Conversational Chatbot using Google Cloud Dialogflow

- Tharani Vadde
- Moulica Vani Goli
- Beryl Caleb

Introduction

Project Overview:

In this project, we are developing a conversational chatbot using Google Cloud Dialogflow. Developing a conversational AI chatbot for a university involves leveraging several AI services in the Google Cloud Platform (GCP), particularly using Dialogflow, Cloud Storage, and Agent Builder.

Objectives:

1. Enhance User Engagement and Experience
2. Optimize Conversation Management
3. Facilitate Seamless Integration
4. Leverage Data-Driven Insights
5. Promote Scalable Solutions

Project Goals

Our project aims to achieve the following specific goals and deliverables:

- **Develop a Functional Chatbot:** We will build a conversational chatbot using Google Cloud Dialogflow that can understand user queries and provide relevant responses.
- **Enable Basic Interactions:** Ensure the chatbot can handle common interactions such as answering FAQs, providing information, or assisting with simple tasks.
- **Integrate with a Platform:** Integrate the chatbot into a chosen platform (e.g., website, mobile app, messaging app) to demonstrate real-world usage.

What We Aim to Achieve with this Project:

- **Demonstrate Conversational AI Skills:** Showcase our ability to design, develop, and deploy a functional conversational AI application.
- **Explore Integration Possibilities:** Investigate how the chatbot can be integrated into different platforms and workflows for practical use.

What is Dialogflow?

Overview of Google Cloud Dialogflow:

Google Cloud Dialogflow is a powerful platform that allows developers to create conversational chatbots and virtual agents. It uses natural language processing (NLP) and machine learning to understand and respond to user queries in a human-like manner.

Why Choose Dialogflow for Chatbot Development?

- **User-Friendly Interface:** Dialogflow provides an intuitive interface for designing conversational flows and training chatbots without extensive coding knowledge.
- **Integration with Google Cloud Services:** It seamlessly integrates with other Google Cloud services like Cloud Functions, allowing for dynamic and personalized responses.
- **Scalability and Reliability:** Dialogflow is hosted on Google Cloud, offering scalability to handle varying loads of user requests and ensuring high availability

Agent Builder Functionality

- **Dialog Management**

  - Intents Creation: Define user intentions for specific actions within the conversation.

  - Entities Organization: Identify and organize data points that can be extracted from user inputs.

- **Context Handling**

  -State Management: Keep track of where the user is in the conversation.

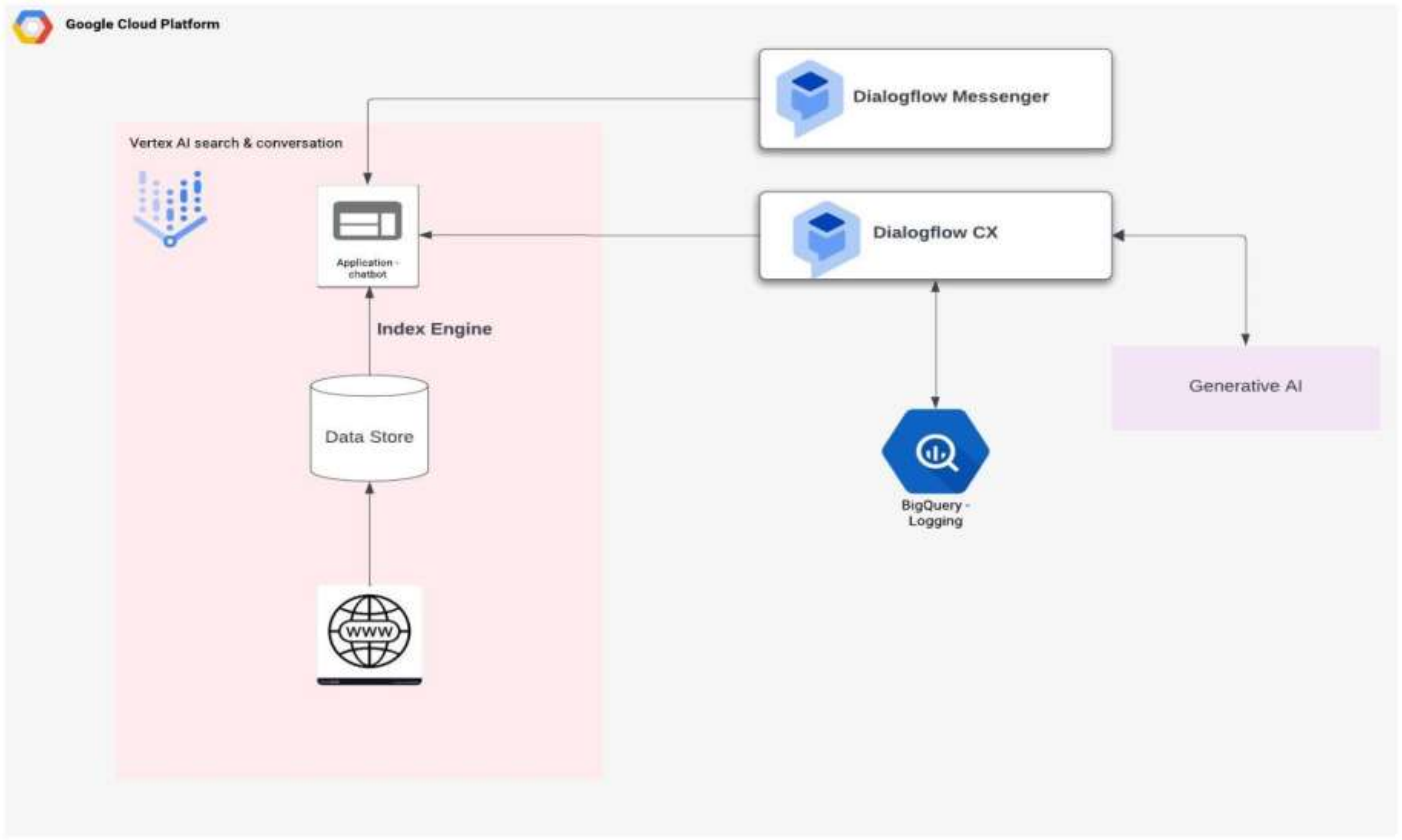
  -Sequence Understanding: Recognize and respond to the sequence of user interactions, maintaining continuity.

