GOVERNMENT COLLEGE OF ENGINEERING BARGUR

( AUTONOMOUS)

## PROJECT TITLE: Image Recognition with IBM Cloud Visual Recognition

TEAM MEMBERS:

ELAMATHY O G ( 61072111108)

PADMAPRIYA T (61072111135)

SATHYAPRIYA R (61072111142)

VIJAYALAKSHMI R (61072111149)

PROBLEM STATEMENT:

The project involves creating an image recognition system using IBM Cloud Visual Recognition. The goal is to develop a platform where users can upload images, and the system accurately classifies and describes the image contents. This will enable users to craft engaging visual stories with the help of AI-generated captions, enhancing their connection with the audience through captivating visuals and compelling narratives.

## DESIGN THINKING APPROACH

### 1. Image Recognition Setup

**Objective:** Set up the IBM Cloud Visual Recognition service and obtain the necessary API keys.

* **Action Steps:**
  + Access the IBM Cloud Visual Recognition service.
  + Create an IBM Cloud account if not already done.
  + Acquire API keys and credentials.
  + Configure access to the service within the project.

### 2. User Interface

**Objective:** Design a user-friendly interface for users to upload images and view the AI-generated captions.

* **Action Steps:**
  + Collaborate with UI/UX designers to create an intuitive and aesthetically pleasing interface.
  + Implement user-friendly features for image upload, caption display, and user interactions.
  + Ensure responsiveness for various devices (desktop, mobile, tablet).

### 3. Image Classification

**Objective:** Implement the image classification process using the IBM Cloud Visual Recognition API.

* **Action Steps:**
  + Integrate the IBM Cloud Visual Recognition API into the system's backend.
  + Develop algorithms for image preprocessing and analysis.
  + Establish a robust connection to the IBM Cloud Visual Recognition service.

### 4. AI-Generated Captions

**Objective:** Integrate natural language generation to create captions for the recognized images.

* **Action Steps:**
  + Identify and integrate a suitable NLG library or service (e.g., GPT-3, BERT).
  + Develop algorithms for generating captions based on image content.
  + Ensure the captions are coherent, contextually relevant, and grammatically correct.

### 5. User Engagement

**Objective:** Design features to allow users to explore, save, and share their AI-enhanced images.

* **Action Steps:**
  + Implement options for users to save and download AI-enhanced images.
  + Create sharing features for various social media platforms.
  + Develop a gallery or portfolio functionality for users to explore their creations.

### 6. Security and Privacy

**Objective:** Ensure robust data security and privacy measures for user-uploaded images.

* **Action Steps:**
  + Implement secure image storage and access control mechanisms.
  + Educate users about data privacy and obtain necessary consents.
  + Comply with data protection regulations (e.g., GDPR, CCPA).

**TOOL USED:**

IBM Watson Visual Recognition

**WORKING:**

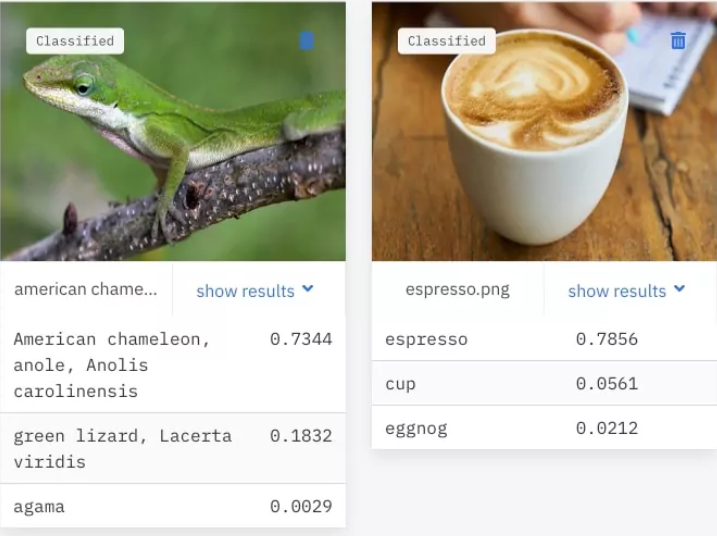
1. **IBM Watson Visual Recognition:** This service can be used for image recognition and description. It can recognize objects, scenes, and concepts within images and provide textual descriptions.
2. **IBM Watson Natural Language Understanding:** To analyze the sentiment and emotions conveyed by the images, you can use IBM Watson Natural Language Understanding. This service can process text data, so you would need to convert the image descriptions or labels generated by Watson Visual Recognition into text. Natural Language Understanding can then analyze the text for sentiment and emotion. It can detect emotions such as joy, sadness, anger, etc., and provide sentiment scores like positive, negative, or neutral.

