**PHASE 4: DEVELOPMENT OF MISSION LEARNING RECOGNITION WITH IBM CLOUD**

**Submitted By**

**Mahrswaran.J**

**920121106301**

**BE-ECE**

**Bharath Niketan Engineering College**

**PROCEDURE:**

1.Use Natural Language Generation:

* + Once you have the recognized classes or labels for the image, you can use a natural language generation library like NLTK, GPT-3, or another NLG tool to generate captions or descriptions based on the recognized classes.

Please note that for the NLG part, you would need to set up the NLG model of your choice and integrate it into your code. GPT-3, for example, would require access to the OpenAI GPT-3 API.

Remember to replace 'YOUR\_API\_KEY' and 'YOUR\_SERVICE\_URL' with your actual IBM Cloud Visual Recognition API key and service URL. This is just a basic outline, and the actual implementation may vary based on your specific use case and choice of NLG tool.

Top of Form

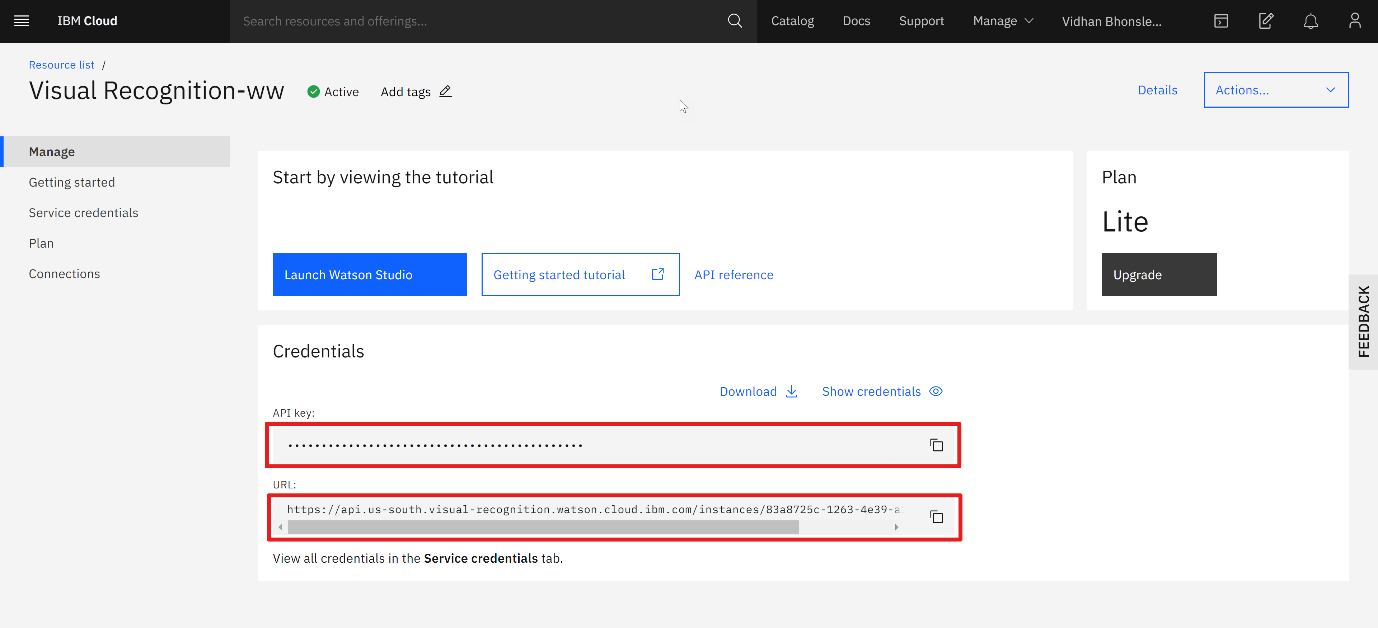
2. Obtain API Keys:

After creating the Visual Recognition service, you'll need to obtain API keys:

Click on your Visual Recognition service from the Resources page.

In the service details, navigate to the "Service Credentials" tab.

Create a new set of credentials or use the existing one, which will contain your API key and URL.



3. Design the Web Interface:

To create a simple web interface for users to upload images and view AI-generated captions, you can use HTML, CSS, and JavaScript.

**INTERFACE PROGRAM:**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>AI Image Captioning</title>

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div id="header">

        <h2 id="head1">Image Recognition with IBM Cloud Visual Recognition</h2>

        <h2 id ="head2">Naan Mudhalvaan Project</h2>

    </div>

    <br>

    <br>

<center>

    <div class="container">

        <h1>AI Image Captioning</h1>

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

        <button id="uploadButton">Upload Image</button>

        <img id="uploadedImage" alt="Uploaded Image">

        <div id="captionOutput">

            <h2>Generated Caption:</h2>

            <p id="captionText"></p>

        </div>

    </div>

</center>

    <script src="script.js"></script>

</body>

</html>

**style.css**

\*{

    margin: 0;

    padding: 0;

}

body {

    font-family: 'Poppins', sans-serif;

    text-align: center;

    background-image: url("6.jpg");

    background-size: cover;

    background-repeat:repeat-y;

}

#header{

    display: flex;

    padding: 30px;

    justify-content: space-between;

    background: rgb(2,0,36);

    background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(9,53,121,1) 34%, rgba(0,212,255,1) 100%);

}

#header h2{

    text-align: left;

    padding-left: 12px;

    color:white ;

}

#head2{

    color:white ;

    font-size: 15px;

    padding-top: 5px;

    padding-right:5px;

}

.container {

    max-width: 400px;

    margin:20px;

    padding: 20px;

    background: rgb(2,0,36);

    background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(9,53,121,1) 34%, rgba(0,212,255,1) 100%);

    border-radius: 8px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);

}

h1 {

    color: white;

}

h2{

    color: white;

}

#imageInput {

    display: none;

}

#uploadButton {

    background-color: #007BFF;

    color: #fff;

    padding: 10px 15px;

    border: none;

    cursor: pointer;

    margin-top: 10px;

    border-radius: 4px;

    transition: background-color 0.3s;

    transition: transform .2s;

}

#uploadButton:hover {

    background-color: #0056b3;

    transform: scale(1.05);

}

#uploadedImage {

    max-width: 100%;

    display: none;

    margin-top: 10px;

}

#captionOutput {

    margin-top: 20px;

}

#captionText {

    color: white;

    font-size: 18px;

}

**script.js**

document.getElementById("uploadButton").addEventListener("click", function () {

    document.getElementById("imageInput").click();

});

document.getElementById("imageInput").addEventListener("change", function () {

    const uploadedImage = document.getElementById("uploadedImage");

    const captionText = document.getElementById("captionText");

    const file = this.files[0];

    if (file) {

        const imageUrl = URL.createObjectURL(file);

        uploadedImage.src = imageUrl;

        uploadedImage.style.display = "block";

        captionText.textContent = "Generating caption...";

    }

});

const express = require("express");

const app = express();

const multer = require("multer");

const upload = multer({ dest: "uploads/" });

app.use(express.static("public"));

app.post("/upload", upload.single("image"), (req, res) => {

    // Process the uploaded image and generate captions using the AI model.

    // Send the generated caption as a response.

    const caption = "A beautiful image of " + req.file.originalname;

    res.json({ caption });

});

const port = 3000;

app.listen(port, () => {

    console.log(`Server is running on port ${port}`);

});

**OUTPUT:**



