**A Chatbot with Microsoft Azure**

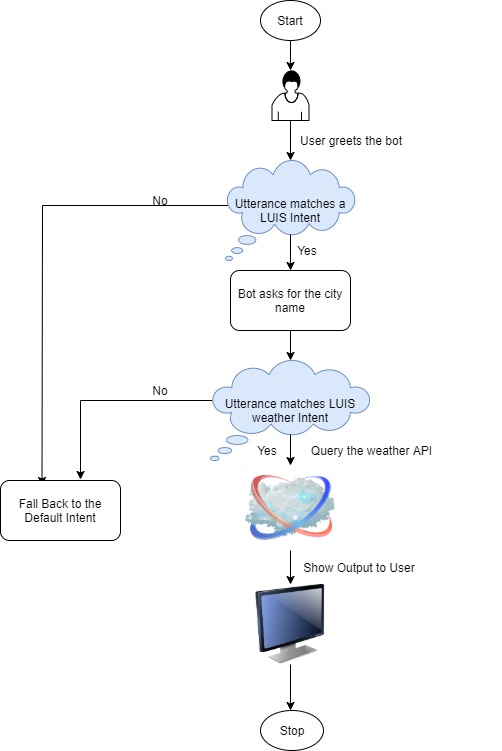
1. **Introduction:**

A chatbot is an application that can initiate and continue a conversation using auditory and/or textual methods as a human would do. A chatbot can be either a simple rule-based engine or an intelligent application leveraging Natural Language Understanding. Many organizations today have started using chatbots extensively. Chatbots are becoming famous as they are available 24\*7, provide a consistent customer experience, can handle several customers at a time, are cost-effective and hence, result in a better overall customer experience.

* 1. **Prerequisites**

The prerequisites for developing and understanding a chatbot using Microsoft Azure are:

* An Azure account.
* A fundamental understanding of python and flask
  1. **Application Architecture**

****

1. **Implementation using Azure LUIS**
   1. **Create a LUIS app**

* Begin with creating the app, Once created, open your app, select build, and click ‘Create new intent’ to create a new intent.

A screenshot of a computer screen

Description automatically generated

* Select build and Create new intents by entering the names for intents by open your app.

A screenshot of a computer screen

Description automatically generated

* Then, entered user utterances and entities, and finally, trained the app. Therefore, click ‘train’ to train the LUIS app

A screenshot of a computer screen

Description automatically generated

* Click ‘Test’ to test the intent and see the confidence of the app for various utterances. Now, Go to the Azure Resources section and copy the Primary key. It will serve as the LUIS API KEY.

A screenshot of a social media post

Description automatically generated

1. **Create python app and integrating with LUIS app**

After creating a folder for chatbot with a name weather chatbot using luis app, I opened it in pycharm. Create a file called app.py and put the code inside.

A screenshot of a computer screen

Description automatically generated

Create different folders logger, luis, weather and create logger.py, luisApp.py, weatherApp.pyA screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Create a file config.ini and write all the details

