# Empowering AI Customization and Accessibility: How Conversational AI in Botpress Transforms User Experiences

## AMS 560/CSE 542

**Abhishek Kalugade(115691502)**

**Peilin Zhu(110306362)**

**Xingtong Zhou(111690523)**

**Instructor: Zhenhua Liu**

# 1.Introduction

Empowering AI Customization and Accessibility by designing a conversational AI in Botpress transforms user experiences. This design can be divided into two parts. The first part is the implementation of the chatbot based on NLP knowledge via Botpress. In this part, we emphasize the Modular Customization and Knowledge Base Customization to provide users more flexibility when designing their own chatbot. In the second part, our goal is to design a low-code/no-code interface with UI accessibility to fit a larger group of users with less related knowledge background.

# 2. Problem and Solution

2.1 Problem statement

Conversational AI has transformed human-computer interaction by enabling natural language communication, yet significant challenges remain in making this technology accessible and customizable for diverse users. These challenges include:

1.Customization Limitations:

- Customization is typically a complex and time-intensive process, hindering adoption by users without programming skills.
- Domain-specific requirements, such as those in science, business, or healthcare, are not easily addressed by generic AI models.

2.Accessibility Challenges:

- Many conversational AI platforms lack user-friendly interfaces, making them inaccessible to non-technical users.
- Traditional configuration methods rely heavily on manual intervention or Infrastructure as Code (IaC), which can be daunting for beginners.

2.2 Proposed Solution(what we want to delivery):

To address these challenges, we propose leveraging Botpress, an open-source conversational AI platform, to enhance both customization and accessibility. The solution focuses on two primary pillars:

1.AI-Driven Customization:

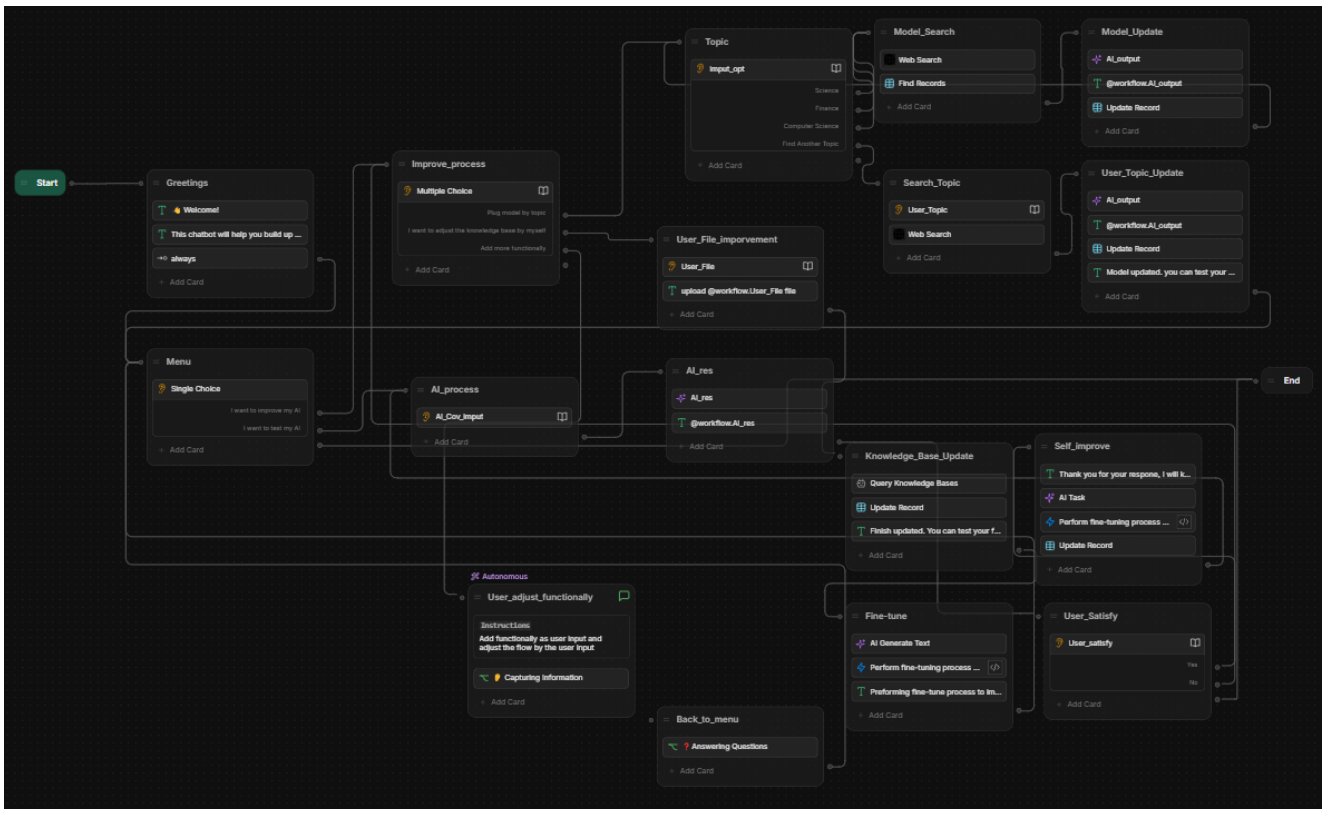
- Users interact directly with the AI to define and refine custom models tailored to their unique requirements.
- The system employs modular components that adapt dynamically to different domains like healthcare, education, and business.
- Continuous user feedback improves the system's ability to provide precise and specialized responses.
- The AI evolves workflows in real-time based on user input, creating personalized experiences without the need for manual configurations.

