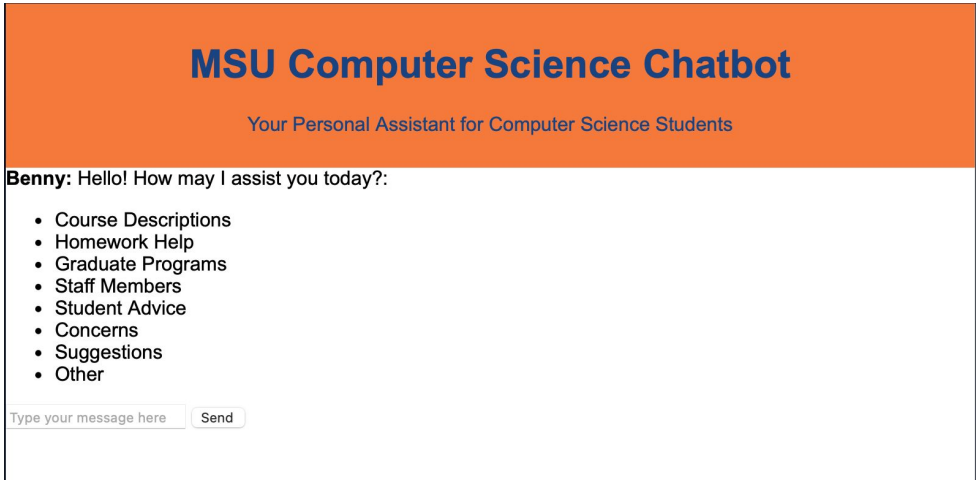
Team 2

Midterm Progress Report

Our development environment

We have developed a website demo page to publish and test our chatbot. We plan to have the page mimic the Morgan State University department of computer science current webpage.



Website Demo Page 1
Source: https://github.com/tellysmithjr5/Team-2-AI-Chatbot.git

Our development environment cont.

index.html > html

```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>MSU Computer Science Chatbot</title>
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <header>
    <div class="container">
      <h1>MSU Computer Science Chatbot</h1>
      <p>Your Personal Assistant for Computer Science Students</p>
    </div>
  </header>

  <main>
    <div class="container">
      <div class="chat-container">
        <div class="chat-box" id="chat-box"></div>
        <input type="text" id="user-input" placeholder="Type your message here">
        <button onclick="sendMessage()">Send</button>
      </div>
    </div>
  </main>

  <script src="script.js"></script>
</body>

</html>
```

script.js > onload

```js
window.onload = function() {
  var welcomeMessage = "<div><strong>Benny:</strong> Hello! How ma
  welcomeMessage += "<ul>";
  welcomeMessage += "<li>Course Descriptions</li>";
  welcomeMessage += "<li>Homework Help</li>";
  welcomeMessage += "<li>Graduate Programs</li>";
  welcomeMessage += "<li>Staff Members</li>";
  welcomeMessage += "<li>Student Advice</li>";
  welcomeMessage += "<li>Concerns</li>";
  welcomeMessage += "<li>Suggestions</li>";
  welcomeMessage += "<li>Other</li>";
  welcomeMessage += "</ul>";

  var chatBox = document.getElementById("chat-box");
  chatBox.innerHTML += welcomeMessage;
}

function sendMessage() {
  var userInput = document.getElementById("user-input").value;
  if (userInput.trim() !== "") {
    var chatBox = document.getElementById("chat-box");
    var userMessage = "<div><strong>You:</strong> " + userInput +
    chatBox.innerHTML += userMessage;

    var botResponse = respondToUserInput(userInput);

    var botMessage = "<div><strong>Chatbot:</strong> " + botRespon
    chatBox.innerHTML += botMessage;


    chatBox.scrollTop = chatBox.scrollHeight;


    document.getElementById("user-input").value = "";
  }
```

script.js > respondToUserInput

```js
}

function respondToUserInput(userInput) {

  userInput = userInput.toLowerCase();

  // Predefined responses based on user input
  var responses = {
    "course descriptions": "You can find course descriptions on th
    "homework help": "For homework help, consider reaching out to
    "graduate programs": "Information about graduate programs can
    "staff members": "You can find information about staff members
    "student advice": "For student advice, consider reaching out t
    "concerns": "If you have concerns, please contact the Morgan S
    "suggestions": "We appreciate your suggestions! Please feel fr
    "other": "How may assist you further?",
    // Adding more responses in this space
    "default": "I'm sorry, I'm not sure how to respond to that. Ca
  };

  for (var keyword in responses) {
    if (userInput.toLowerCase().includes(keyword)) {
      return responses[keyword];
    }
  }

  return responses["default"];
}
```

style.css > .chat-container

```css
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
}

header {
  background-color: #F47937;
  color: #1B4383;
  text-align: center;
  padding: 10px 0;
}

footer {
  background-color: #333;
  color: black;
  text-align: center;
  padding: 10px 0;
  position: static;
  bottom: 0;
  width: 100%;
}

.chatbot {
  margin: 20px auto;
  max-width: 600px;
  border: 1px solid #ccc;
  border-radius: 5px;
  padding: 20px;
}

.chat-container {
  height: 300px;
  overflow-y: auto;
}
```