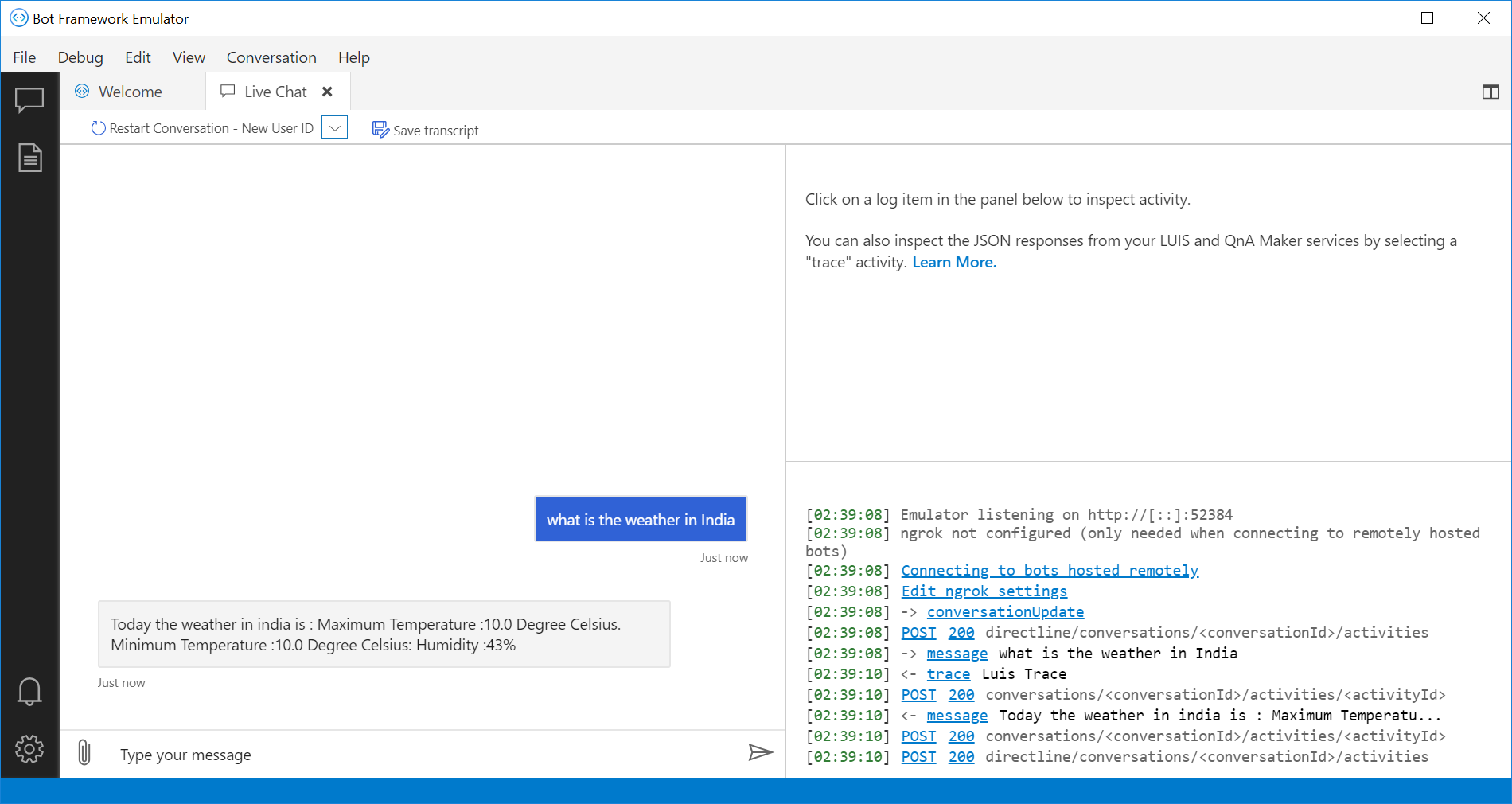
A screenshot of a cell phone

Description automatically generated

1. **Testing app.py on bot emulator**

**A screenshot of a computer

Description automatically generated**



1. **Deploy to azure**

* Go to azure portal dashboard and create resources. Here I have created web application and python bot registration.

A screenshot of a computer

Description automatically generated

Search web app and Go to web app and complete required information such as app name, resource group , runtime stack , region, select 1 GB size and click on review and create to create web app.

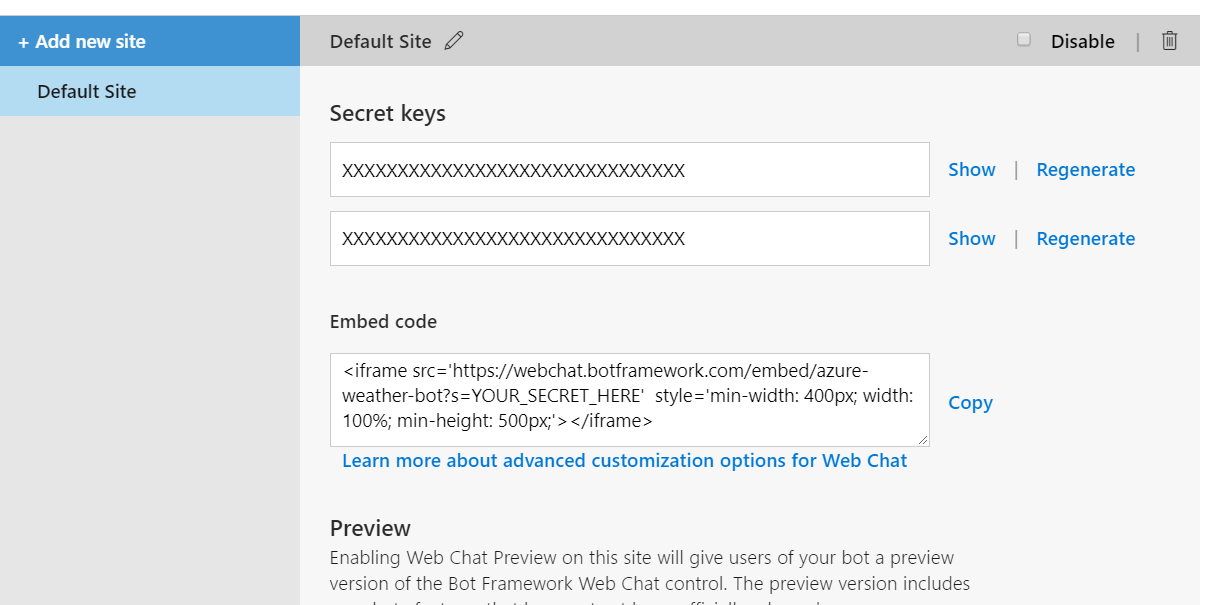
A screenshot of a cell phone

Description automatically generated

Once the deployment is completed, open the app and go to the ‘Deployment Center’ option. Select ‘local git’ for source control and click continue.

Go to the channels section of your bot.

The bot can be deployed as an embedding to an existing HTML page by selecting the get bot embedded code option



* Once deployment is complete, go to resources, Select deployment center and select local git

A screenshot of a social media post

Description automatically generated

Then select kudo ‘App service build provider’ as the build provider and click continue. Click finish.

A screenshot of a cell phone

Description automatically generated

Copy git clone url

A screenshot of a social media post

Description automatically generated

* Open a command prompt and navigate to your project folder.
* Run git init to initialize an empty git repository
* Create a new remote git alias using the command: git remote add <alias> <git clone url>
* Use git add . to add all the files to the local git repository.
* Use git commit –m “First Commit” to commit the code to the git repo.
* Push the code to the remote repo using git push <alias> master –f
* This prompts for a username and password. Go to the ‘Deployment Credentials’ section and copy the username and password to enter in the prompt.