2. AI-Driven Accessibility:

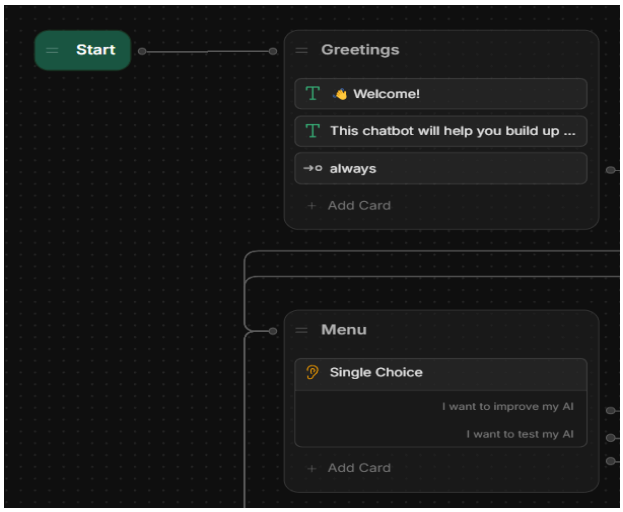
- Combines traditional IaC methods with AI-driven, low-code/no-code interfaces to simplify the setup process.
- Modular templates allow users to configure their AI systems as easily as assembling LEGO blocks, making the platform more approachable.
- AI guides users through the configuration process, automating complex tasks and ensuring error-free setups.

# 3. Methodology

## 3.1 AI-Driven Customization



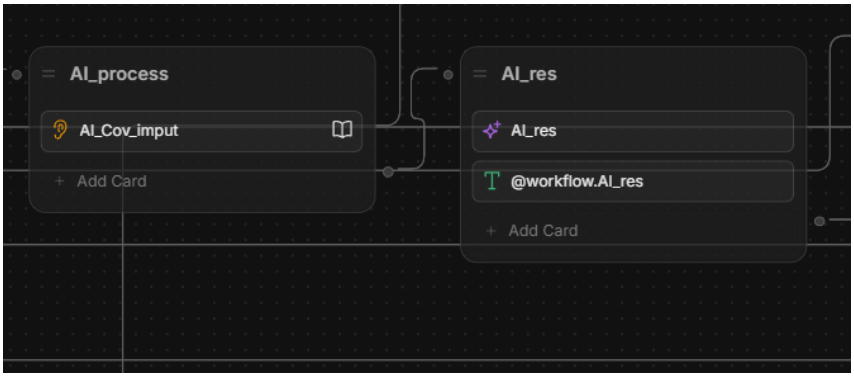

Here is our overview of our conversational AI model workflow. I will get into details of each node functionality.