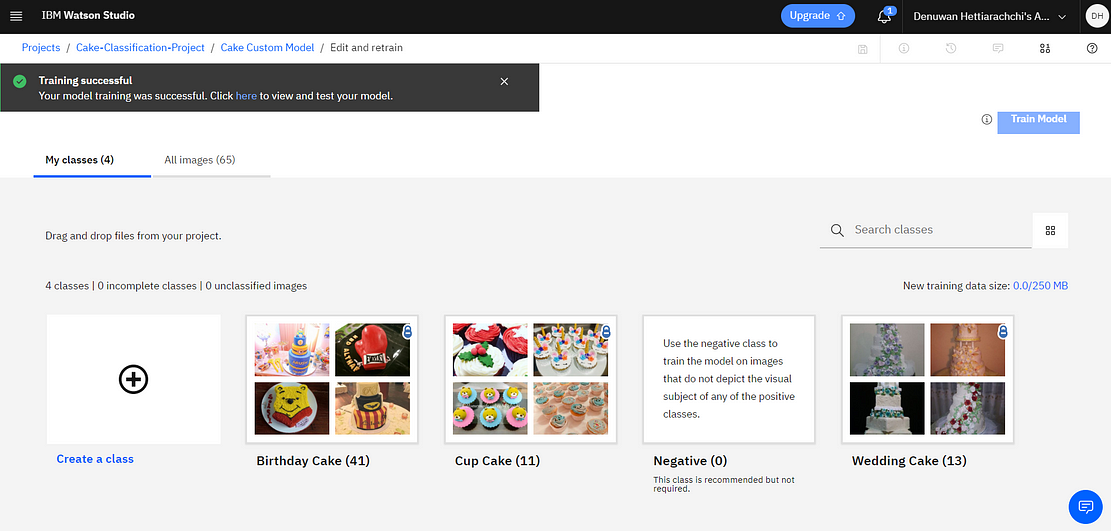
Here's how you might use these services together:

1. **Image Processing:** Upload the image to IBM Watson Visual Recognition for image recognition and description. This will provide you with text-based descriptions of what is in the image.
2. **Text Analysis:** Take the descriptions or labels generated by Watson Visual Recognition and pass them to IBM Watson Natural Language Understanding for sentiment and emotion analysis. This service will analyze the text to determine the emotional tone and sentiment expressed in the image descriptions.

By combining these two IBM Watson services, you can gain insights into both the visual content of the images and the emotional context they convey. This can be useful in various applications, such as social media sentiment analysis, brand monitoring, and content analysis, where understanding the emotions and sentiments associated with visual content is essential.

