**Name : DINESH S**

**Register No : 210621205014**

**Phase : 3**

**Project title: Image Recognition with IBM Cloud Visual Recognition**

**Developing and Building,**

Building a complete project for image recognition with IBM Cloud Visual Recognition involves multiple components, including setting up a backend server, creating a web-based front-end, and integrating with the Visual Recognition service. Below is a high-level outline of the steps involved,

Step 1: Setting Up the Backend Server

You can use a server-side technology of your choice, such as Node.js, Python, or Java, to handle file uploads, authenticate with IBM Cloud, and interact with the Visual Recognition service. Here's a basic example using Node.js with Express:

1. Install the required Node.js modules:

bash

npm install express ibm-watson multer

2. Create a server script (e.g., `server.js`) to handle file uploads and call the Visual Recognition service.

javascript

const express = require('express');

const app = express();

const multer = require('multer');

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

// Configure IBM Cloud Visual Recognition

const visualRecognition = new VisualRecognitionV3({

version: '2018-03-19',

authenticator: {

apiKey: 'YOUR\_API\_KEY',

},

url: 'https://api.us-south.visual-recognition.watson.cloud.ibm.com',

});

// Configure file upload using multer

const storage = multer.memoryStorage();

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

// Define a route for image classification

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

const params = {

imagesFile: req.file.buffer,

};

visualRecognition.classify(params)

.then(result => {

// Handle and return the classification result

res.json(result);

})

.catch(error => {

// Handle errors

res.status(500).json({ error: error.message });

});

});

// Start the server

const port = process.env.PORT || 3000;

app.listen(port, () => {

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

});

```

Step 2: Creating the Front-End

Design a web-based front-end where users can upload images and view the AI-generated captions. You can use HTML, CSS, and JavaScript for this purpose, as mentioned in the previous responses.

Step 3: Integrating the Front-End with the Backend

In the HTML form's JavaScript code, send a POST request to your backend server (`/classify` route) with the uploaded image. You'll need to handle the server response to display the AI-generated captions on your web page.

Step 4: Deploy the Application

You can deploy your backend server on a platform of your choice (e.g., Heroku, AWS, or IBM Cloud) and host your web front-end on a web hosting service. Make sure to keep your API keys and credentials secure.

Step 5: Testing and Enhancements

Test your application thoroughly with different images. You can also enhance the project by adding features such as user authentication, saving image classifications, and improving the user interface.

This is a high-level overview of building an image recognition project using IBM Cloud Visual Recognition. The specific implementation details may vary based on your chosen technologies and requirements.