IMAGE RECOGNITION WITH IBM CLOUD VISUAL RECOGNITION

AGENDA

Introduction

IBM Cloud

Design the Web Interface

Set up Backend Server

Deploy you Application

Output

Conclusion

INTRODUCTION

Image recognition, also known as image detection or computer vision, is a technology that enables machines to interpret and understand visual information from images or videos. It involves the use of algorithms and deep learning models to identify objects, patterns, text, or any other relevant information within an image or video frame. Image recognition systems analyze and process visual data, making it possible for computers to "see" and comprehend the content of images similar to the way humans do.

IBM CLOUD

Create an IBM Cloud Account:

Go to the IBM Cloud website (https://cloud.ibm.com/).
Sign up for a new account or log in if you already have one.

Create a Visual Recognition Service:

After logging in, navigate to the IBM Cloud Catalog.
Search for "Visual Recognition" and select the service.
Choose a Lite plan and create the service instance.
Once the service is created, note down the API key and endpoint URL. You'll need these for authentication.

DESIGN THE WEB INTERFACE

1.HTML and CSS:

Create an HTML file for the user interface.
Design a simple form allowing users to upload images.

Source file(index.html)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Image Recognition</title>
  <style>
    /* Add your CSS styles here */
  </style>
</head>

<body>
  <h1>Image Recognition</h1>
  <form id="upload-form">
    <input type="file" id="image-input" accept="image/*" required>
    <button type="submit">Upload</button>
  </form>
  <div id="result"></div>

  <script src="script.js"></script>
</body>

</html>
```

2.JavaScript

Create a javascript file to handle form submission and API cells.

Source file(script.js)

```
document.getElementById('uploadform').addEventListener('submit', function (event) {
    event.preventDefault();
    const formData = new FormData();
    const fileInput = document.getElementById('image-input');
    formData.append('images_file', fileInput.files[0]);
    console.log('Before fetch request');
    fetch('/api/upload', {
        method: 'POST',
        body: formData
    })
    .then(response => response.json())
    .then(data => {
        console.log('Fetch successful:', data);
        document.getElementById('result').innerText = `Caption: ${data.caption}`;
    })
    .catch(error => console.error('Error:', error));
});
console.log('After fetch request');
```

SET UP BACKEND SERVER

Install Dependencies:
Create a package.json file and install necessary dependencies using npm install express ibm-watson.

Create a Server (server.js):
Set up a Node.js server to handle image uploads and communicate with the Visual Recognition service.

Source file:

```
const express = require('express');
const app = express();
const multer = require('multer');
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: '1hu9Y61O_OY3Wg-bVAwIYEZ8bTb9Gd2pcwK2ZMwbVYxX' }),
  url: 'https://api.eu-gb.assistant.watsson.cloud.ibm.com/instances/3c3d735d-62b0-4e73-bf6b-3be5c41c6275',
});

const storage = multer.memoryStorage();
const upload = multer({ storage: storage });

app.post('/api/upload', upload.single('images_file'), (req, res) => {
  const params = {
    images_file: req.file.buffer,
  };

  visualRecognition.describe(params)
  .then(response => {
    const caption = response.result.images[0].classifiers[0].classes[0].class;
    res.json({ caption });
  })
  .catch(error => {
    console.error(error);
    res.status(500).json({ error: 'Internal Server Error' });
  });
});

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

DEPLOY YOUR APPLICATION

Deploy your backend (Node.js) application to a hosting service like IBM Cloud, or any other platform of your choice.

Upload your frontend (HTML, CSS, and JavaScript) files to the hosting service or use a static file hosting service.

Now, users can access your web interface, upload images, and view the AI-generated captions using the IBM Cloud Visual Recognition service.