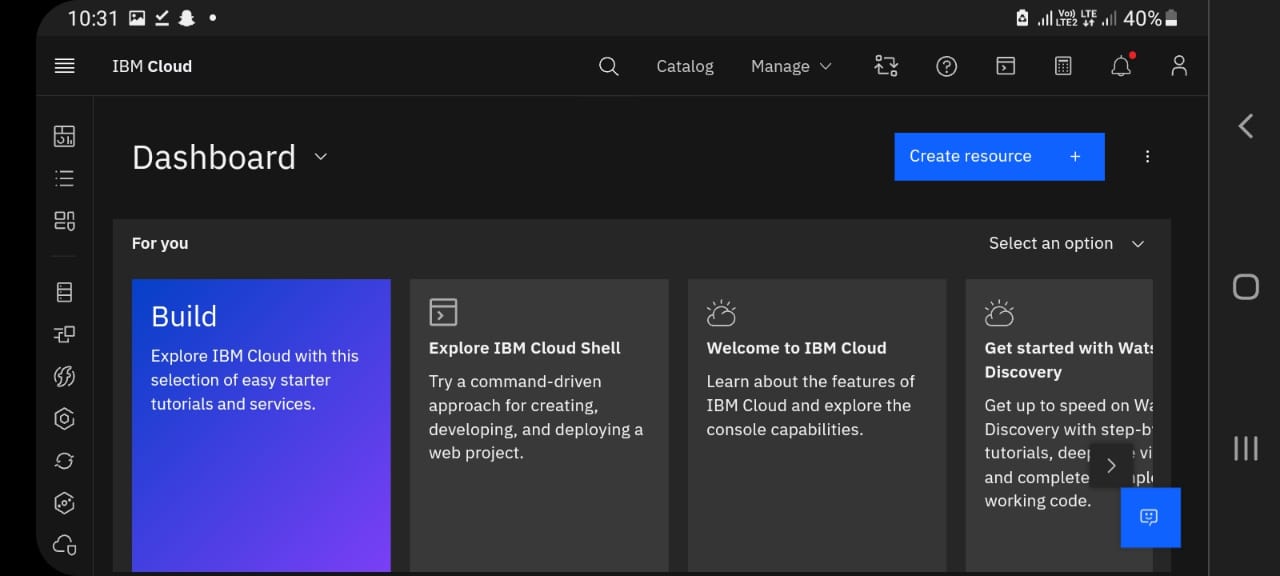
**EXAMPLE:**

****



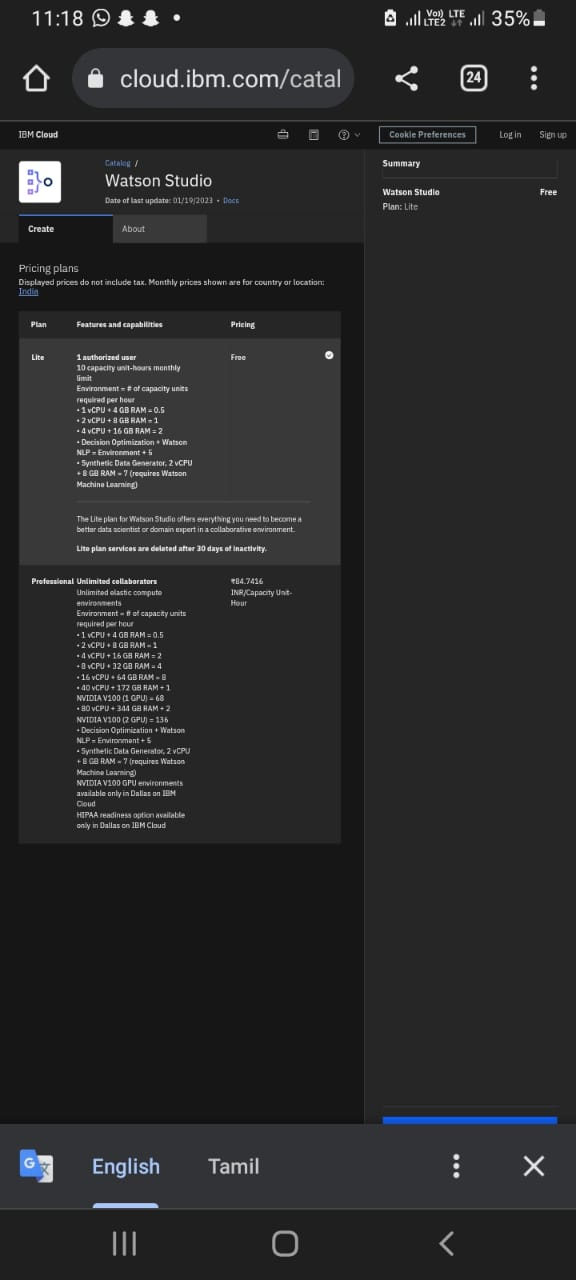
**TO LOAD AND PREPROCESSING THE DATASET:**

1. **Create an IBM Cloud Account:**



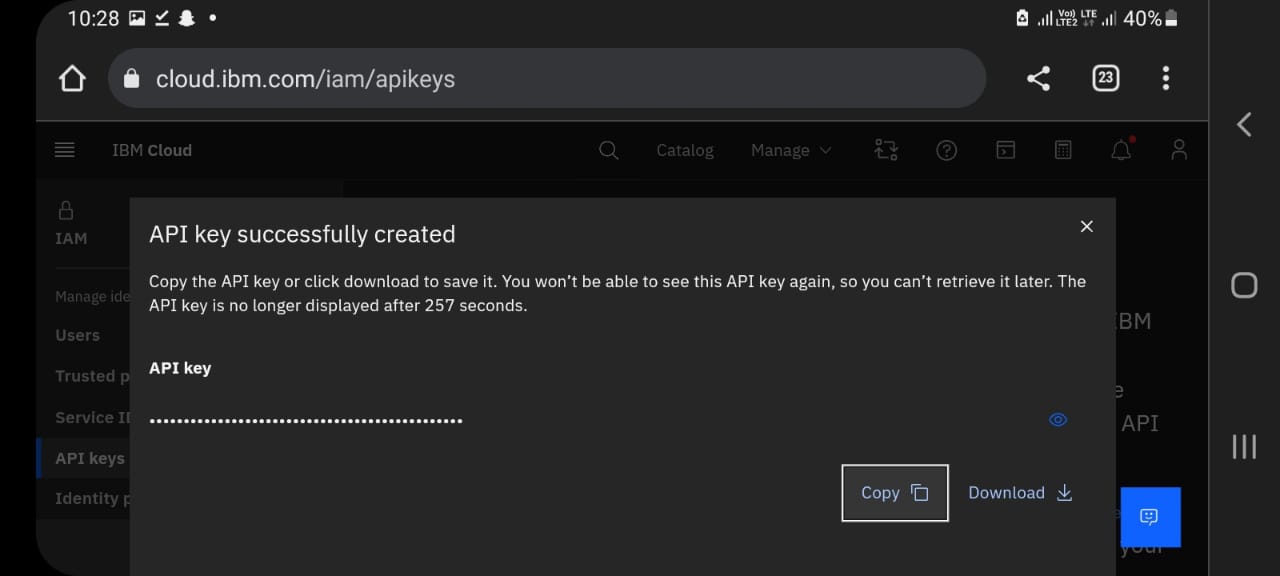
**2. Set Up the Visual Recognition Service:**