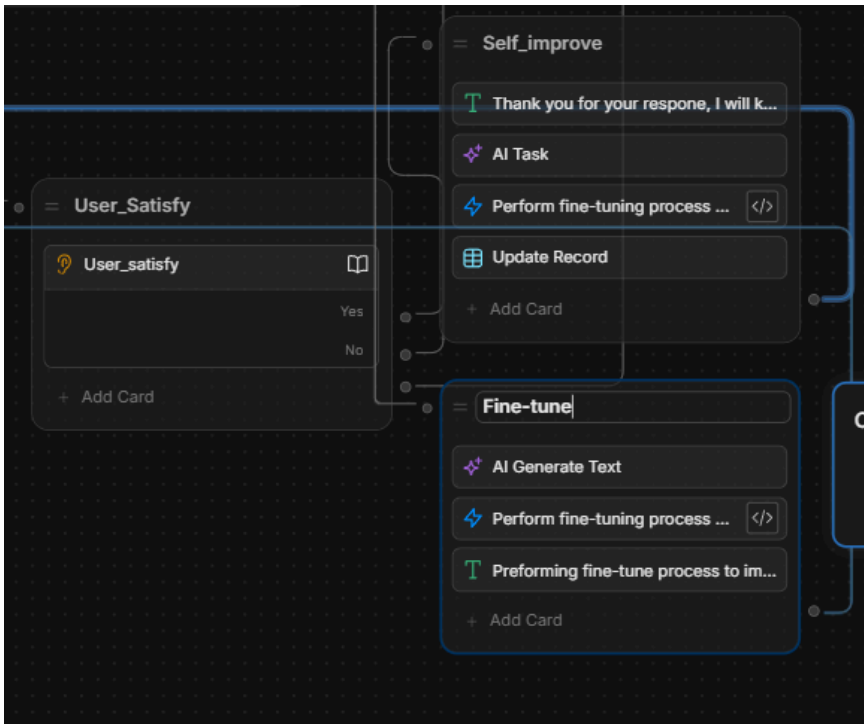



This chatbot starts with a **Greetings** node, welcoming the user and transitioning to the **Menu** node. In the Menu, users are presented

with two options: **'I want to improve my AI**' or **'I want to test my AI**,' allowing for a tailored interaction based on their goal.

The **AI_process** node listens to user input and passes it to the **AI_res** node. Within the **AI_res** node, the **AI_res** card generates a response based on the user input, and the **workflow.AI_res** card outputs the AI response to the user.