- **Fulfillment Configuration**

  - Webhooks Setup: Connect to external APIs, databases, or services to fetch or send data.

  - Logic Execution: Implement complex business logic that the chatbot can execute during conversations.

Architecture Diagram



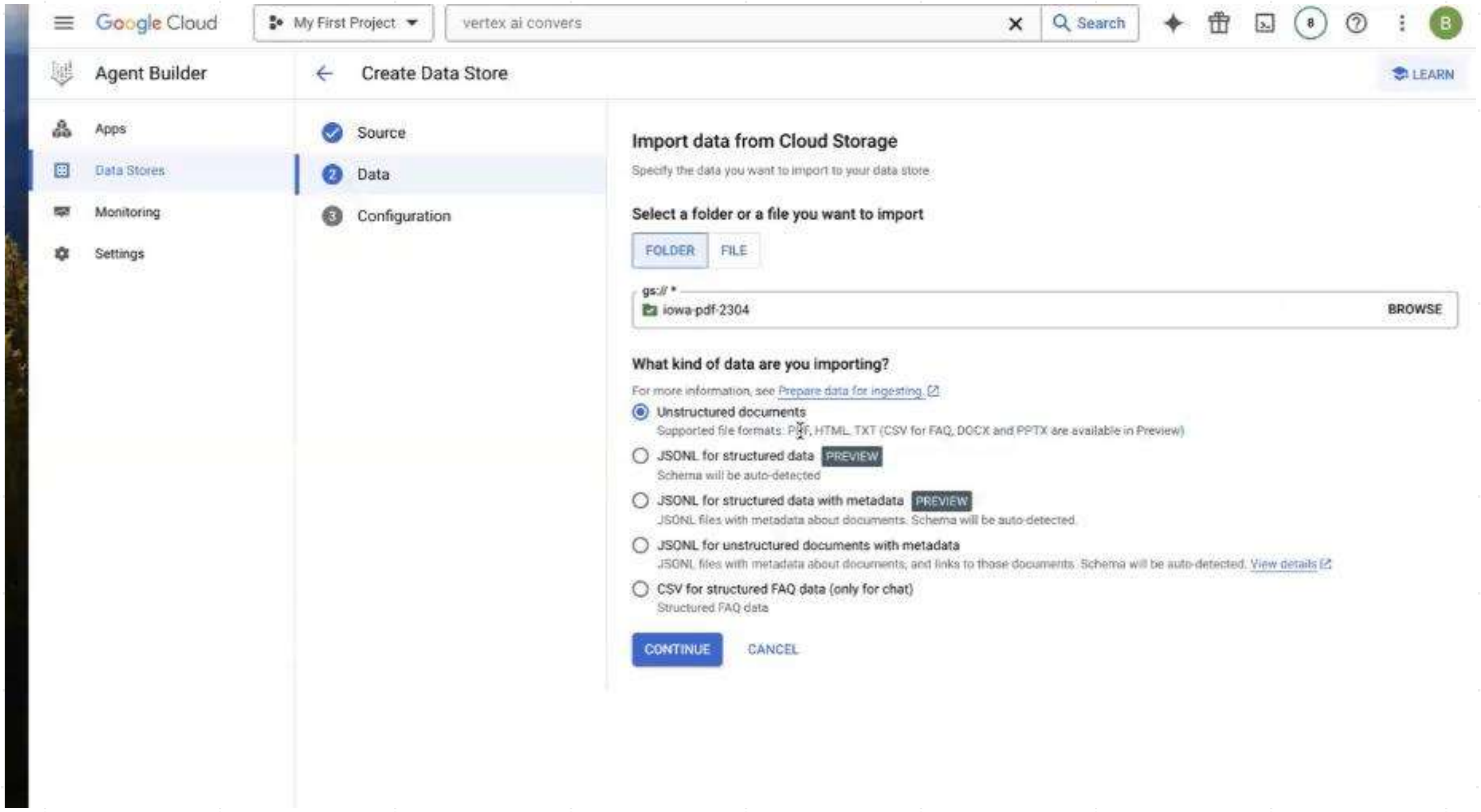
Development :

- First, navigate to the Service Agent Builder to begin our plan for creating the chatbot.
- In the Agent Builder, proceed to the Datastore section, where you'll create a datastore. Choose a source, such as Cloud Storage, and then specify a bucket for storage.
- After storing the data into the bucket, create an app using the chat feature. Select the storage option, pulling data from th e Datastore, and proceed to create the app.
- When you click on the app, it will redirect you to Dialogflow. Here, you can customize the agent settings according to your preferences, or you can stick with the default settings.
- Next, integrate the chatbot with your website. In Dialogflow, utilize the Integration feature. You can integrate the chatbot with various apps, such as Slack. However, in our case, we'll integrate it with Dialogflow Messenger for the website.
- Generate an authorization code snippet and embed it into your HTML code. Save the modified HTML file locally. Clicking on the local file will redirect you to the chatbot without any background webpage since it's opened via the local machine.