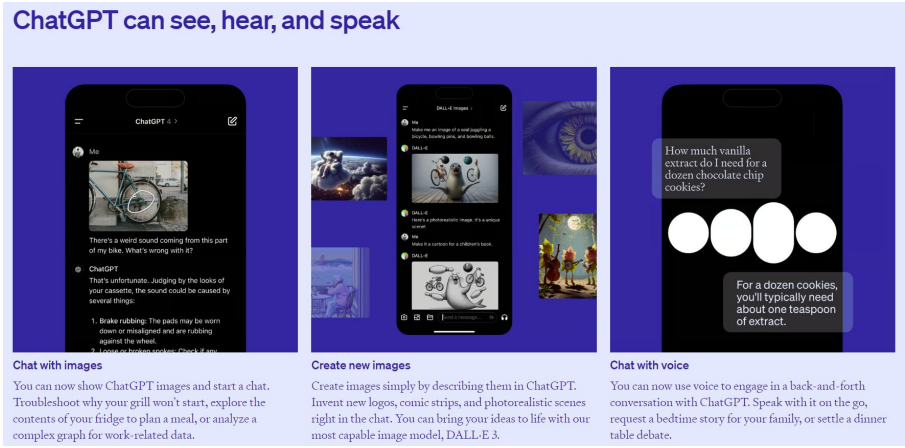
Our Progress

The team has been evaluating the pros and cons of Microsoft Copilot and OpenAI chatbot based on the research and testing currently being conducted by team members. So far the following features are being evaluated:

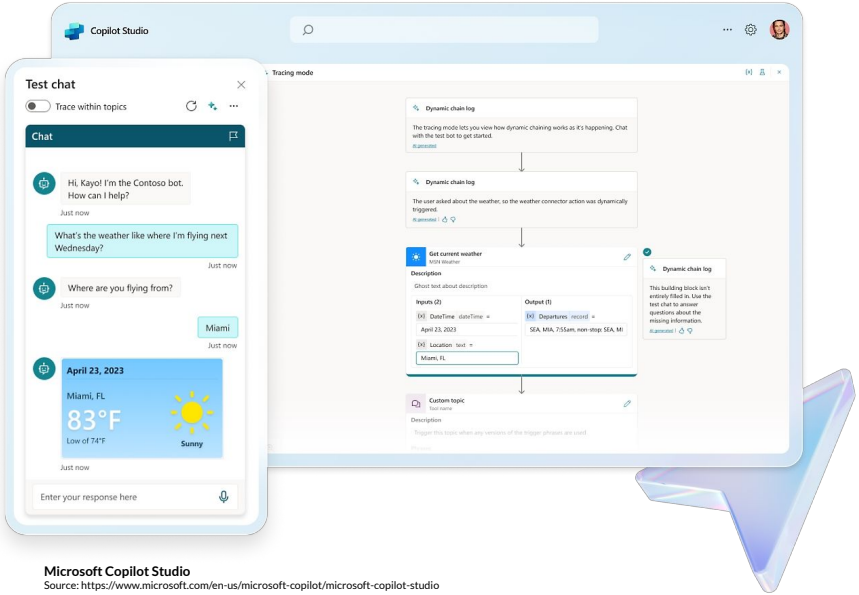
- The implementation and capabilities
- Cost
- Ease of use

FEATURE	CHATGPT	COPILOT FOR 365
PRIMARY USE	General-purpose chatbot and text generation tool	Productivity assistant for Microsoft 365 users
PRICING	Free for basic use, $20/month for ChatGPT 4	Requires a Microsoft 365 subscription
AVAILABILITY	Available through APIs and various platforms	Integrated into Microsoft 365 applications
STRENGTHS	Wide range of applications, including creative writing, code generation, and chat conversations	Can help with writing emails, documents, presentations, and more
WEAKNESSES	Can sometimes generate nonsensical or irrelevant text	May not be as helpful for users who are not familiar with Microsoft 365
BEST FOR	Users who need a versatile AI tool for various tasks	Microsoft 365 users who want to improve their productivity

