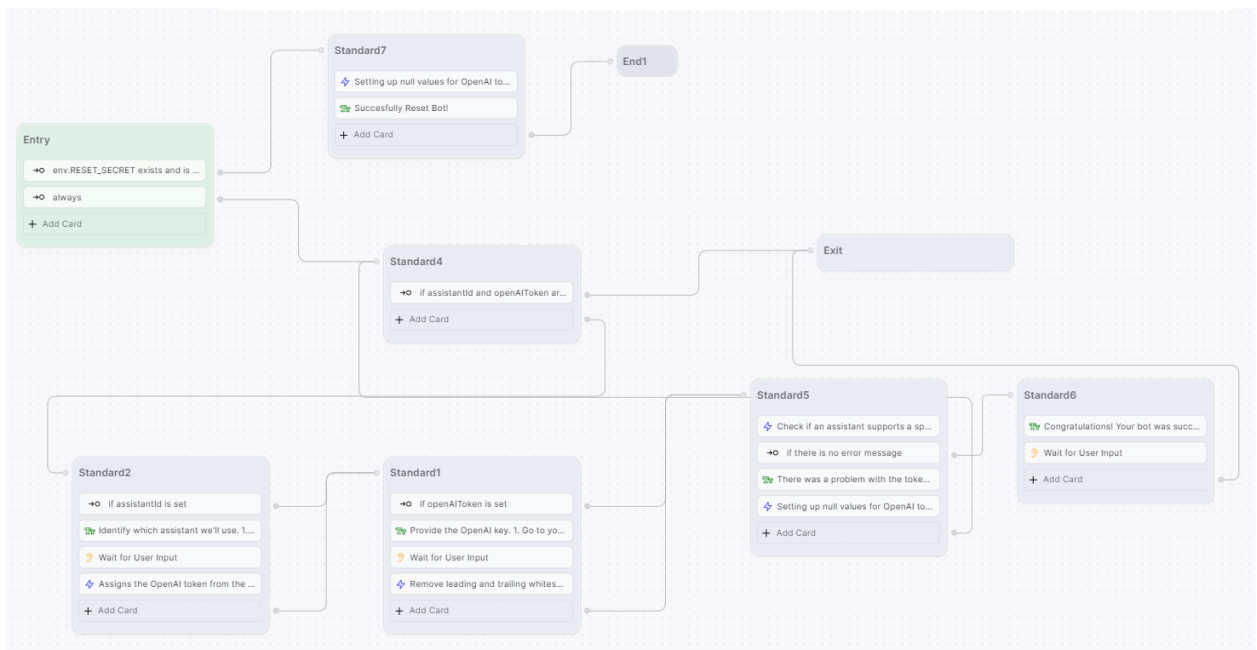


Figure 5: Configure Bot

II. Emulator Gate

It ensures that all responses provided to the user maintain the correct sequence without any disruption in order.

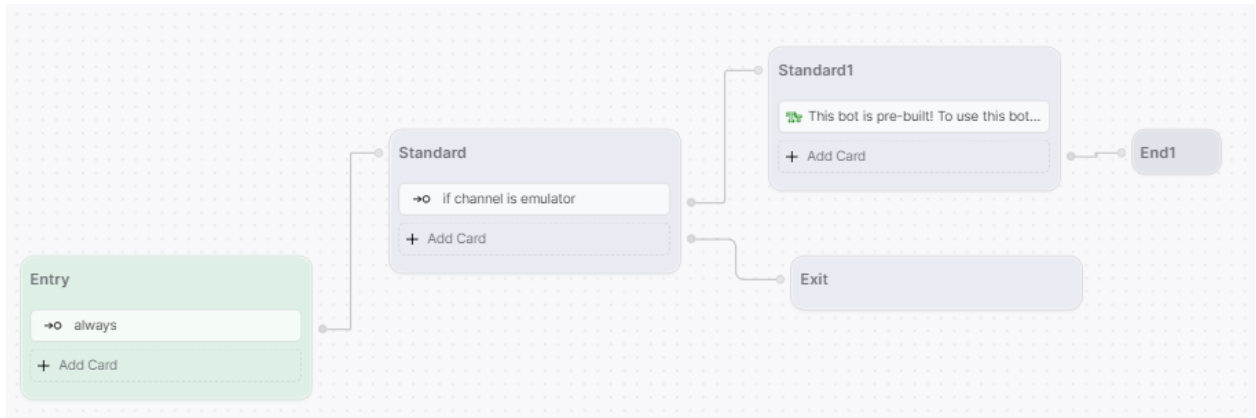


Figure 6: Emulator Gate

III. Handle Responses

It handles and the responses of the user and the bot.

