**KGISL INSTITUTE OF TECHNOLOGY**

(AFFILIATAED TO ANNA UNIVERSITY)

Saravanampatti, Coimbatore – 641035



**Cloud computing**

*Submitted by*

*Abarna M* (711721106001)

Eswari k (711721106034)

Maghaa Miruthaa B (711721106060)

Hariharan R (711721106036)

Jaya Ganesh R (711721106048)

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

- If you don't already have an IBM Cloud account, sign up at [IBM Cloud](https://cloud.ibm.com/registration).

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

- Log in to your IBM Cloud account.

- From the IBM Cloud Dashboard, click "Create Resource."

- Search for "Visual Recognition" and select it.

- Follow the prompts to set up the service.

- Once provisioned, note your API keys or credentials. You will need them to access the service.

3. **Design a Simple Web Interface:**

- Create a folder for your web project and create an HTML file (e.g., index.html).

- Design a basic interface with an image upload feature. Here's a simple example:

html

<!DOCTYPE html>

<html>

<head>

<title>Image Recognition System</title>

</head>

<body>

<h1>Image Recognition System</h1>

<form id="imageUploadForm">

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

<button type="submit">Upload Image</button>

</form>

<div id="imageCaption">

<!-- AI-generated captions will be displayed here -->

</div>

</body>

</html>

4. **Integrate Visual Recognition Service:**

- In your web project, use JavaScript to handle image uploads and interact with IBM Visual Recognition.

- You'll need to make API requests to the Visual Recognition service using the API keys obtained earlier. You can use the IBM Watson Node.js SDK or make HTTP requests directly.

Here's an example using JavaScript (note that you'd need to replace 'YOUR\_API\_KEY' and 'YOUR\_MODEL\_ID' with your actual credentials):

javascript

const VisualRecognitionV3 = require('ibm-watson/visual-recognition/v3');

const { IamAuthenticator } = require('ibm-watson/auth');

const visualRecognition = new VisualRecognitionV3({

version: '2018-03-19',

authenticator: new IamAuthenticator({ apikey: 'YOUR\_API\_KEY' }),

});

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

const imageCaption = document.getElementById('imageCaption');

const imageUploadForm = document.getElementById('imageUploadForm');

imageUploadForm.addEventListener('submit', (e) => {

e.preventDefault();

const file = imageInput.files[0];

const params = {

imagesFile: file,

classifierIds: ['YOUR\_MODEL\_ID'], // Replace with your custom model ID

};

visualRecognition.classify(params)

.then(response => {

const caption = response.result.images[0].classifiers[0].classes[0].class;

imageCaption.textContent = `AI-generated caption: ${caption}`;

})

.catch(error => {

console.error(error);

imageCaption.textContent = 'AI caption generation failed.';

});

});

5. **Testing and Deployment:**

- Test your web interface by uploading various images to check the AI-generated captions.

- Once satisfied with the functionality, you can deploy your web application to a hosting platform or web server to make it accessible to users.

This guide provides the basic steps to create an image recognition system using IBM Cloud Visual Recognition. You can expand upon this foundation by improving the interface, adding more features, and integrating advanced functionalities based on your project's goals.