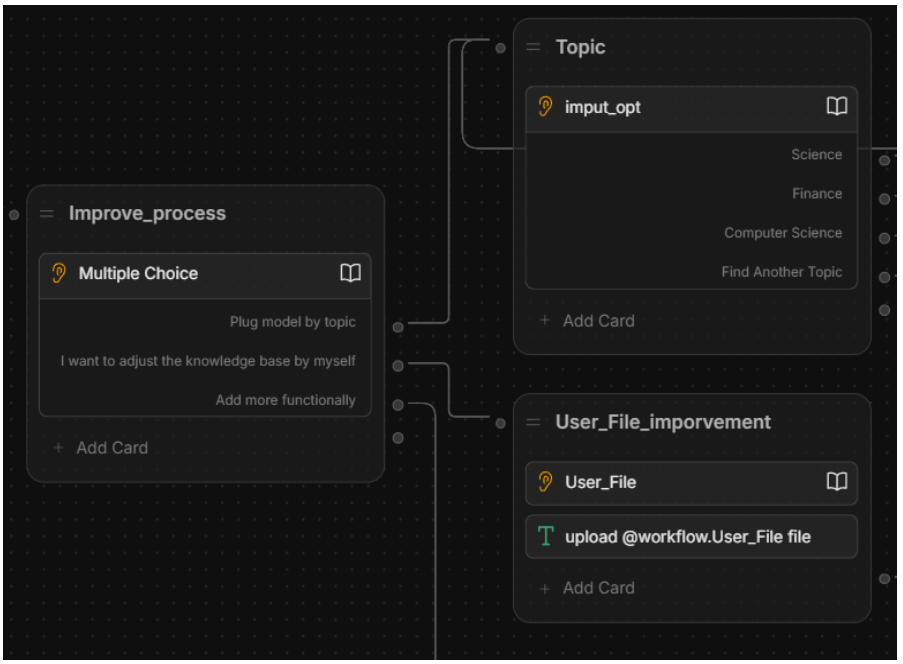


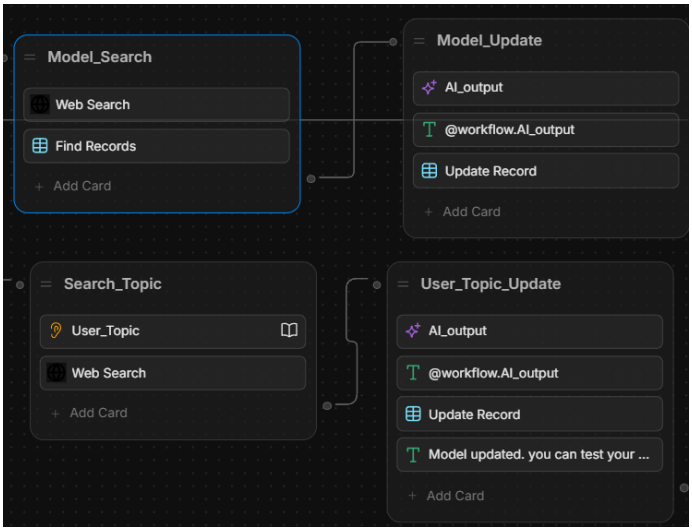




The user satisfaction process begins with the **User_Satisfy** node , after every conversation user is prompted to indicate their satisfaction. If the user selects 'Yes,' they are directed to the **Self_improve** process, where the AI acknowledges their response, performs a fine-tuning task, and updates records to the Knowledge base. Afterward, the process transitions to the **Fine-tune** node, where additional fine-tuning steps are carried out to improve AI performance. If the user selects 'No,' they are redirected to an alternative improvement process that we will present in the next.

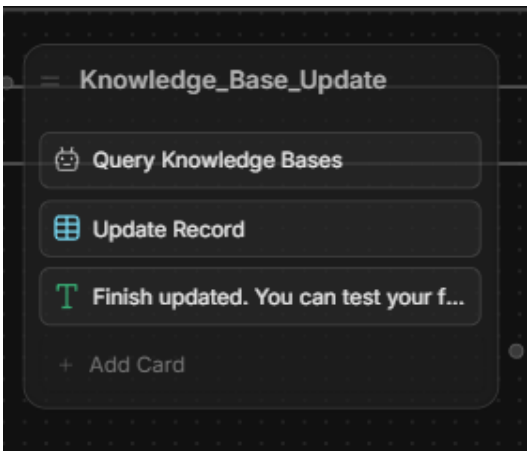
This is the improvement process with the node **Improve_process** connect being connected by the **menu** node and **User_Satisfy** node. Whenever any user selects the “I want to improve my AI” or is not satisfied with the AI response, it will transfer to here. Ai will listen to the user's

choice to **“Plug model by topic”** or “**adjust the knowledge base by myself**”. If the user chose to plug in the AI model by topic, it will process to **Topic** node it has some default topic like **“Science”, “Finances”, “Computer Science”**, or user can select **“Find Another Topic”** to manually input the topic. The “**adjust the knowledge base by myself**” opinion will lead the user to **User_File_imporvement** Node which will lead the user to upload the knowledge base file. AI will use the file to expand the knowledge base which we will present in the next graph. (Since our website deployment fails due to the time constraint, users unable to upload the file in Botpress showcase website,we have to manually upload the file in the back end).