ChatGPT vs. Copilot
Source: https://www.ntiva.com/blog/chatgpt-vs-microsoft-copilot



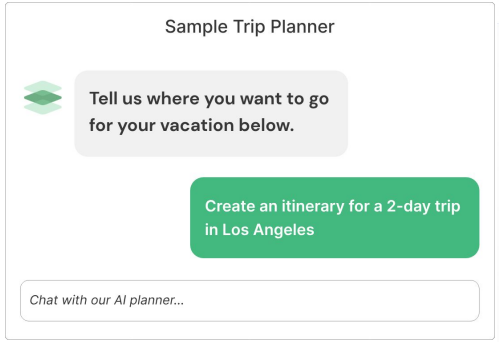
OpenAI features
Source: https://openai.com/chatgpt



Microsoft Copilot Studio
Source: https://www.microsoft.com/en-us/microsoft-copilot/microsoft-copilot-studio

OpenAI Research & Testing

The following are examples of what the team has been recently researching and testing using OpenAI:

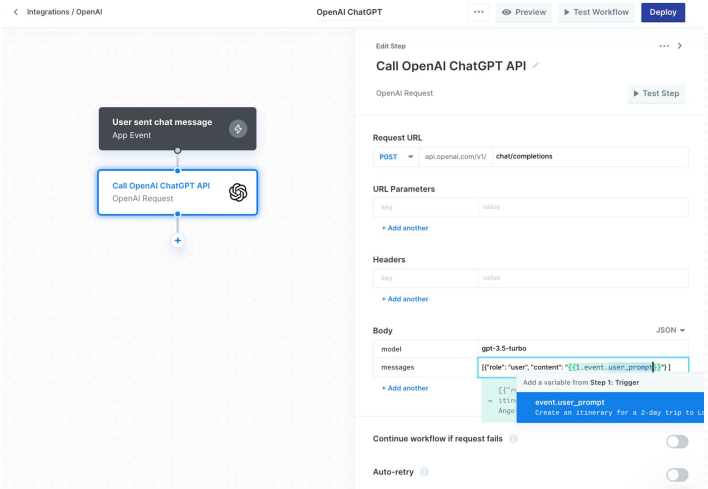


OpenAI Sample Request
Source: https://www.useparagon.com/blog/how-to-build-a-native-openai-integration

-This is an example of a sample request to the OpenAI API using chat API.

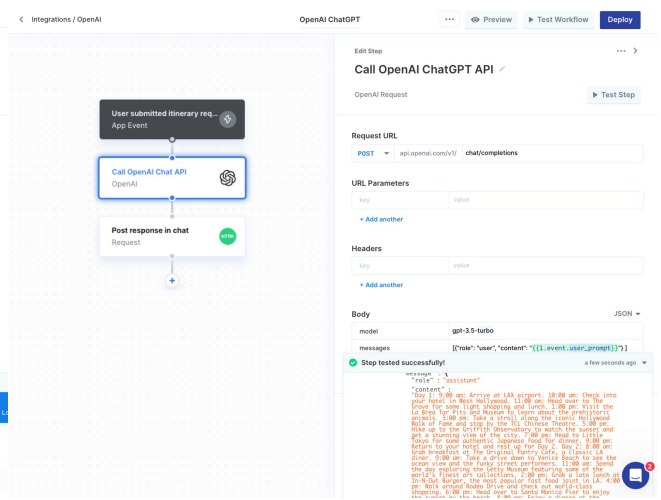
-We learned that OpenAI offers several different APIs and each of them offer a specific focus and use case, so you'll need to choose the API that best fits the use case that you want to provide your users.

-For conversational features to an application, chat API is recommended



OpenAI Workflow Example 1
Source: https://www.useparagon.com/blog/how-to-build-a-native-openai-integration

Here's what the workflow behind the scenes looked like. The user's chat message is used as an app event to trigger the OpenAI integration workflow.