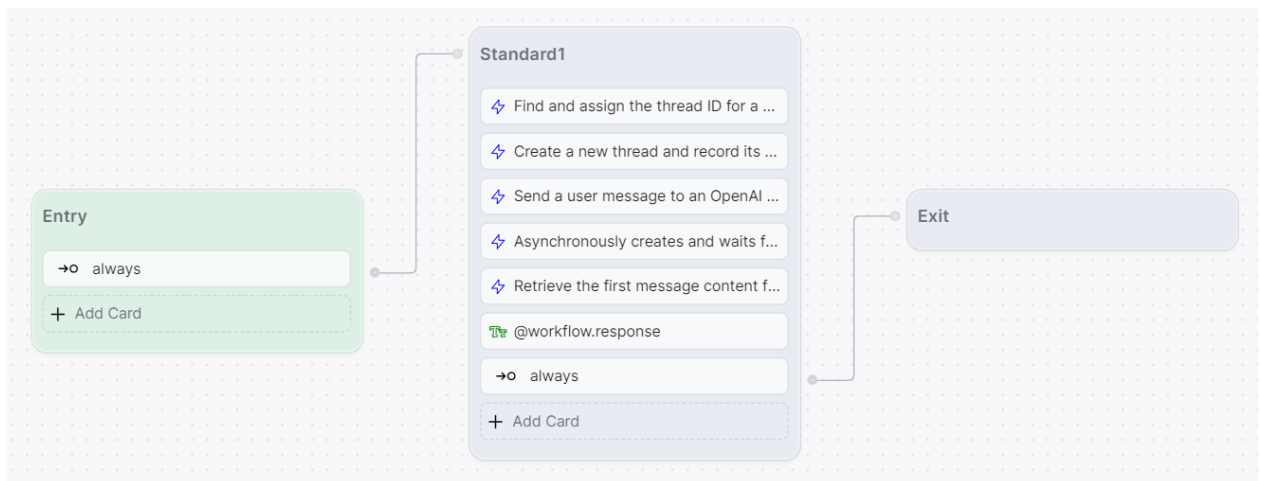


Figure 7: Handle Responses

IV. Index

Index node makes sure that all the other nodes in the workflow are in sync and the responses are coming.