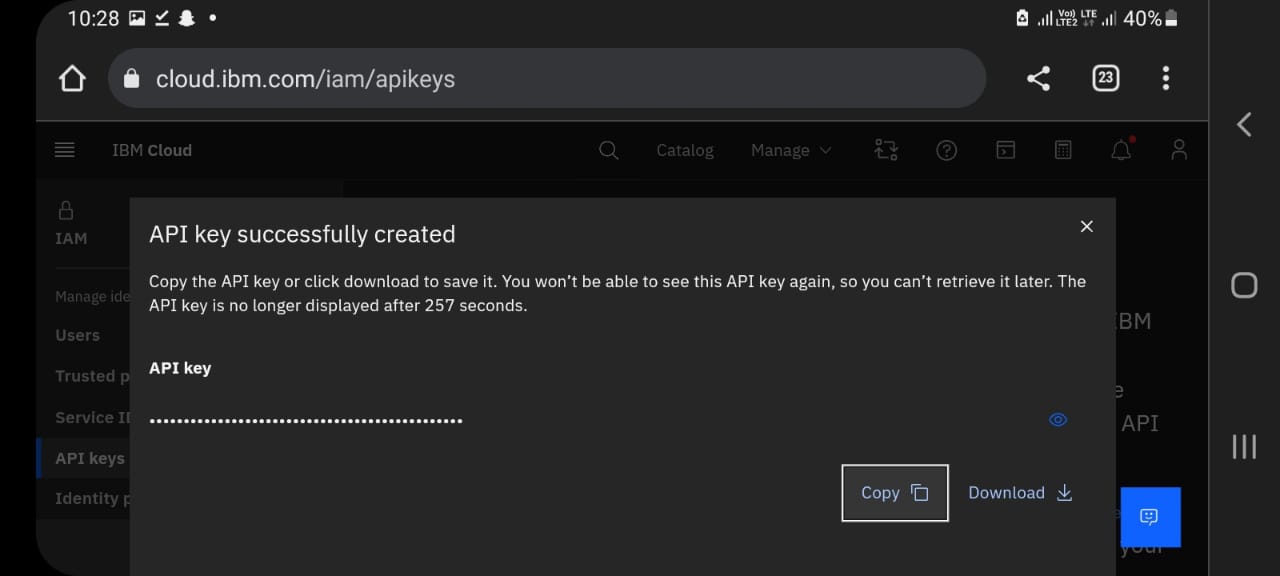
* Log in to your IBM Cloud account.
* Go to the IBM Cloud Catalog.
* Search for "Visual Recognition" and create an instance of the service.

****

**3. Obtain API Keys:**

To obtain API keys for the Visual Recognition service, follow the steps mentioned earlier in this conversation (see the "how to obtain an API key using IBM cloud account" section).

Top of Form



**4.** **DESIGN A SIMPLE WEB INTERFACE:**

You can create a basic web interface for users to upload images and view AI-generated captions. Here's a simple example using HTML, JavaScript, and the Visual Recognition service.

**PROGRAM:**

<!DOCTYPE html>

<html>

<head>

<title>Image Recognition with IBM Visual Recognition</title>

</head>

<body>

<h1>Image Recognition with IBM Visual Recognition</h1>

<input type="file" id="imageInput" accept="image/\*">

<button onclick="uploadImage()">Upload Image</button>

<div id="result">

<h2>AI-Generated Caption:</h2>

<p id="caption"></p>

<img id="uploadedImage" style="max-width: 400px;" />

</div>

<script>

function uploadImage() {

var imageInput = document.getElementById('imageInput');

var resultDiv = document.getElementById('result');

var captionDiv = document.getElementById('caption');

var uploadedImage = document.getElementById('uploadedImage');

var file = imageInput.files[0];

if (file) {

var formData = new FormData();

formData.append('image', file);

// Send the image to IBM Visual Recognition for classification

// Use the API keys and endpoints you obtained earlier

var apiKey = 'YOUR\_API\_KEY';

var apiUrl = 'YOUR\_API\_URL';

fetch(apiUrl, {

method: 'POST',

headers: {

'Accept': 'application/json',

'Authorization': 'Basic ' + btoa('apikey:' + apiKey),

},

body: formData

})

.then(response => response.json())

.then(data => {

// Display the AI-generated caption

captionDiv.innerText = data.description.captions[0].text;

// Display the uploaded image

uploadedImage.src = URL.createObjectURL(file);

})

.catch(error => console.error('Error:', error));

} else {

resultDiv.innerHTML = '<p>Please select an image to upload.</p>';