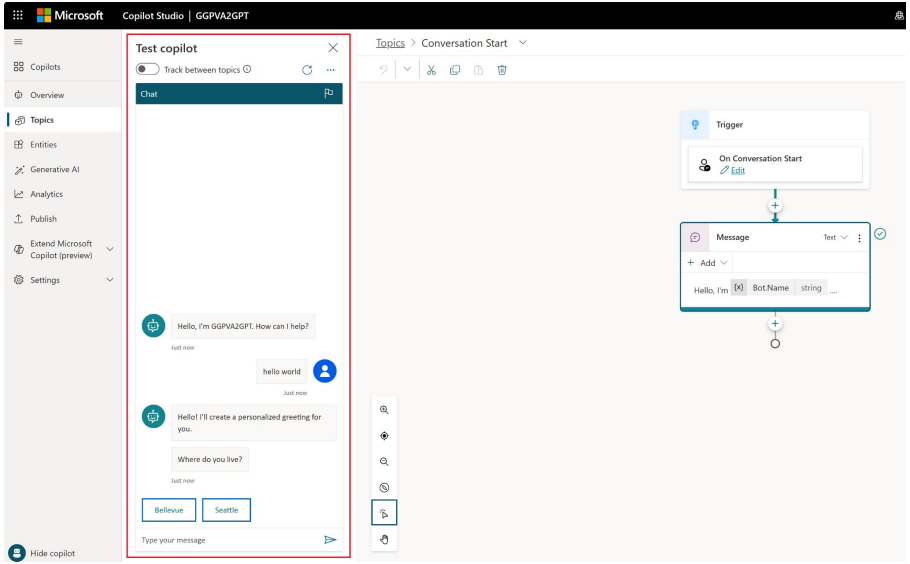


OpenAI Workflow Example 2
Source: https://www.useparagon.com/blog/how-to-build-a-native-openai-integration

When the test request is executed the in the workflow the response gets generated by OpenAI's chat API

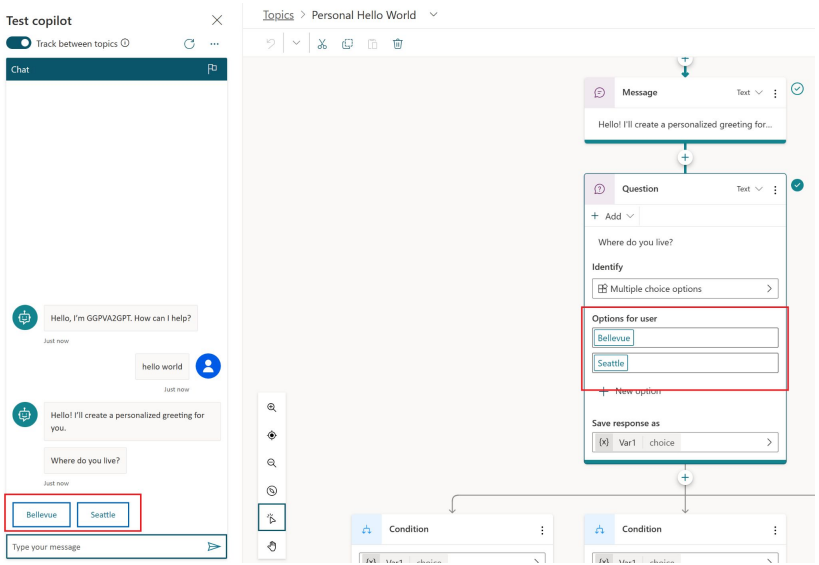
Microsoft Copilot Research & Testing

The following are examples of what the team has been recently researching and testing using Microsoft Copilot:



Copilot Studio Example 1
Source: https://learn.microsoft.com/en-us/microsoft-copilot-studio/fundamentals-get-started?tabs=web

Our team researched and tested content in real time using tutorials like this. In this example, content is authored into a dialog tree like the one seen here on the right. Then the conversation is tested in real time to see if it's working as expected using the copilot pane in Copilot Studio.



Copilot Studio Example 2
Source: https://learn.microsoft.com/en-us/microsoft-copilot-studio/fundamentals-get-started?tabs=web

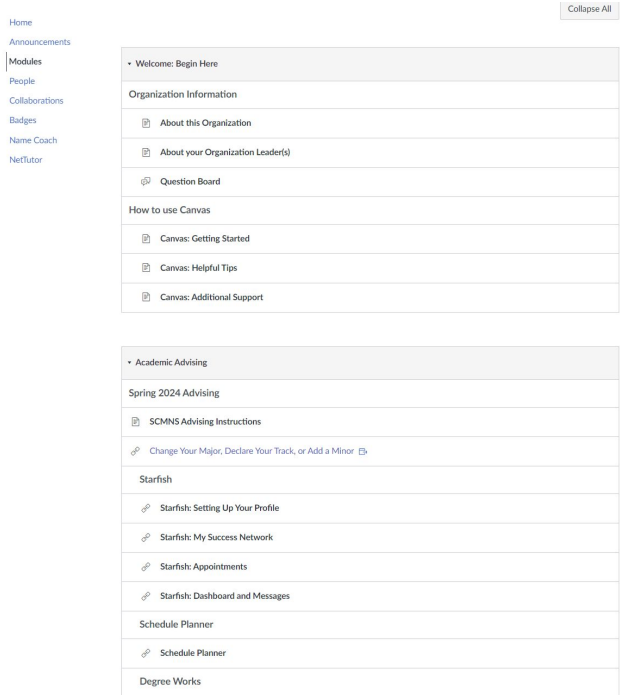
-Here the user types "hello world" in the chat window, and sends the message to the copilot.