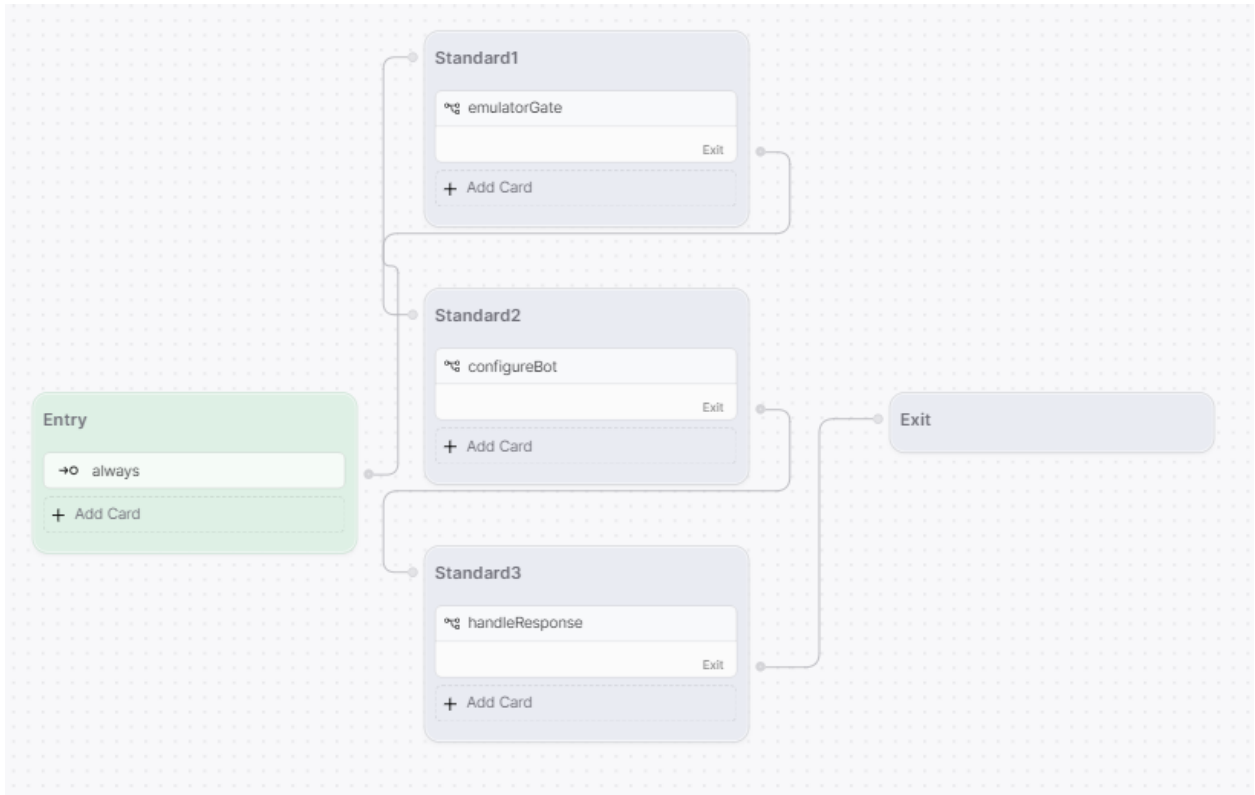


Figure 8: Index

3.4 STYLING THE BOT

The seamless functionality and visual appeal of our chatbot have been achieved through meticulous website styling using CSS. Our dedicated team has worked diligently to design an intuitive and user-friendly interface, ensuring an optimal user experience.

The deployment process has been executed with precision, making the chatbot accessible to users in a responsive and visually engaging manner.

The following code has been used to style the chatbot:

```
/* This CSS rule sets the default font size of the entire document */
html {
    font-size: 16px;
}

/* These CSS rules sets the text direction of the paragraph (RTL or
LTR) */
p,
.bpw-from-bot > div,
#input-message {
    direction: ltr;
}

/* This rule changes the color of the text in the input message */
#input-message {
    color: #ffffff;
}

/* Changes the color of the new message indicator */
.bpw-new-messages-indicator {
    background-color: #000e14;
    color: #ffffff;
}

/* Styling for the chat header container */
.bpw-header-container {
    margin: 10px;
    background: #000000;
    border-radius: 10px;
    position: relative;
    overflow: hidden;
    border: 1px;
    border-radius: 10px;
    color: #ffffff;
}

.bpw-header-name{
    color: #ffffff;
}

/* Styling for the typing bubble */
.bpw-typing-bubble {
    background: #ffffff;
}

/* Styling for the chat bubble content */
.bpw-chat-bubble-content {
    background-color: #000e14;
    border-radius: 3px;
    border: 1px transparent;
}

/* Styling for the date container, header name, and header subtitle */
.bpw-date-container,
```

```

.bpw-header-name,
.bpw-header-subtitle {
  color: #ffffff;
}

/* Styling for the layout of the chat bubble. Width, height, border,
position and radius of the chat bubble */
.bpw-layout {
  width: 360px !important;
  height: 60vh;
  border-radius: 10px;
  right: 52px;
  bottom: 52px;
  border: 1px;
}
/* Responsive design rules for devices with width less than or equal to
767px */
@media screen and (max-device-width: 767px) {
  .bpw-layout {
    width: 100% !important;
    height: 100%;
    right: 0;
    bottom: 0;
    border-radius: 0;
  }
}

/* Changes the fill color of the header icons to white */
.bpw-header-icon,
.bpw-header-icon svg,
.bpw-header-icon svg path {
  fill: #ffffff !important;
}

/* Changes the color of the placeholder text in the input message */
#input-message::placeholder {
  color: rgba(255,255,255,.50);
}

.bpw-composer textarea{
  outline: none !important;
  border: 1px solid rgba(255,255,255,.35);
}
.bpw-composer textarea:focus{
  outline: none !important;
  border: 1px solid rgba(255,255,255,.75);
}

/* Styling for the keyboard single choice option */
.bpw-keyboard-single-choice {
  background-color: #02010a;
  border: none;
}

/* Styling for the buttons in the chat interface */
.bpw-button,
.bpw-button-alt {
  background-color: #000;
  color: #ffffff;
  border-radius: 10px;
  border: none;
}