Development

- When you click on the app, it will redirect you to Dialogflow. Here, you can customize the agent settings according to your preferences, or you can stick with the default settings.
- Next, integrate the chatbot with your website. In Dialogflow, utilize the Integration feature. You can integrate the chatbot with various apps, such as Slack. However, in our case, we'll integrate it with Dialogflow Messenger for the website.
- Generate an authorization code snippet and embed it into your HTML code. Save the modified HTML file locally. Clicking on the local file will redirect you to the chatbot without any background webpage since it's opened via the local machine.

- Now, using the App Engine, navigate to the folder containing the related HTML file. Deploy your code snippet using a YAML file.
- To establish the connection between the local machine and the Google Cloud ,utilize the Google Cloud SDK . Using the console, deploy the application to Google Cloud Platform (GCP).
- Note : https://iowademo-dot-alert-autumn-416405.uc.r.appspot.com/
- The above link will redirect to the chatbot we have created .



Create Data Store

- Source
- 2 Data
- 3 Configuration

Import data from Cloud Storage

Specify the data you want to import to your data store

Select a folder or a file you want to import

[✓ FOLDER] FILE

gs:// *
iowa-pdf-2304
BROWSE

What kind of data are you importing?

For more information, see Prepare data for ingesting

[✓ Unstructured documents]
Supported file formats: PDF, HTML, TXT (CSV for FAQ, DOCX and PPTX are available in Preview)

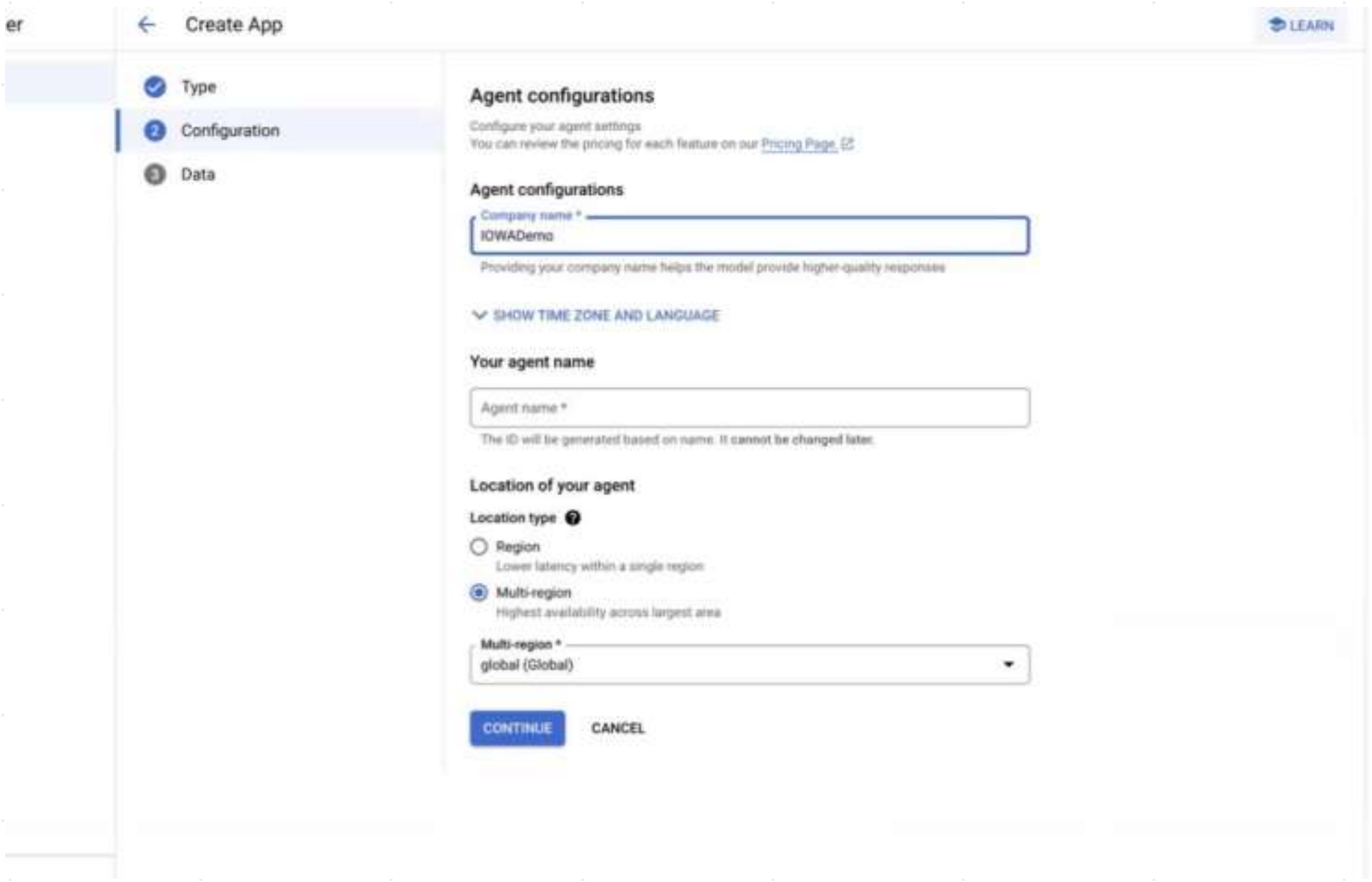
JSONL for structured data PREVIEW
Schema will be auto-detected

JSONL for structured data with metadata PREVIEW
JSONL files with metadata about documents. Schema will be auto-detected.