}

}

</script>

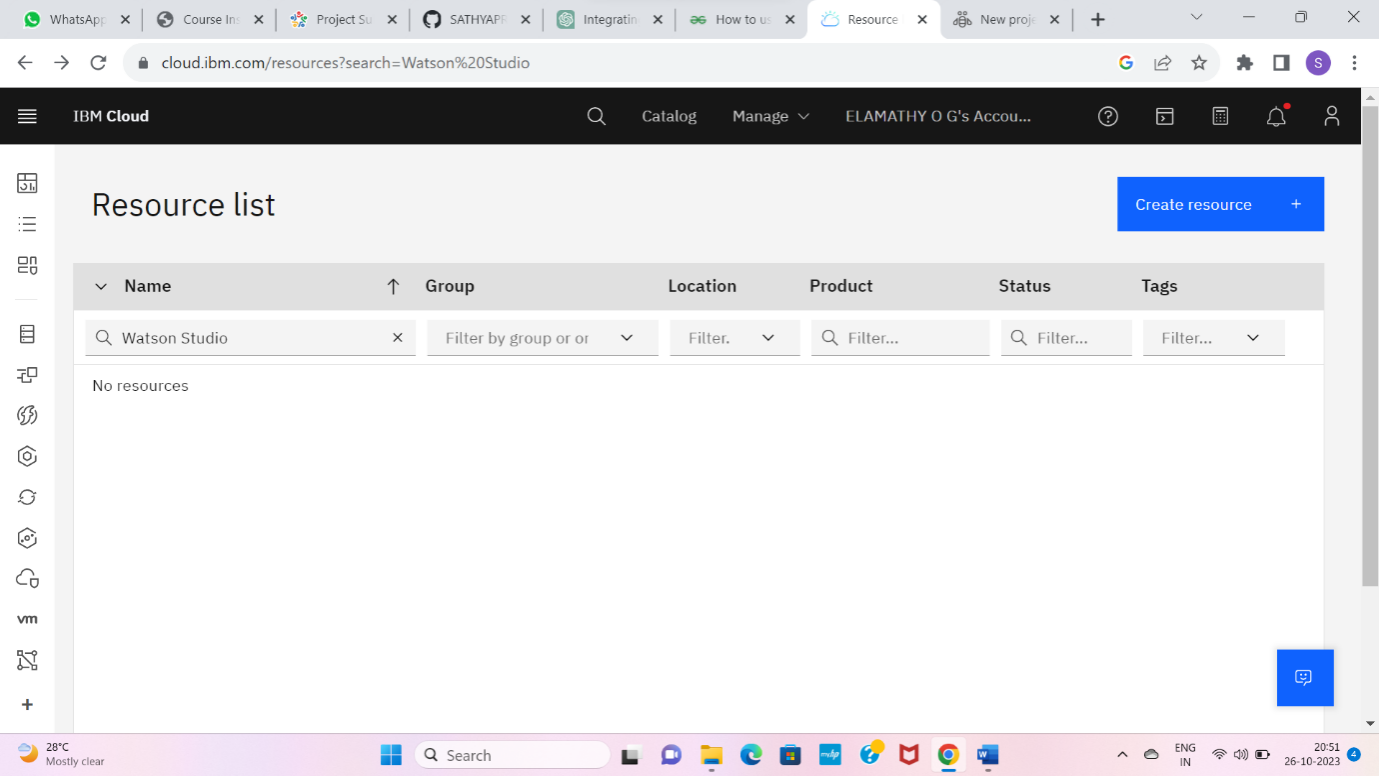
</body>

</html>

BUILDING UP THE WATSON STUDIO :

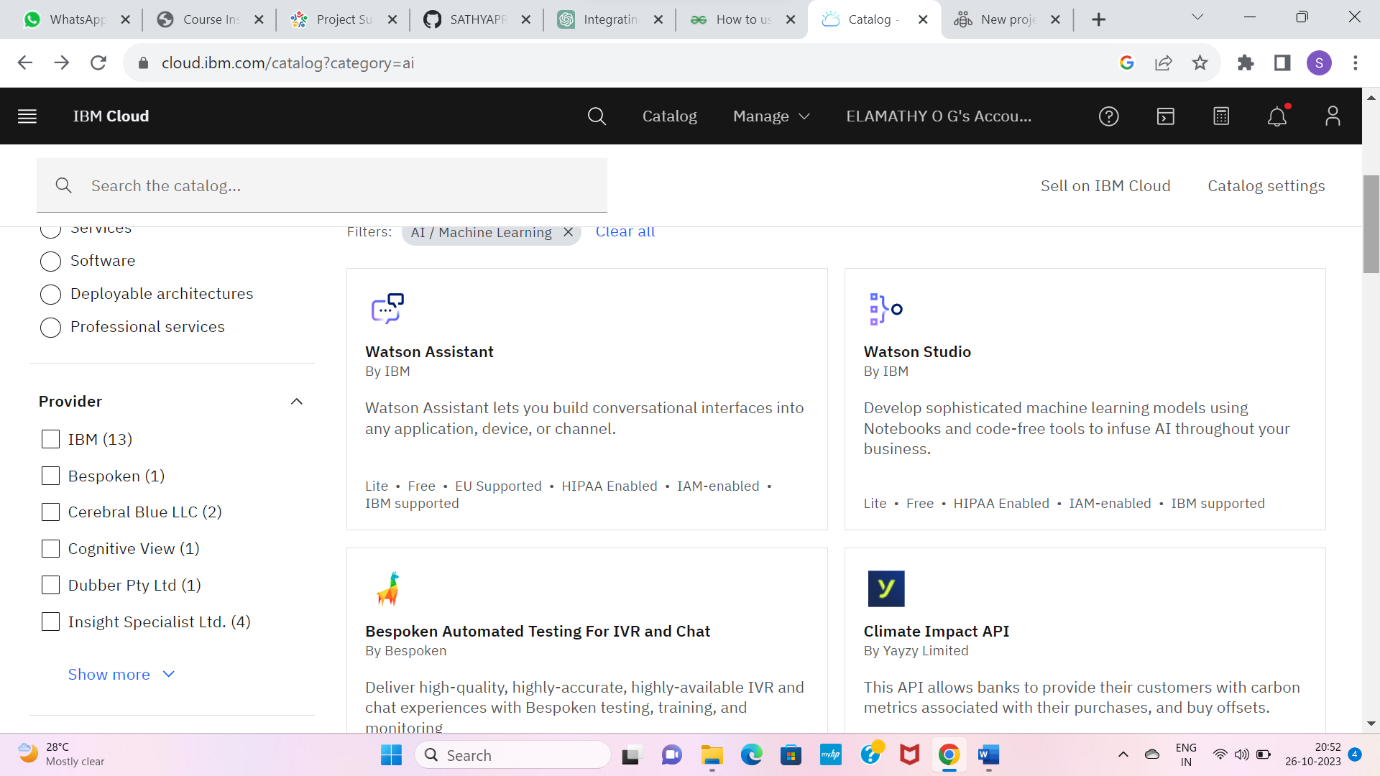
**Step : Create a New Resource**

On your dashboard page, click on the **Create a resource** on the top right to create a new source.