```

```

}

/* Styling for hover effect on the buttons in the chat interface */
.bpw-button:hover,
.bpw-button-alt:hover {
    background-color: #000e14;
}

/* Styling for hyperlinks in the chat interface */
a {
    color: #ffffff;
    text-decoration: underline;
}

/* Styling for the chat container, including color, scrollbar width and
color, and border */
.bpw-chat-container {
    background-color: #02010a;
    scrollbar-width: thin;
    scrollbar-color: #000e14 #02010a;
    border: none;
}

/* Styling for the chat bubble content when it's from the bot */
.bpw-from-bot .bpw-chat-bubble .bpw-chat-bubble-content {
    background-color: #000e14;
    color: #ffffff;
}

/* Styling for the chat bubble content when it's from the user */
.bpw-from-user .bpw-chat-bubble .bpw-chat-bubble-content {
    background-color: #00141f;
    color: #ffffff;
}

/* Styling for the composer section of the chat interface */
.bpw-composer {
    background-color: #02010a;
    border-top: none;
}

/* Styling for the scrollbar in the chat container */
.bpw-chat-container::-webkit-scrollbar,
.bpw-chat-container::-moz-scrollbar {
    width: 10px;
    background-color: #02010a;
    border: none;
}

/* Styling for the avatar of the bot */
.bpw-bot-avatar img,
.bpw-bot-avatar svg {
    background: #ffffff;
    border: 3px solid #ffffff;
}

/* Styling for the general scrollbar in the web page */
::-webkit-scrollbar {

```

```

    width: 0.5rem;
}

/* Styling for the track of the general scrollbar and the chat
container scrollbar */
::-webkit-scrollbar-track,
.bpw-chat-container::-webkit-scrollbar-track,
.bpw-chat-container::-moz-scrollbar-track {
    background-color: transparent;
}

/* Styling for the thumb of the general scrollbar and the chat
container scrollbar */
::-webkit-scrollbar-thumb,
.bpw-chat-container::-webkit-scrollbar-thumb,
.bpw-chat-container::-moz-scrollbar-thumb {
    background-color: #02010a;
    border-radius: 1rem;
    border: 0.5rem solid transparent;
}

/* Styling for the floating button icon in the chat interface */
.bpw-floating-button i svg path {
    fill: #3030e4;
}

/* Styling for the 'powered by' section of the chat interface */
.bpw-powered {
    text-align: center;
    padding: 10px;
    color: #ffffff;
    background: #02010a;
    font-size: 14px;
    border-bottom-right-radius: 10px;
    border-bottom-left-radius: 10px;
}

/* Styling for the hyperlinks in the 'powered by' section of the chat
interface */
.bpw-powered a {
    color: #ffffff;
    text-decoration: underline;
}

/* Styling for hover effect on the hyperlinks in the 'powered by'
section of the chat interface */
.bpw-powered a:hover {
    text-decoration: underline;
}

.bpw-send-button{
    background: #000000;
}

/* Change Bot Widget Icon */

.bpw-widget-btn{
    border-radius: 50%;

```

```

    background:none;
}

.bpw-floating-button::before {
background:url (https://images.contentstack.io/v3/assets/bltb6530b271fdd
d0b1/blte8a8889d3b17a9b9/5eb7cf1e1ea0c32e33b95fa6/TX_Reyna_X.png);
background-repeat: no-repeat;
    background-position: center;
background-size: contain;
height:undefinedpx !important;
width:undefinedpx !important;
}

.bpw-floating-button {
    box-shadow: none !important;
}

.bpw-floating-button:hover {
    box-shadow: none !important;
}

// .bpw-floating-button > i {
// display: none;
// }