JSONL for unstructured documents with metadata
JSONL files with metadata about documents, and links to those documents. Schema will be auto-detected. View details

CSV for structured FAQ data (only for chat)
Structured FAQ data

CONTINUE CANCEL



Agent configurations

Configure your agent settings
You can review the pricing for each feature on our Pricing Page.

Agent configurations

Company name *: IOWADemo

Providing your company name helps the model provide higher-quality responses

Your agent name

Agent name *:

The ID will be generated based on name. It cannot be changed later.

Location of your agent

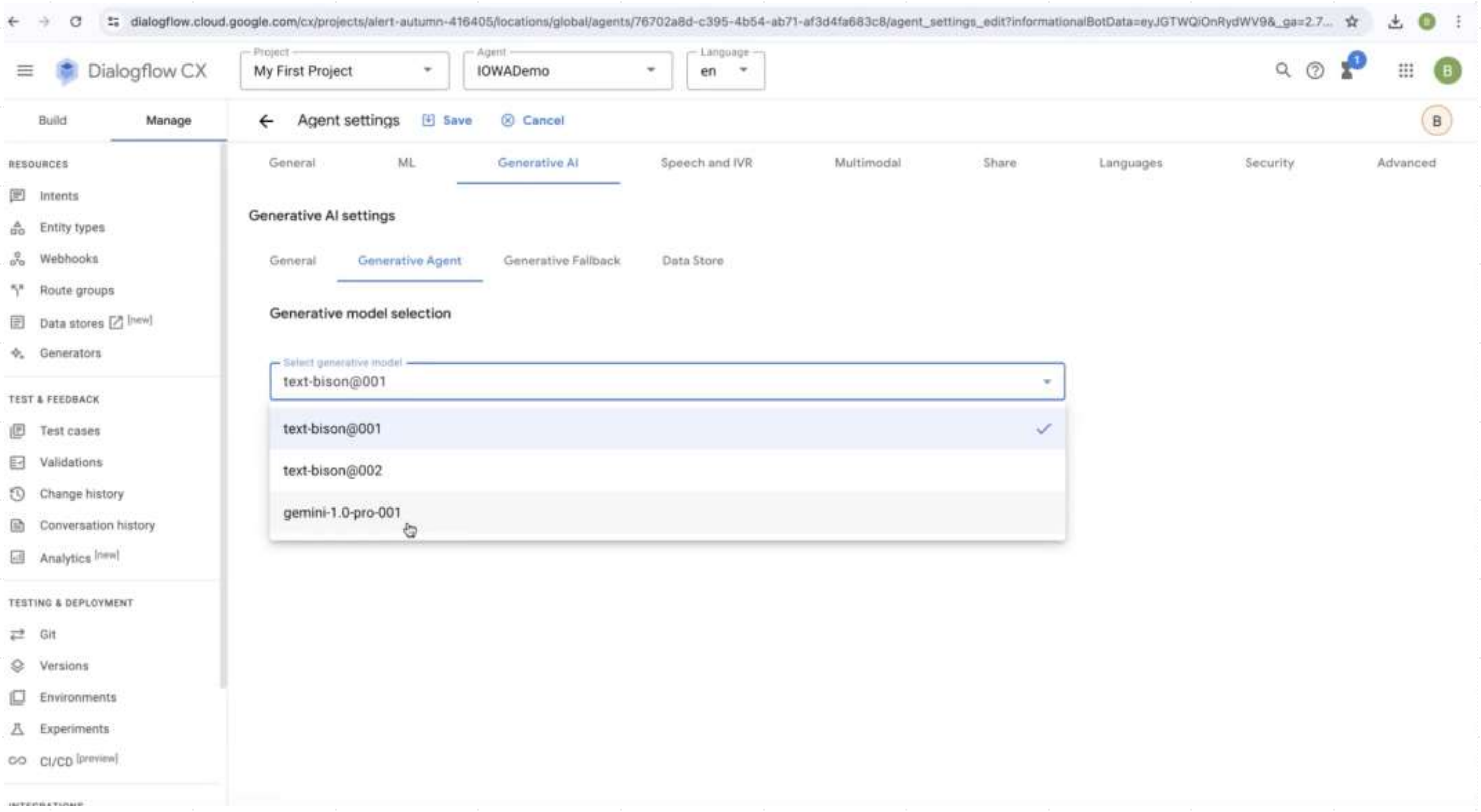
Location type

Region [✓ Multi-region]

Region: Lower latency within a single region
Multi-region: Highest availability across largest area

Multi-region *: global (Global)