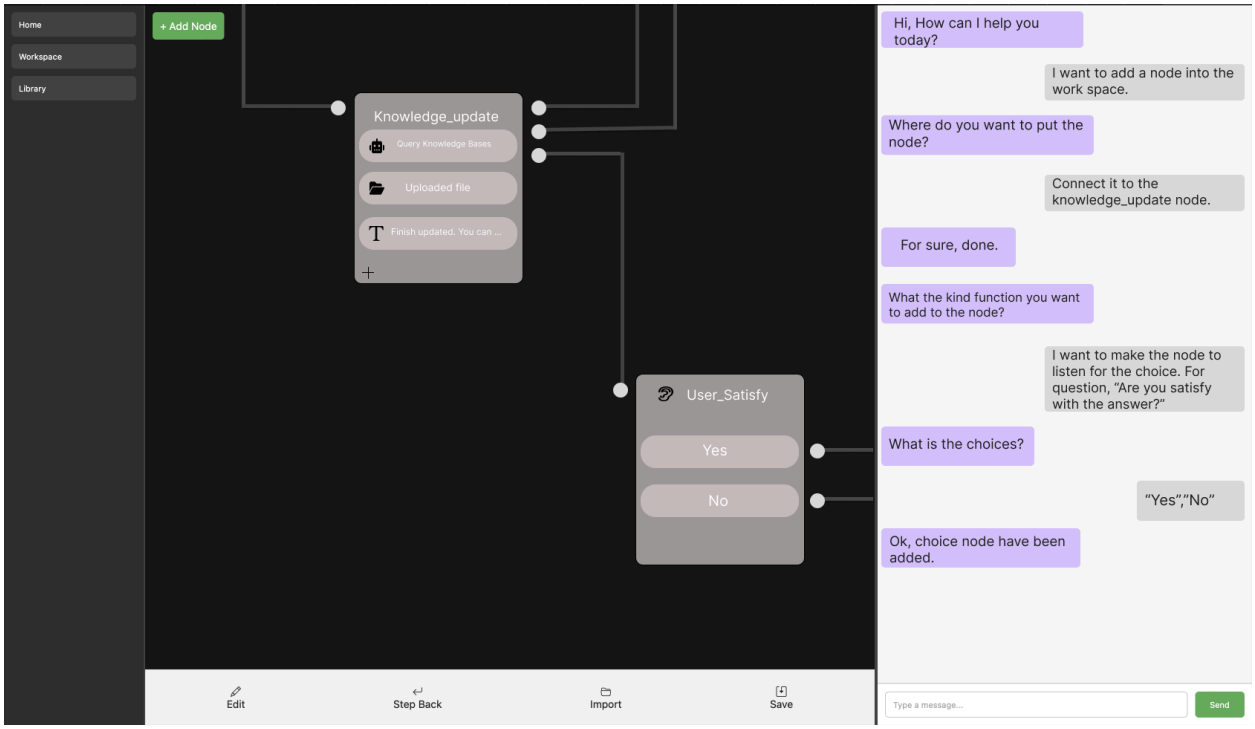

This process is a Model plugin process which is pretty similar for user input model **Search_Topic** will listen to user input then use the input to search which model is the best for the topic then try to plugin the similar model as much as it can. Therefore we have a default topic in case we can't find any model that works. AI will plugin the model and at same time update the record so AI will know what kind of model it plugged in.




This **Knowledge_Base_Updata** node shows how AI will adopt user input knowledge base data. First AI will query the knowledge bases to make sure the data

upload is not repeated. Then upload the data we get from the **User_File_improvement** node to expand the knowledge base of our AI model.

3.2 AI-Driven Accessibility




We design a low-code interface for users with AI deployed. On the right side of the screen, users can interact with the chatbot to set up the configuration of their own bot. IDE space is right in the middle for users to make adjustments after AI generates configuration based on the chat. Configuration parts are already modularized and presented in a low-code way by flow-like diagrams to make the design more user friendly. More detail and basis about the UI evaluation will be explained in the "Performance Evaluation" section.

# 4. Accomplished(Delivered)

We met some technical issues to deploy our chatbot into HTML. In this case, we can't finish the backend implementation of the javascript, so that users can't upload customization files at the web-end. As a result, we design the basic UI in HTML, and demo an ideal page of how it should be worked in Figma.

4.1 AI-driven Customization:

- Users are able to directly engage with the AI to define and refine custom models tailored to their specific needs.

- The system leverages modular components that dynamically adapt to various domains, such as healthcare, education, and business.
- Continuous user feedback enhances the system's precision and ability to deliver specialized responses.
- Additionally, the AI evolves workflows in real time based on user input, enabling personalized experiences without requiring manual configurations.

4.2 AI-driven Accessibility:

- HTML file for the overall GUI design
- Figma demo for the low-code interface with GUI design