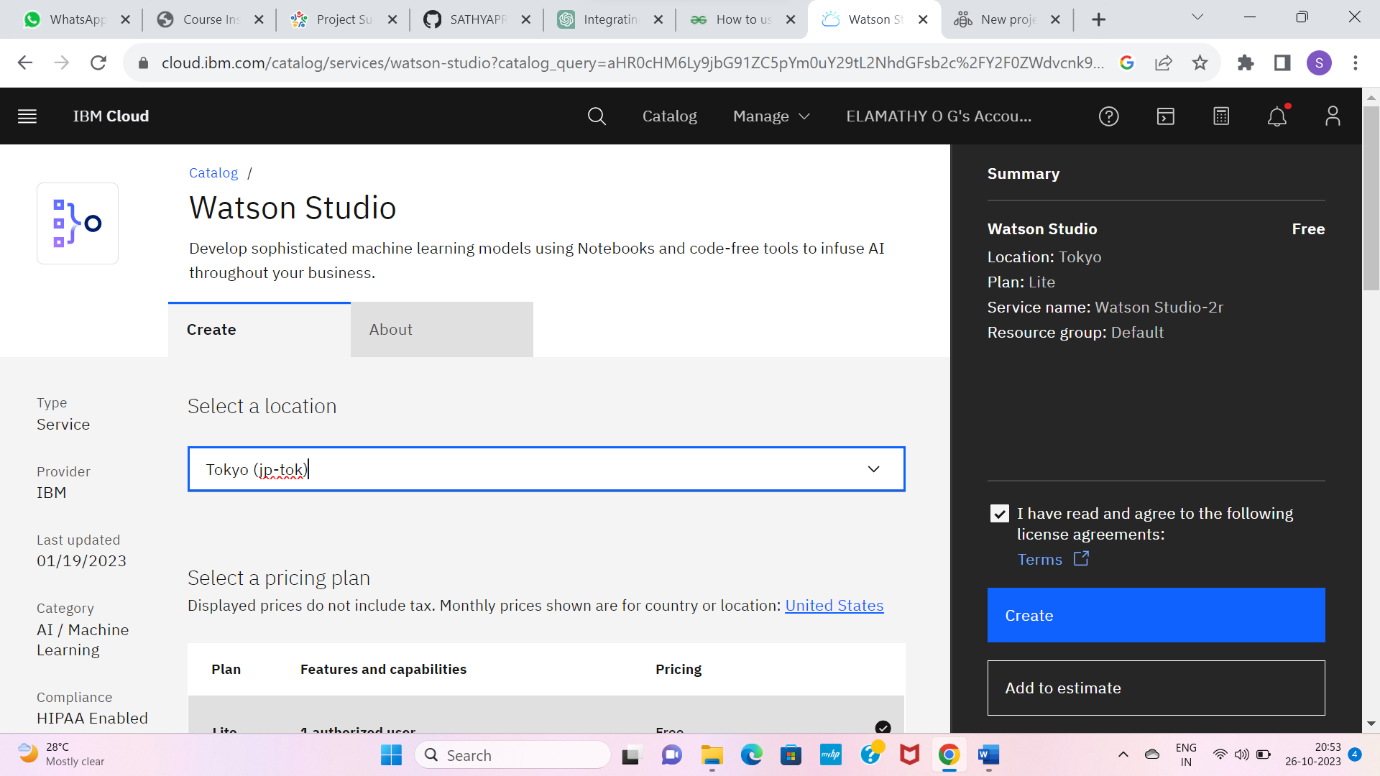


On the Catalog page, select the **AI** category from the left pane, and then select the **Watson Studio** resource.