Development Process



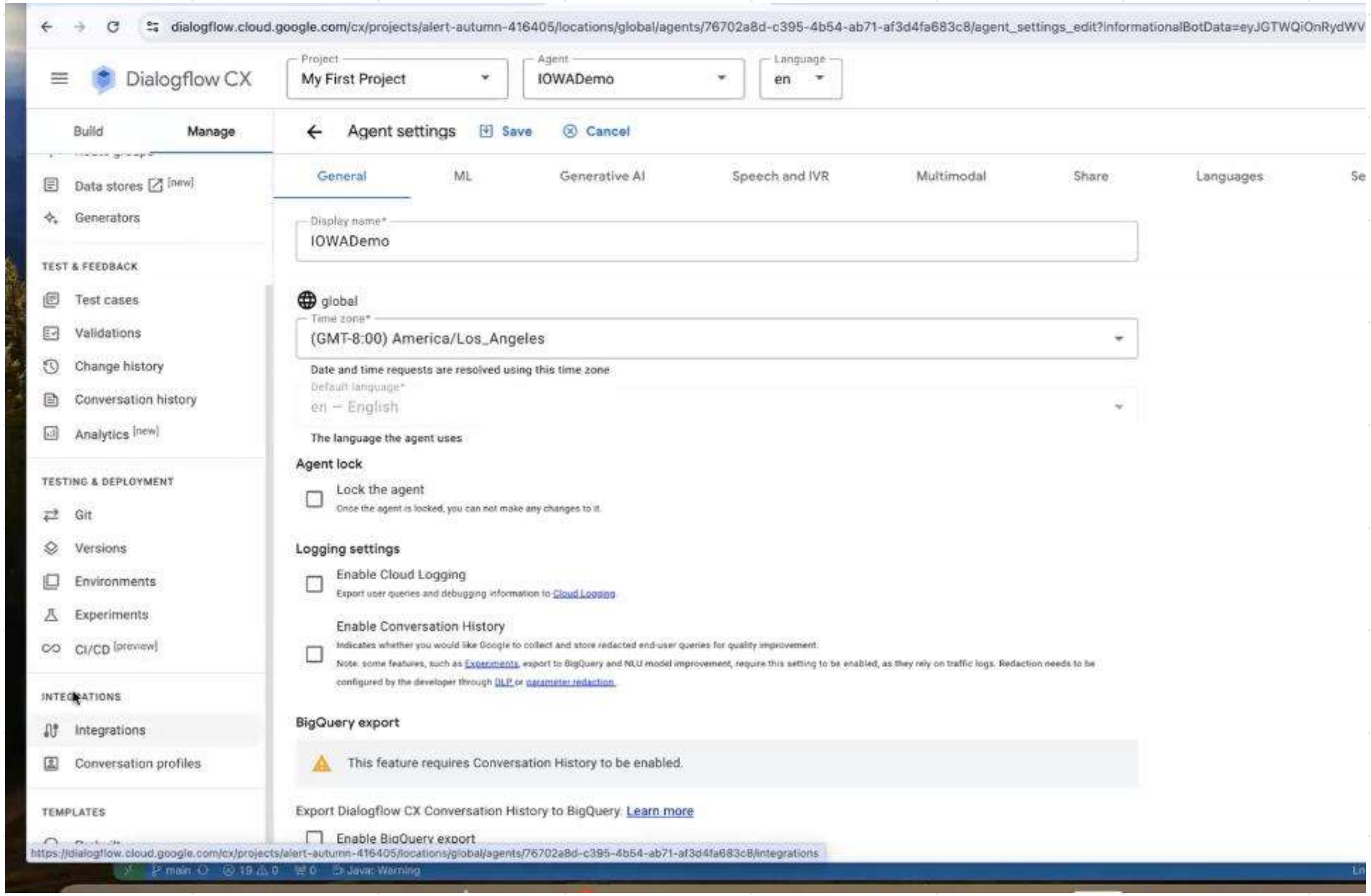
dialogflow.cloud.google.com/cx/projects/alert-autumn-416405/locations/global/agents/76702a8d-c395-4b54-ab71-af3d4fa683c8/agent_settings_edit?informationalBotData=eyJGTWQiOnRydWV9&_ga=2.7...

Generative AI settings

Generative model selection

Select generative model: text-bison@001

- text-bison@001 ✓
- text-bison@002
- gemini-1.0-pro-001



Project: My First Project

Agent: IOWADemo

Language: en

Display name*: IOWADemo

global

Time zone*: (GMT-8:00) America/Los_Angeles

Date and time requests are resolved using this time zone

Default language*: en – English

The language the agent uses

Agent lock

☐ Lock the agent
Once the agent is locked, you can not make any changes to it.

Logging settings

☐ Enable Cloud Logging
Export user queries and debugging information to Cloud Logging

☐ Enable Conversation History
Indicates whether you would like Google to collect and store redacted end-user queries for quality improvement.
Note: some features, such as Experiments, export to BigQuery and NLU model improvement, require this setting to be enabled, as they rely on traffic logs. Redaction needs to be configured by the developer through DLP or parameter redaction.

BigQuery export

⚠ This feature requires Conversation History to be enabled.

Export Dialogflow CX Conversation History to BigQuery. Learn more

☐ Enable BigQuery export

https://dialogflow.cloud.google.com/cx/projects/alert-autumn-416405/locations/global/agents/76702a8d-c395-4b54-ab71-af3d4fa683c8/integrations

Development Process

Steps Involved in Developing the Chatbot

1. Designing Conversation Flows

- **Define User Scenarios:** Identify typical user interactions and potential conversation paths.
- **Create Dialogflow Intents:** Map out different user intents (what users want) and corresponding responses.