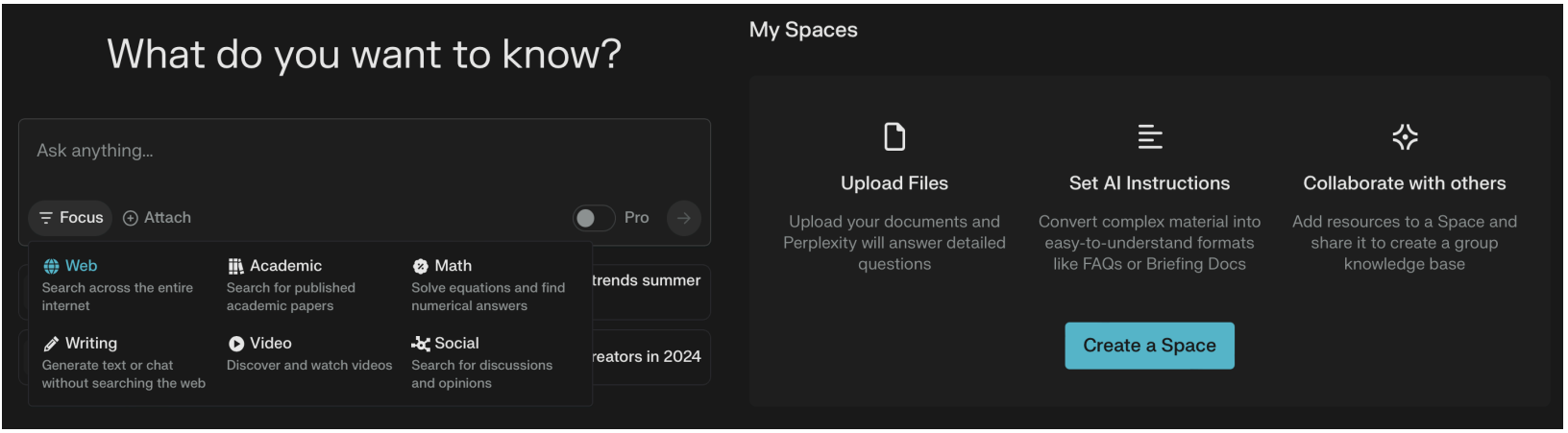
# 5. Responds to the comments

5.1 Comparing to similar tools

To respond to the question in the final presentation, we think Prof. Liu tried to guide us by comparing some similar tools first, and evaluate the performance of our design. After a lot of searching, it is determined that a free version of those tools are missing. Most of these tools are enterprise facing. Under such circumstances, we decided to list some of the similar tools to show the feasibility of our design. Following this, we will also provide some methods to evaluate the effectiveness of these kinds of tools.

1.Similar tool for AI-driven customization design:

Perplexity.ai. This web-based AI chatbot has similar functionalities in Modular Customization and Knowledge Base Customization. From the "focus" bottom, users can first decide a module which is related to the topic of their question. After that, by creating a space, users can upload all related files for the reference of the chatbot. By setting up a chatbot like this, users can consult for questions from specific areas. With module training for specific topics, it has a larger tendency to provide users more accurate answers.

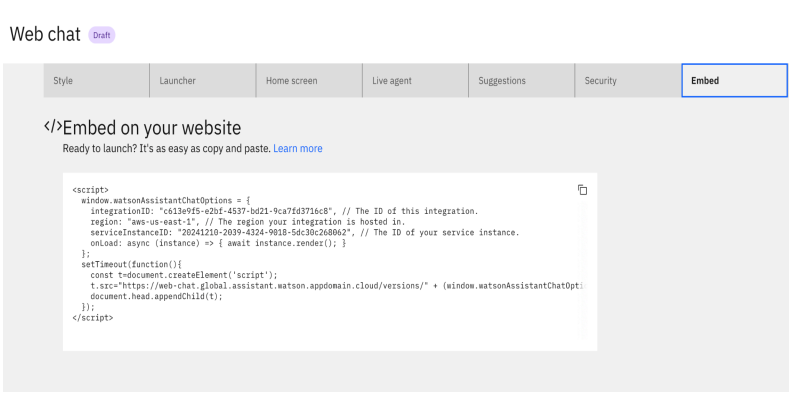


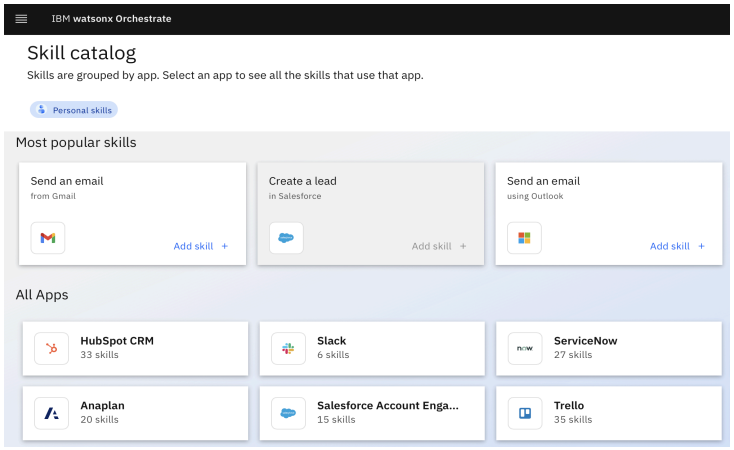
2.Similar tool for AI-driven accessibility design