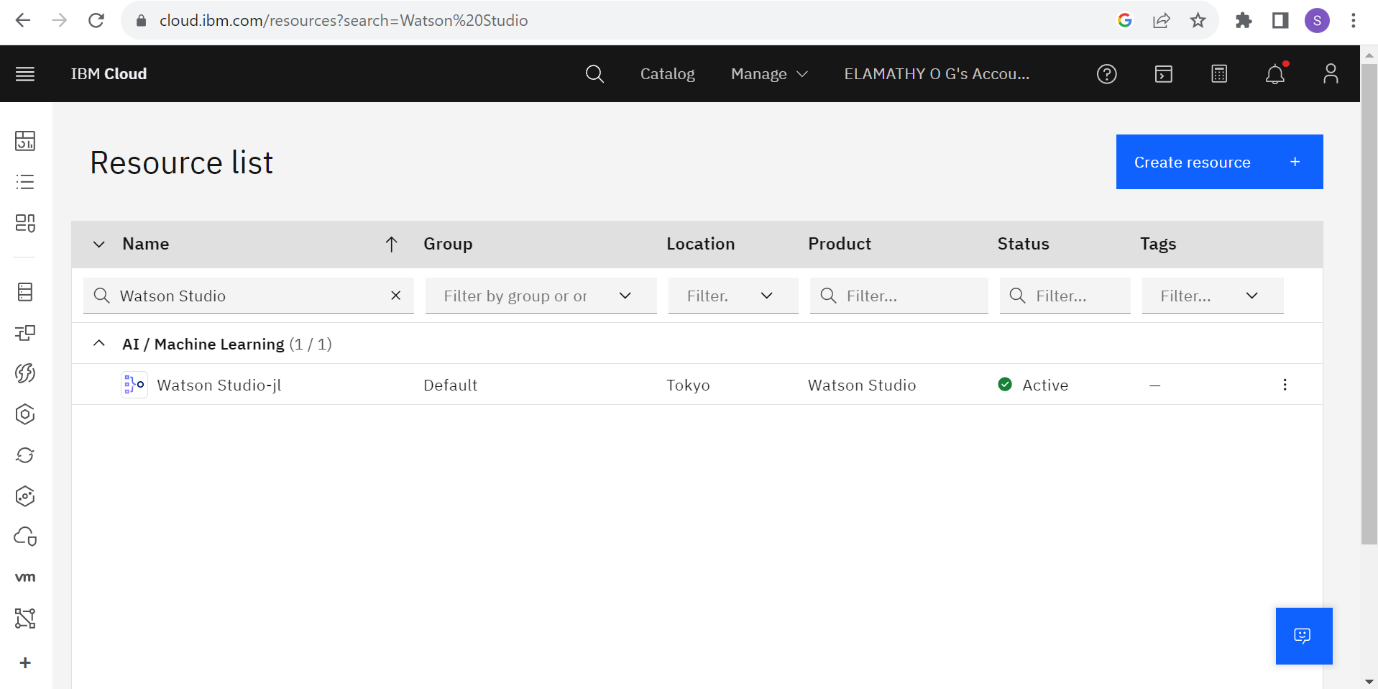




On the next page, you will get to name your service instance and choose your region. Click on the arrow to reveal the drop-down menu of regions. Make sure to select the region that is closest to you. Then scroll down and make sure that the **lite** plan is selected, and click the Create button.



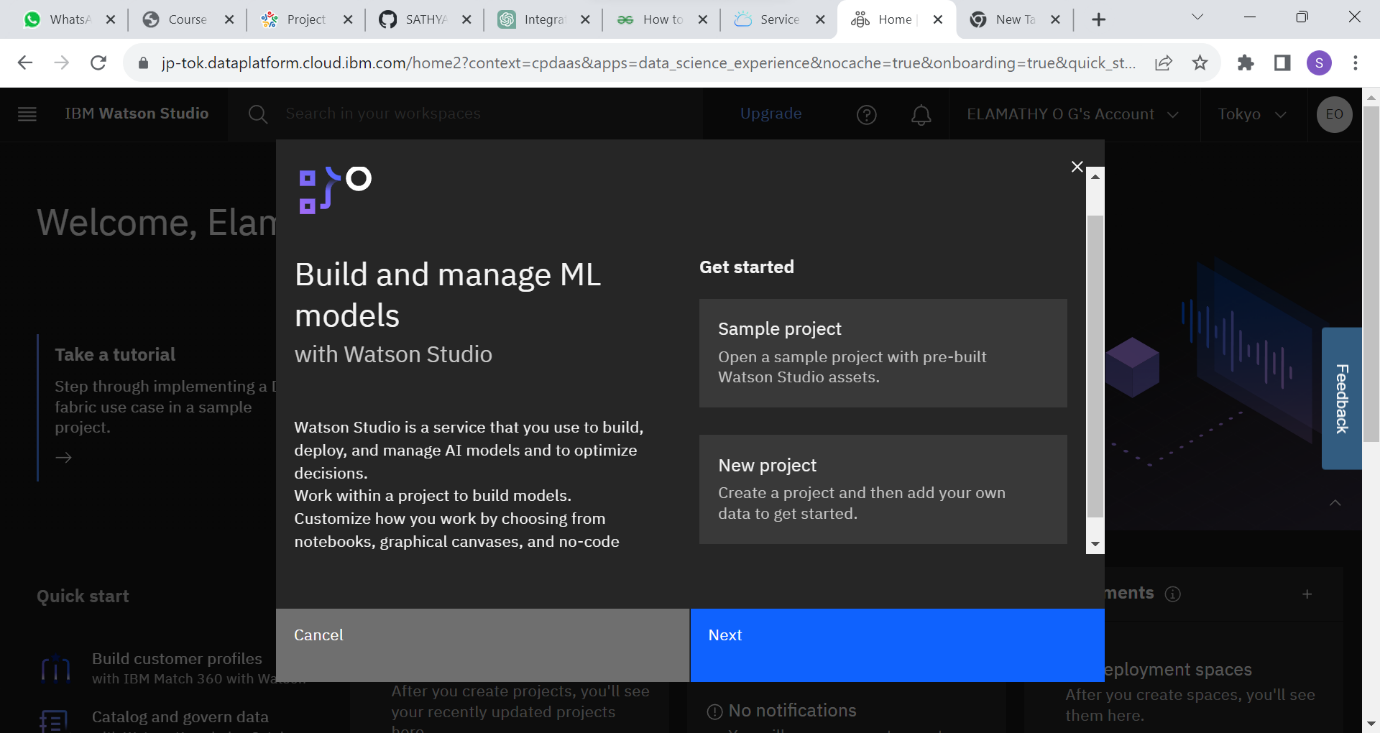






**Step :Create a Project**

Once you land on the IBM Watson Studio main page, start by **creating a project.** Click the New project.