2. **Configuring Fulfillment for Dynamic Responses**

- **Set up Fulfillment:** Connect Dialogflow with backend services (e.g., Google Cloud Functions, webhooks) to handle user requests.
- **Integrate External APIs:** Access external resources (e.g., databases, third-party APIs) to fetch or update information based on user requests.

Integration and Testing

1. **Integration with Messaging Platforms:**

- **Web Chat:** Embed the Dialogflow chatbot widget on a website for real-time interactions.
- **Mobile Apps:** Integrate Dialogflow into mobile applications to enable chatbot functionality.
- **Messaging Apps:** Connect Dialogflow with messaging platforms like Facebook Messenger, Slack, or WhatsApp.

2. Integration with Backend Services:

- **Google Cloud Functions:** Utilize serverless functions to handle chatbot requests and provide dynamic responses.
- **Webhooks:** Connect Dialogflow with custom web services or APIs to retrieve or update information as needed.

3. Unit Testing:

- **Test Intents and Entities:** Validate that Dialogflow correctly recognizes and responds to predefined intents and entities.
- **Mock Responses:** Simulate user inputs to ensure the chatbot generates accurate responses.

4. Integration Testing:

- **End-to-End Testing:** Validate the entire chatbot flow, including user interactions, intent recognition, and backend integration.
- **Test Across Platforms:** Ensure consistent behavior and performance across different platforms and messaging channels

Results and Performance

Key Metrics for Evaluating Chatbot Performance:

1. Response Effectiveness:

- **Response Time:** Monitor the average time taken by the chatbot to respond to user queries.
- **Completeness of Responses:** Assess whether the chatbot's responses adequately address user queries or requests.

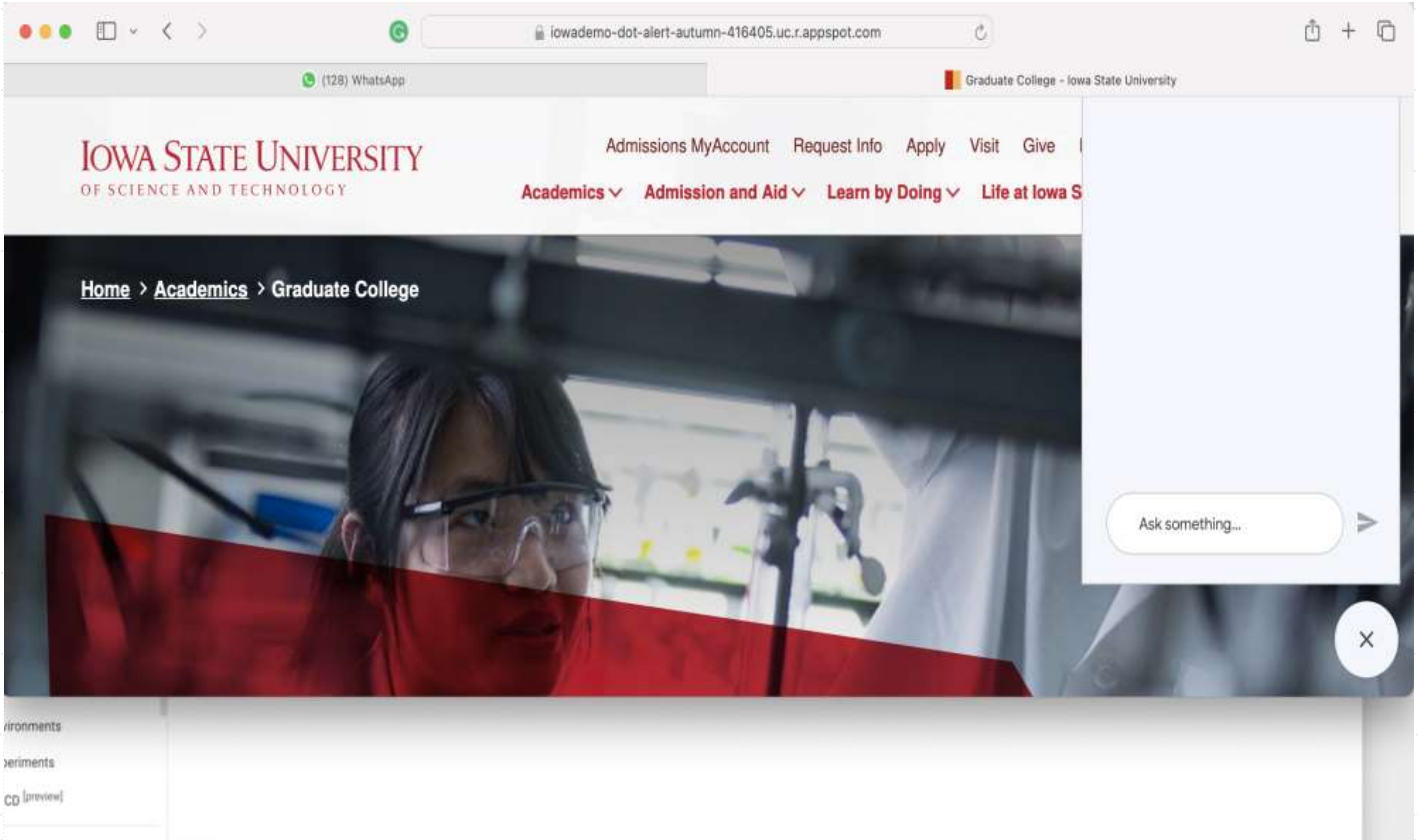
2. Conversation Flow:

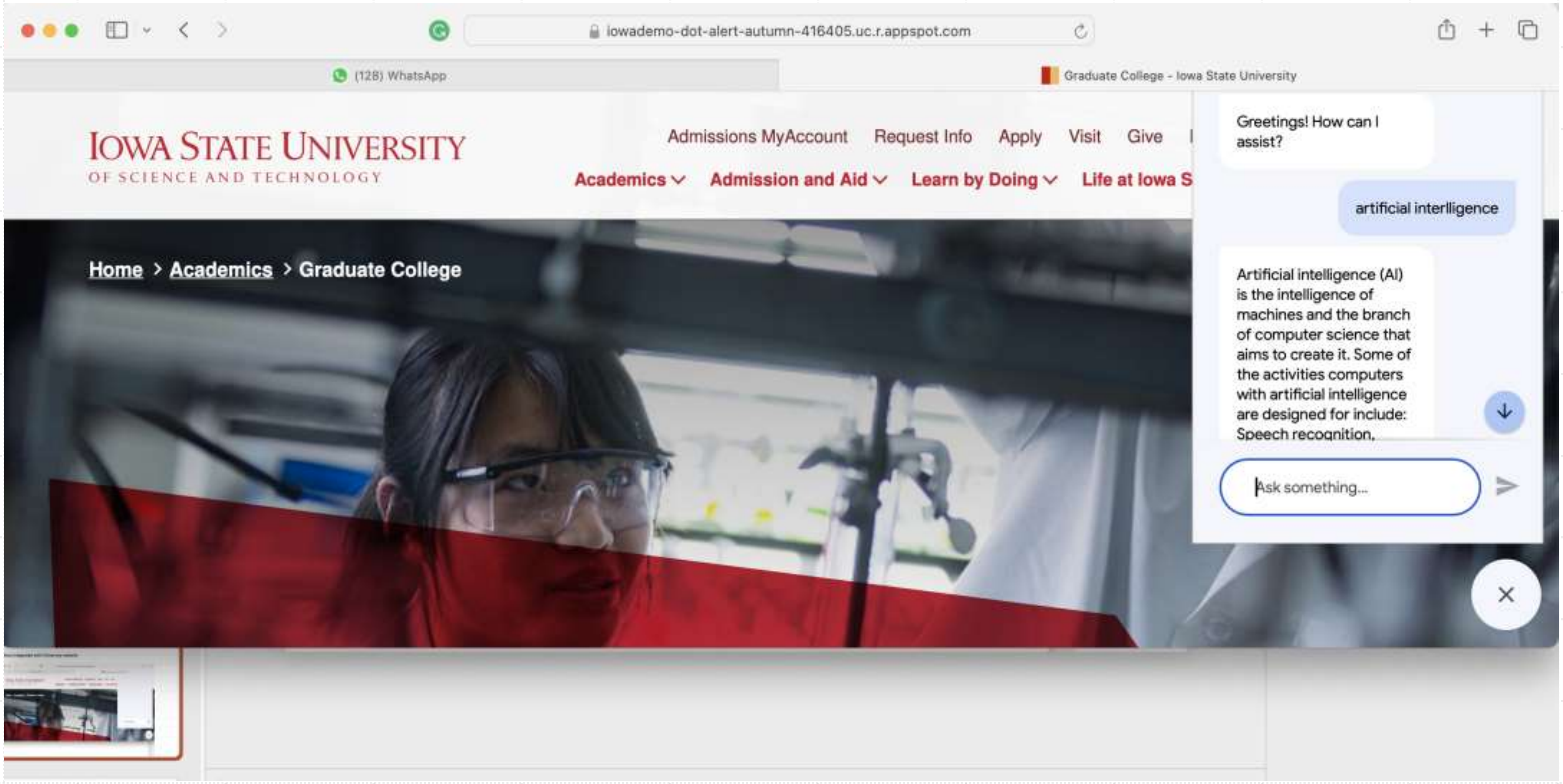
- **Context Management:** Evaluate the chatbot's ability to maintain context and continuity in conversations across multiple messages.
- **Fallback Handling:** Measure how well the chatbot handles unexpected or ambiguous user inputs.

3.Evaluating Results and Iteration

- **Continuous Improvement:** Use collected metrics and user feedback to iteratively improve the chatbot's performance and user experience.
- **Adjustment of Dialogflow Settings:** Fine-tune Dialogflow configurations (e.g., intents, entities, fallback responses) based on observed performance and feedback.