IBM watsonx Orchestrate. IBM watsonx Orchestrate is a web-based application which allow users to create their own chatbot for automation purpose with some apps plug-in. This product has an overall similarity with our design. It provides users a clear flow-like structure interface for them to adjust the important actions within the conversation of their bot. However, it lacks the chat AI help during the design process. In this case, users require a time investment to go through the documentation in order to familiarize the basic structure of the set up.

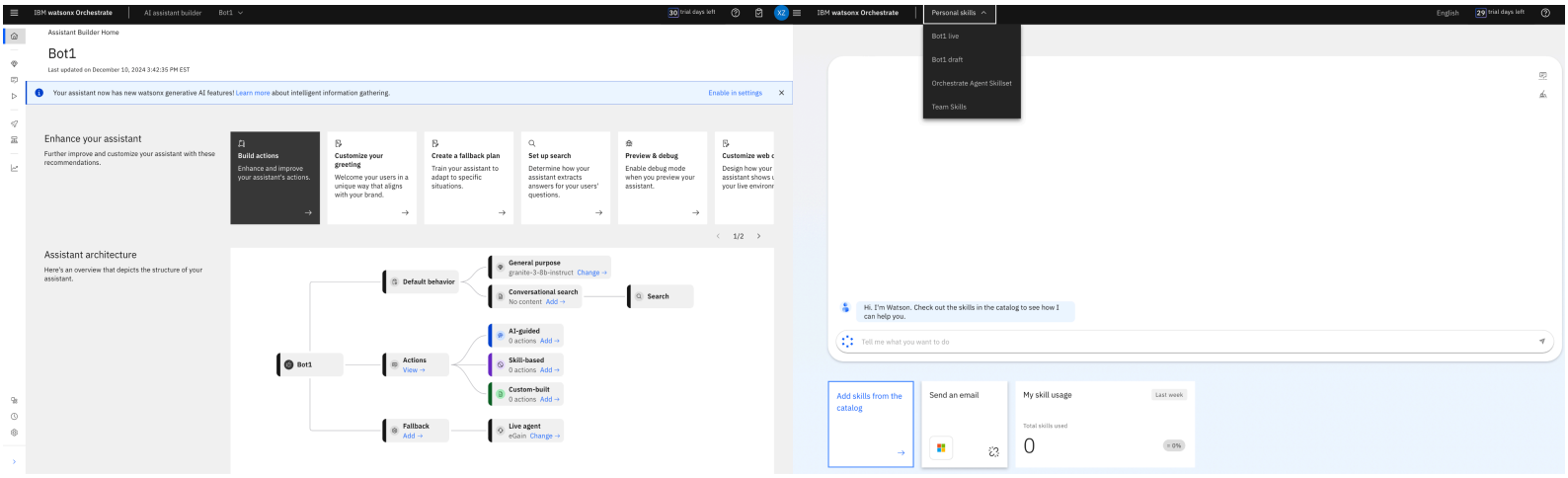
Crucial functionality that it provides:

Easy HTML embedment of the chatbot;Abundant skills base for other applications




Flow-like design structure for actions of chatbot

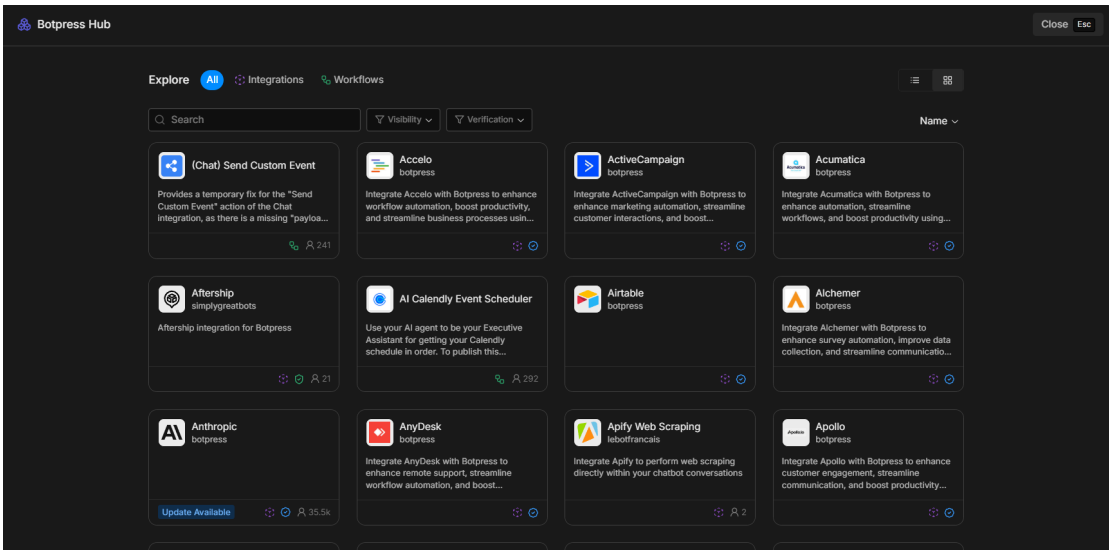



Legacy chat: quick deployment of own designed chatbot for adjustment and tryout

5.2 Performance Evaluation

1. Model Evaluation:

   Since we use botpress as our platform to build up our model and the conversation embedding in the botpress is ChatGPT-4o, so it is hard to make evaluation on top of it. Also for the other plugin model like the picture shows, different models will have different Standardized Metrics. Evaluating Botpress depends heavily on the complexity and variety of conversation flows designed within the tool.Depends on how well the flows handle edge cases, user intent recognition, and natural dialogue progression.Botpress relies on integrated or external NLU models, which may perform inconsistently across different languages, dialects, or domains.Training NLU models for custom intents and entities can be resource-intensive and may not generalize well across diverse user inputs.This variability makes it hard to isolate Botpress' platform efficiency from the effectiveness of the NLU models being used.




2. Pointing: target acquisition:

To study the efficiency of GUI design, the movement time between the starting position and the target icon needs to be measured . We are using 2D-Fitts' law to justify and evaluate our UI design. This law as $MT = a + b * ID$, where a is the initial movement time, b is the slope, and ID( index of difficulty) is defined as $ID = log_2((D/min(W, H)) + 1)$. (D is the distance between the target and cursor; W and H are the width and height of the target icon.)

With this law applied, we decided to put the IDE interaction space in the middle, so that the movement time to the side bar, bottom bar, and chatbox can be minimized at the same time. Moreover, by applying the 1D-Fitts' law, where $ID = log_2((D/W) + 1)$, it can be determined that the icons should be flattened and adhere to the side of the interface to maximize accessibility.

3. User satisfaction survey

To further evaluate the performance of our design, quantitative methods such as set a user satisfaction survey in some measurable perspect. In the survey, ordinal scalable

parameters will be used, which can be further quantified in the later study for iteration. Kinds of aspects will be setted up for the user to rate such as the quality of the responses from chatbot, the level of ease of use, the waiting time of the server, and etc. With limitations in the scale of this project, regrettably, we could set up such an expensive study. However, this is still a feasible method after the early stage of application development.

# 6. Contribution among group members.

Peilin handled the design part of the chatbot model on Botpress.
Abhishek did the documentation and provided background research of the feasibility of the project in designing phrases.
Xingtong handled the Web-GUI design and Figma demo part.

# 7. Reference

Botpress: https://botpress.com/
Perplexity.ai: https://www.perplexity.ai/
IBM watsonx Orchestrate: https://www.ibm.com/products/watsonx-orchestrate
Materials from CSE323 Human Computer Interaction

# 8. Link to our file

Botpress design:
https://cdn.botpress.cloud/webchat/v2.2/shareable.html?configUrl=https://files.bpcontent.cloud/2024/10/20/02/20241020022226-6E8KBPJO.json

Figma design:
https://www.figma.com/design/1hzDr9Zx5ZF8Woqad1oD5O/ui_-basedon_html?node-id=0-1&t=1yrhWlktyxsEF5JK-1

HTML:
https://github.com/Bleak-bleak/Web_HTML.git