```

3.5 KNOWLEDGE BASE

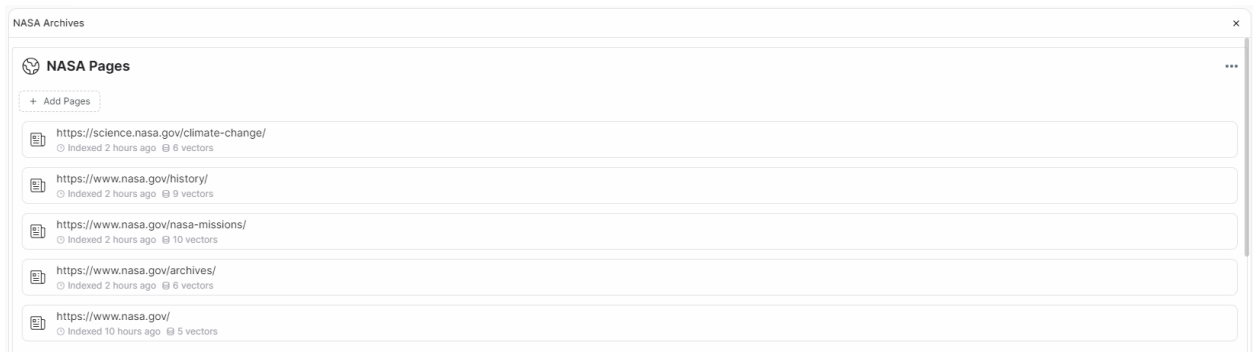


Figure 9: Knowledge Base Data

In Botpress, the knowledge base serves as a vital repository of information that empowers the chatbot to deliver accurate and up-to-date responses to user queries. One distinctive feature involves feeding diverse datasets from different websites into the bot, ensuring a comprehensive and dynamic understanding of various topics.

This process not only enriches the bot's knowledge but also allows it to stay current with the latest information available on the web. By continuously updating the knowledge base with content from various websites, the chatbot becomes adept at providing users with relevant and precise responses, making the conversation more informative and engaging.

This approach aligns with the commitment to delivering a chatbot experience that is both intelligent and responsive, catering to the ever-evolving nature of user queries and the dynamic landscape of online information.

CHAPTER 4: IMPLEMENTATION

4.1 DEPLOYING CHATBOT



Figure 10: Chatbot “Donna”

In order to integrate the chatbot into any website, a specific code is necessary for its successful deployment and functionality.

The following code is used to deploy the chatbot on the website:

```
<script src="https://cdn.botpress.cloud/webchat/v1/inject.js"></script>
<script>
  window.botpressWebChat.init({
    "composerPlaceholder": "Namaskar!",
    "botConversationDescription": "Empowering Space Education",
    "botId": "b1bd7315-3e9a-45f6-a419-737586465998",
```

```

    "hostUrl": "https://cdn.botpress.cloud/webchat/v1",
    "messagingUrl": "https://messaging.botpress.cloud",
    "clientId": "b1bd7315-3e9a-45f6-a419-737586465998",
    "webhookId": "677a3b77-f32c-4015-b8c4-96c79256cd27",
    "lazySocket": true,
    "themeName": "prism",
    "botName": "Donna",
    "avatarUrl": "https://lolfinality.com/wp-
content/uploads/2020/08/Empress.png",
    "stylesheet": "https://webchat-styler-
css.botpress.app/prod/8c26e738-9a2e-4bd2-b6b1-
2ad426f1f7fc/v43987/style.css",
    "frontendVersion": "v1",
    "useSessionStorage": true,
    "theme": "prism",
    "themeColor": "#2563eb"
  });
</script>

```

4.2 CREATING TEMPLATE WEBSITE

To ensure the seamless functionality of the bot, we've developed a template website specifically designed to assess and verify the chatbot's responses.