ALGORITHM:

1. Import necessary libraries:
   * The script starts by importing the **requests** library for making HTTP requests and the **json** library for working with JSON data.
2. Define API Key and URL:
   * You need to replace **"YOUR\_API\_KEY"** and **"YOUR\_INSTANCE\_ID"** in the **api\_key** and **url** variables with your actual IBM Cloud Visual Recognition API key and instance ID. The **url** variable also includes a version parameter in the URL.
3. Define the NLG Model:
   * The script declares a function **generate\_caption(image\_url, confidence\_score)** that is expected to use an NLG model to generate a caption for the image. The specific NLG model used is not provided in the code and should be implemented separately.
4. Function to Classify the Image:
   * The **classify\_image(image\_url)** function is defined to classify an image using the IBM Cloud Visual Recognition API.
   * It constructs the necessary headers and parameters for the API request.
   * Sends a GET request to the API endpoint with the image URL and API key.
   * Parses the JSON response to extract the classified class and confidence score of the image.
   * If the classification is successful, it returns the class and confidence score; otherwise, it returns **None** and a confidence score of 0.
5. Main Function:
   * The **main(image\_url)** function is defined as the main entry point of the script.
   * It calls the **classify\_image(image\_url)** function to classify the provided image URL.
   * If the image is successfully classified, it then calls the **generate\_caption(image\_url, confidence\_score)** function to generate a caption for the image.
   * The classified class, confidence score, and generated caption are printed to the console.
   * If image classification fails (e.g., due to an invalid API key or URL), an error message is displayed.
6. Script Execution:
   * The **if \_\_name\_\_ == "\_\_main\_\_":** block ensures that the script is executed when run as the main program.
   * It defines the **image\_url** variable with the URL of the image you want to classify and generate a caption for.
   * The **main(image\_url)** function is called to initiate the image classification and caption generation process.

To make this code functional, you need to implement the **generate\_caption\_using\_nlg(image\_url)** function to generate captions using an NLG model. Additionally, you must replace the placeholder values with your actual IBM Cloud Visual Recognition API credentials.

**PROGRAM:**

import requests

import json

# IBM Cloud Visual Recognition API Key and URL

api\_key = "YOUR\_API\_KEY"

url = "https://api.us-south.visual-recognition.watson.cloud.ibm.com/instances/YOUR\_INSTANCE\_ID/v3/classify?version=YYYY-MM-DD"

# NLG model (You can use libraries like GPT-3, GPT-4, or train your own)

def generate\_caption(image\_url, confidence\_score):

# Call the NLG model to generate a caption

caption = generate\_caption\_using\_nlg(image\_url)

# Post-process the caption (e.g., remove special characters, format)

processed\_caption = post\_process\_caption(caption)

return processed\_caption

# Function to classify an image using IBM Cloud Visual Recognition

def classify\_image(image\_url):

headers = {

"apikey": api\_key

}

params = {

"url": image\_url

}

response = requests.get(url, headers=headers, params=params)

response\_json = response.json()

if 'images' in response\_json and len(response\_json['images']) > 0:

classified\_class = response\_json['images'][0]['classifiers'][0]['classes'][0]['class']

confidence = response\_json['images'][0]['classifiers'][0]['classes'][0]['score']

return classified\_class, confidence

else:

return None, 0

# Main function

def main(image\_url):

# Classify the image

classified\_class, confidence\_score = classify\_image(image\_url)

if classified\_class:

# Generate a caption for the image

caption = generate\_caption(image\_url, confidence\_score)

print(f"Classified as: {classified\_class}")

print(f"Confidence Score: {confidence\_score}")

print(f"Generated Caption: {caption}")

else:

print("Image classification failed.")

if \_\_name\_\_ == "\_\_main\_\_":

image\_url = "URL\_OF\_YOUR\_IMAGE"

main(image\_url)

SAMPLE OUTPUT:

Classified as: Class\_Label

Confidence Score: 0.XXXX

Generated Caption: This is a sample generated caption for the image.

Here's what each line in the output means:

* "Classified as": This line will display the class or label that IBM Cloud Visual Recognition assigned to the input image based on its classification. Replace "Class\_Label" with the actual class label detected by the service.
* "Confidence Score": This line will show the confidence score (a probability) associated with the detected class. Replace "0.XXXX" with the actual confidence score.
* "Generated Caption": This line will display the caption generated by your NLG model based on the image classification and the confidence score. Replace "This is a sample generated caption for the image." with the actual generated caption.

The actual output will depend on the image you provide, the performance of IBM Cloud Visual Recognition in classifying the image, and the capabilities of your NLG model in generating captions.

Top of Form

CONCLUSION:

IBM Watson Visual Recognition stands as a versatile and sophisticated tool that empowers businesses and developers to harness the capabilities of artificial intelligence for image recognition, description, and sentiment analysis.