IMAGE RECOGNITION WITH IBM CLOUD VISUAL RECOGNITION

YOGAPRIYA N

REETHIKA R

SUMAIYABEGAM S

SURUTHI Y

SANTHOSH P

1. Create an IBM Cloud Account:

If you don't have an IBM Cloud account, go to the IBM Cloud website and sign up.

2. Set up the Visual Recognition:

Once you're logged into IBM Cloud, navigate to the IBM Cloud Catalog and search for "Visual Recognition."

Select the Visual Recognition service from the catalog.

Follow the instructions to create a new instance of the Visual Recognition service.

3. Obtain API Keys:

After creating the Visual Recognition service instance, go to the IBM Cloud Dashboard.

Navigate to your Visual Recognition service instance and click on it.

In the service details page, find the "Service credentials" tab.

Create new credentials if you don't have any, and note down the API key and other relevant information.

4. Set Up Your Project:

Create a new directory for your project on your local machine.

Inside the project directory, create a file to store your API keys securely (e.g., config.json). Store the Visual Recognition API key and any other relevant information.

5. Design a Simple Web Interface:

Create an HTML file (e.g., index.html) for your web interface. This file will include a form for users to upload images.

HTMLCopy code

```
<!DOCTYPE  html>

<html  lang="en">

<head>

    <meta charset="UTF-8">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Image Recognition System</title>

</head>

<body>

     <h1>Image Recognition System</h1>

     <form id="uploadForm" enctype="multipart/form-data">

         <input type="file" name="image" accept="image/*">

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

</form>

       <div id="result"></div>
       <script src="main.js"></script>

 </body>

 </html>
```

Output:

Image Recognition System

Upload Image

Create a JavaScript file (e.g., main.js) to handle the image upload and communicate with the Visual Recognition API.

javascriptCopy code

```
// main.js

 async function uploadImage() {
    const form = document.getElementById('uploadForm');
    const formData = new FormData(form);
    try {
        const response = await fetch('/recognize',{
            method: 'POST',
            body: formData,
        });
           const result = await response.json();
           displayResult(result);
      }
           catch (error) {
          console.error('Error uploading image:', error);
      }
    }
    function displayResult(result) {
     const resultDiv = document.getElementById('result');
    resultDiv.innerHTML = `<p><strong>AI-Generated Caption:</strong>
    ${result.caption}</p>`;
     }
```

Output:

```
node /tmp]/rlJ3DZEhyk.js
```

6. Set Up a Server:

Create a server file (e.g., server.js) using a server framework like Express (you'll need Node.js installed).

JAVASCRIPTCOPY CODE

```
// server.js


    const express = require('express');
     const multer = require('multer');
     const VisualRecognitionV3 = require('ibm-watson/visual-recognition/v3');
     const { IamAuthenticator } = require('ibm-watson/auth');
    const app = express();
    const port = 3000;
     const visualRecognition = new VisualRecognitionV3({
          version: '2018-03-19',
          authenticator: new IamAuthenticator({
              apikey: 'YOUR_API_KEY',
     }),
           url: 'https://api.us-south.visual-recognition.watson.cloud.ibm.com',
     });
    const storage = multer.memoryStorage();
     const upload = multer({ storage: storage });
    app.use(express.static('public'));
     app.post('/recognize', upload.single('image'), async (req, res) => {
         try {
              const classifyParams = {
                   imagesFile: req.file.buffer,
                    classifierIds: ['default'],
         };
        const result = await visualRecognition.classify(classifyParams);
        const caption = result.result.images[0].classifiers[0].classes[0].class;
    res.json({ caption });

} catch (error) {

    console.error('Error recognizing image:', error);
    res.status(500).json({ error: 'Internal server error' }); } });

app.listen(port, () => {

        console.log(`Server listening at http://localhost:${port}`);

});
```

Install the necessary Node.js packages by running:

BASHCOPY CODE

```
npm install express multer ibm-watson
```

7. Run Your Project:

Run your server:

BASHCOPY CODE

```
node server.js
```

Visit http://localhost:3000 in your web browser and test the image recognition system.

This is a basic setup, and you can enhance it further by adding more features, improving the user interface, and handling errors more gracefully. Additionally, consider securing your application if you plan to deploy it to production.

THANK YOU