I. HTML

```

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width,
initial-scale=1.0">

```



```

<title>NASA Website</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

    <h1>NASA</h1>

    <script
src="https://cdn.botpress.cloud/webchat/v1/inject.js"></script
>

    <script>

        window.botpressWebChat.init({

            "composerPlaceholder": "Namaskar!",

            "botConversationDescription": "Empowering Space
Education",

            "botId": "b1bd7315-3e9a-45f6-a419-737586465998",

            "hostUrl": "https://cdn.botpress.cloud/webchat/v1",

            "messagingUrl": "https://messaging.botpress.cloud",

            "clientId": "b1bd7315-3e9a-45f6-a419-737586465998",

            "webhookId": "677a3b77-f32c-4015-b8c4-96c79256cd27",

            "lazySocket": true,

            "themeName": "prism",

            "botName": "Donna",

            "avatarUrl": "https://loifinity.com/wp-
content/uploads/2020/08/Empress.png",

```

```

        "stylesheet": "https://webchat-styler-
css.botpress.app/prod/8c26e738-9a2e-4bd2-b6b1-
2ad426f1f7fc/v43987/style.css",

        "frontendVersion": "v1",

        "useSessionStorage": true,

        "theme": "prism",

        "themeColor": "#2563eb"

    });

</script>

</body>

</html>

```

II. CSS

```

body {

    margin: 0;

    padding: 0;

    background: radial-gradient(circle at center, #1b263b
50%, #0d1b2a 100%);

    color: #fefae0;
    /* #fefae0; */

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    height: 100vh;

```

```

        font-family: 'Nasalization Rg';

        overflow: hidden;

    }

    @font-face {

        font-family: 'Nasalization Rg';

        src:          local('nasalization'),          url('nasalization.otf')
        format('opentype');

        font-weight: normal;

        font-style: normal;

    }

    #nasa-logo {

        opacity: 1;

        animation: left 2s ;

        animation-fill-mode: forwards; /* Keep the final state
after the animation */

        animation-delay: 1s;

    }

    h1 {

        opacity: 0;

        animation: right 2s ;

        animation-fill-mode: forwards; /* Keep the final state
after the animation */

        animation-delay: 1s;

    }

```

```

@keyframes right {

    from {

        transform: translateX(0px);

    }

    to {

        opacity: 1;

        transform: translateX(100px);

    }

}

@keyframes left {

    from {

        transform: translateX(0) ;

    }

    to {

        opacity: 1;

        transform: translateX(-150px) scale(0.75);

    }

}

h1 {

    font-size: 5em;

    text-align: center;

```

```

        font-weight: bolder;

        margin-top: -5px;

        z-index: -1 ;

        position: absolute;

    }

    #nasa-logo {

        width: 300px;

        margin-top: -60px;

        z-index: 5 ;

        position: absolute;

    }

    /* Mobile Layout*/

    @media only screen and (max-width: 600px) {

        html, body{

            overflow: hidden;

        }

        @keyframes right {

            from {

                transform: translateX(0px);

            }

            to {

```

```

        opacity: 1;

        transform: translateX(70px);

    }

}

@keyframes left {

    from {

        transform: translateX(0) ;

    }

    to {

        opacity: 1;

        transform: translateX(-70px) scale(0.75);

    }

}

h1 {

    font-size: 3em;

    text-align: center;

    font-weight: bolder;

    margin-top: 0;

    z-index: -1 ;

    position: absolute;

}

#nasa-logo {