-Once the message is sent you will see on the top right portion of the dialog tree is highlighted in green that Seattle and Bellevue are presented as user options in the test copilot pane.

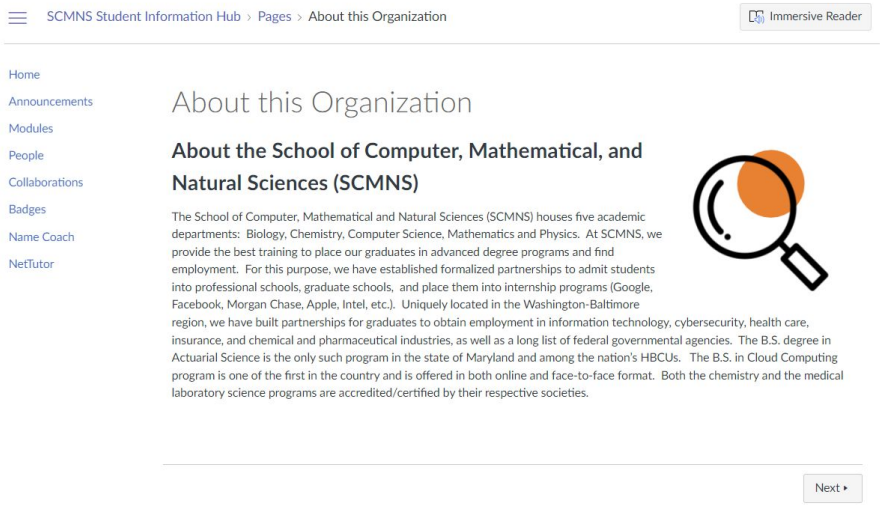
-Now the copilot is waiting for the user to respond while displaying suggestions on how to respond. The suggestion buttons are going to reflect what you authored within your dialog tree in the Ask a question node. Then, in the test copilot you can either select these suggestion buttons to continue, or you can enter your response into the chat window.

Gathering Training Data

We have been gathering training data to answer the users frequently asked questions/answers. Our current focus is on current students inquires.



Training Data Source Example 1: SCMNS Student Information Hub
Source: SCMNS Student Information Hub Modules on Canvas



Training Data Source Example 2: SCMNS Student Information Hub
Source: SCMNS Student Information Hub Modules on Canvas

Gathering Training Data

- Reviewed feedback and frequently asked questions from current CS students to understand the topics they have inquire about. (course registrations, homework assistant, faculty contact information, academic resources)
- Data drawn from Morgan State official website, Canva SCMS Student Information Hub, and emails sent out from faculty members.

★ Who can the students contact for:
  - Transferring credits
  - Academic advising
  - Internship/job opportunities
  - Specific interest, (game development,SWE AI/ML, cybersecurity)
  - Research opportunities
  - Scholarship
  - Funding for events/organizations

★ Can you provide more information on this course and its syllabus?
  - Listed below are

★ What academic support services are available to computer science students?
  - Academic advising
  - Internship opportunities
  - Organizations pages

★ What are the STEM oriented organizations on campus?
  - WiCS, GSDC, SWE,NSBE, SACS,

★ Where can I go to get help with resume building, technical interviews, career counseling?
  - Tyler Hall, Library, Counseling service, CASA, PEERS bears

★ What equipment and resources does the department offer to students? (devices,lab,website subscriptions)
  - Microsoft word, Adobe Photoshop, Canva,

★ What graduate programs are aligned with the Computer Science curriculum?
  - M.S. Advanced computing
  - M.S. Bioinformatics
  - M.S. Integrated Science

Goals for the next two weeks

1. Upload team documentation to github
2. Select a framework (Microsoft Copilot or OpenAI)
3. Prepare training data for chatbot
4. Continue to develop demo website page design
5. Create chatbot user interface design
6. Test chatbot queries
7. Publish the chatbot to our demo website page