OUTPUT

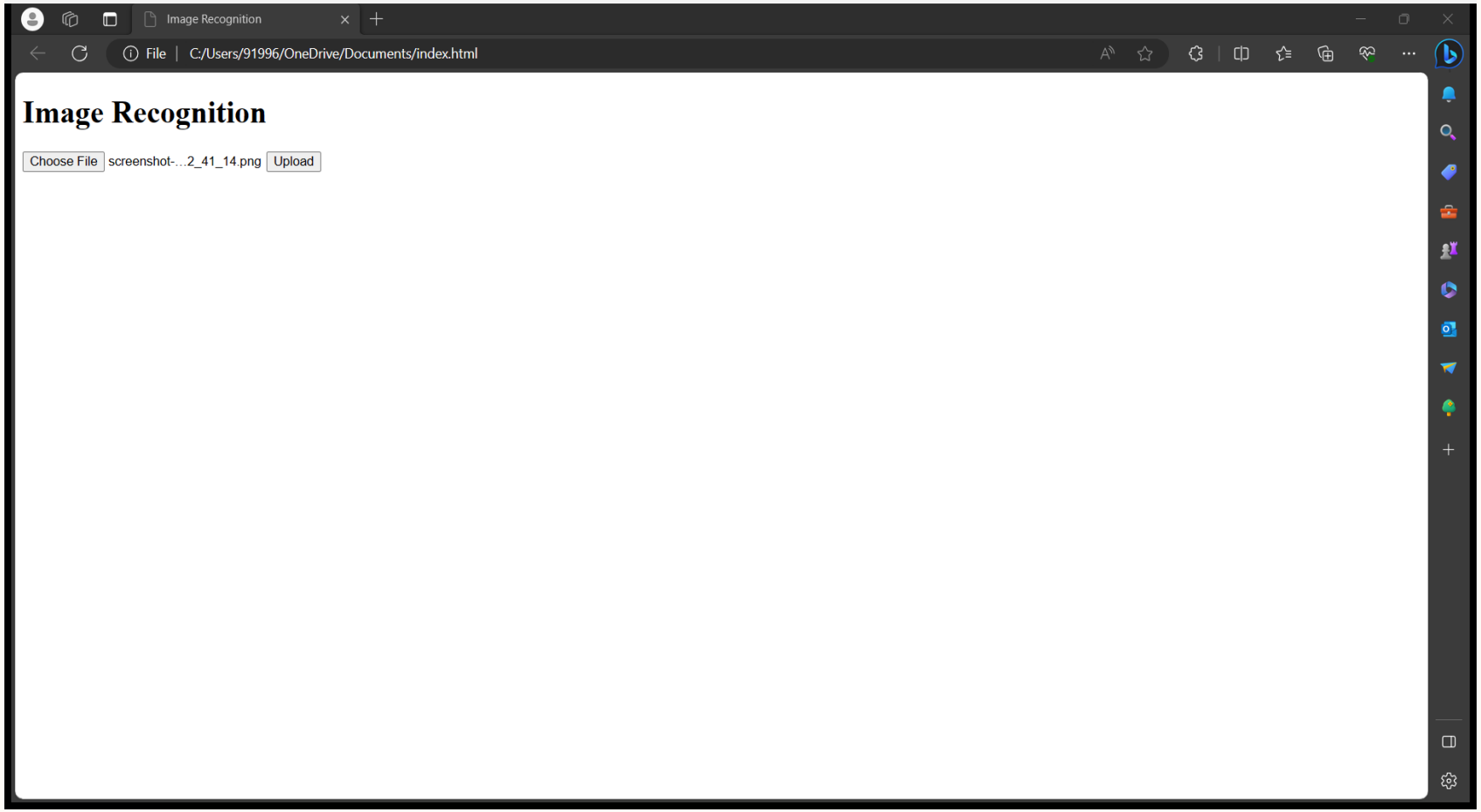
Image Recognition
File | C:/Users/91996/OneDrive/Documents/index.html
Image Recognition
Choose File screenshot-…2_41_14.png Upload

CONCLUSION

Thus the simple web interface is created to upload images and to view AI generated captions Successfully.

Now, users can access your web interface, upload images, and view the AI-generated captions using the IBM Cloud Visual Recognition service.