```

```

width: 150px;

margin-top: -35px;

z-index: 5 ;

}}

```

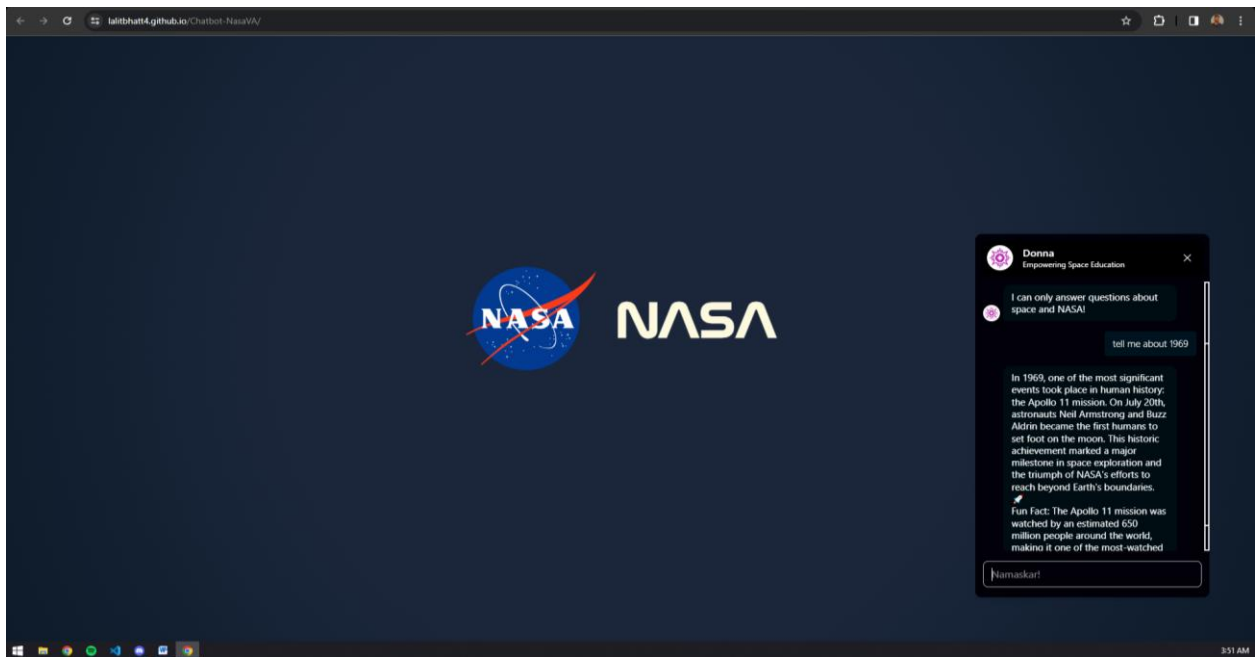


Figure 11: Template Website with Working Chatbot

CHAPTER 5: CONCLUSION

The introduction of the NASA Virtual Assistant was spurred by the imperative need for instant, personalized, and intuitive assistance within the vast repository of NASA's information. OpenAI's collaboration with NASA underscores a commitment to enhancing user experiences and providing real-time support through the amalgamation of advanced artificial intelligence and user-friendly design.

As users engage with the NASA Virtual Assistant, they can anticipate a transformative experience, tapping into a wealth of knowledge effortlessly. Beyond being a technological milestone, this project exemplifies a dedication to fostering a more interactive and informative online space exploration experience. It symbolizes a new era of exploration and discovery, fueled by the power of artificial intelligence.

Furthermore, the exploration of virtual assistant applications in space agencies, as outlined in the review, emphasizes the potential for streamlined communication, information provision, and user engagement. The integration of AI-driven assistants, particularly GPT-3 by OpenAI, into the NASA Virtual Assistant introduces an unprecedented level of responsiveness and intelligence.

The implementation of the project involved leveraging tools such as Visual Studio Code (VS Code) for website styling and OpenAI's API for accessing the capabilities of GPT-3. Additionally, Botpress played a pivotal role in simplifying the development of the chatbot, offering a versatile and open-source platform for creating sophisticated conversational experiences.

In conclusion, the NASA Virtual Assistant project represents a harmonious fusion of technological innovation and user-centric design, serving as a testament to the transformative potential of artificial intelligence in advancing the accessibility and interactivity of information within the realm of space exploration.

REFERENCES

1. NASA. Official NASA Website. Retrieved from <https://www.nasa.gov/>
2. Botpress. Open-source Conversational Platform. Retrieved from <https://botpress.io/>
3. Microsoft. Visual Studio Code. Retrieved from <https://code.visualstudio.com/>
4. OpenAI. OpenAI API Documentation. Retrieved from <https://beta.openai.com/docs/>
5. Kaggle. Kaggle: Your Machine Learning and Data Science Community. Retrieved from <https://www.kaggle.com/>