Chatbot integrated with University website





iowademo-dot-alert-autumn-416405.uc.r.appspot.com

(128) WhatsApp

Graduate College - Iowa State University

Iowa State University
of Science and Technology

Admissions MyAccount Request Info Apply Visit Give

Academics Admission and Aid Learn by Doing Life at Iowa S

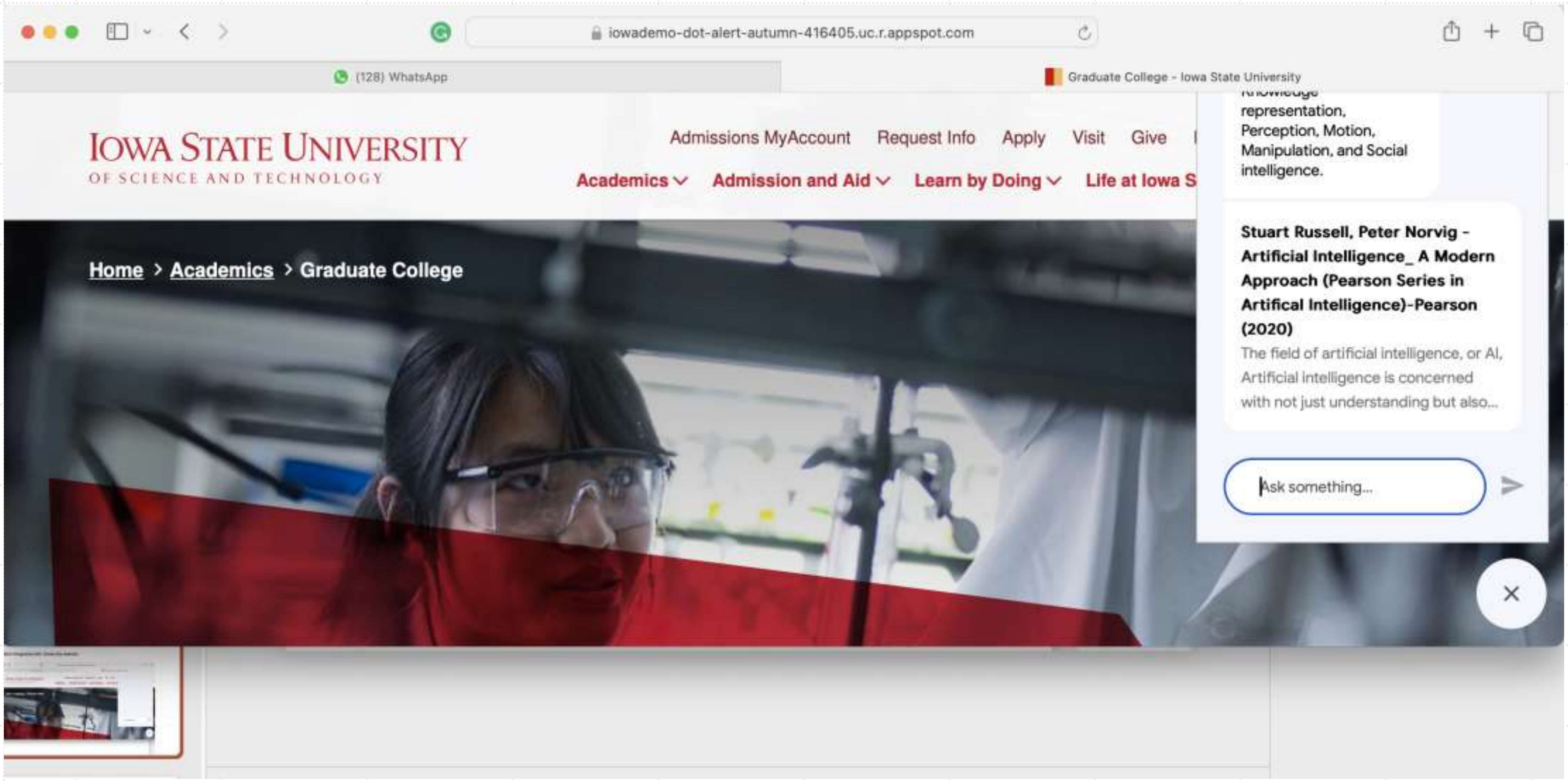
Home > Academics > Graduate College

Greetings! How can I assist?

artificial interlligence

Artificial intelligence (AI) is the intelligence of machines and the branch of computer science that aims to create it. Some of the activities computers with artificial intelligence are designed for include: Speech recognition,

Ask something...


iowademo-dot-alert-autumn-416405.uc.r.appspot.com

representation, Perception, Motion, Manipulation, and Social intelligence.

Stuart Russell, Peter Norvig - Artificial Intelligence_ A Modern Approach (Pearson Series in Artifical Intelligence)-Pearson (2020)

The field of artificial intelligence, or AI, Artificial intelligence is concerned with not just understanding but also...

Future Enhancements

1. **Advanced Natural Language Understanding (NLU):**Implement more sophisticated NLU models to improve intent identification and understanding of complex user queries.

2. **Personalization and Contextualization:** Implement dynamic content generation based on user history or preferences to enhance user engagement.

3. **Multilingual Support and Localization:** Extend chatbot capabilities to support multiple languages to cater to a global audience.

4. **Handling Increased User Load:** Optimize the chatbot architecture to handle increased user traffic and concurrent interactions without performance degradation

5. **Integration with Additional Services:** Integrate the chatbot with more external services and APIs to provide a wider range of functionalities (e.g., payment processing, content retrieval).

Conclusion

Summary of Achievements:

- **Functional Chatbot Development:** Successfully developed a conversational chatbot using Google Cloud Dialogflow that can understand user intents and provide relevant responses.
- **Integration and Deployment:** Integrated the chatbot with various platforms (e.g., website, mobile app) and deployed it for real-world usage.
- **Performance Optimization:** Improved chatbot performance through continuous testing, iteration, and enhancements based on user feedback.

Importance of Conversational AI in Modern Applications:

- **Enhanced User Interaction:** Conversational AI enables more natural and intuitive interactions between users and applications, improving accessibility and usability.
- **24/7 Availability:** Chatbots provide round-the-clock support and assistance, enhancing customer service and user engagement.