## PHASE 5:

### SYNOPSIS:

1.Project`s objective.

2.Designing thinking.

3.Development phases.

4.Userinterface and engagement.

## 1.PROJECT`S OBJECTIVE:

- Image recognition is a computer vision task where an algorithm or model is trained to identify and classify objects or entities within an image. In this context, the goal is to leverage IBM's Visual Recognition service to develop a system capable of accurately and efficiently recognizing objects, scenes, and attributes within given images.

## 2.Design Thinking :

● Understand the end-users' needs and challenges related to image recognition. Consider their specific use cases and requirements.

● "Develop a reliable image recognition system using IBM Cloud Visual Recognition to accurately identify objects within images."

● Generate ideas for training data selection, model architecture, and integration methods. Encourage creativity in addressing the defined problem.

● Create a basic model using a small dataset to test feasibility. Explore different configurations and options provided by IBM Cloud Visual Recognition.

● Gather feedback from users and stakeholders on the prototype's accuracy and performance. Evaluate how well it identifies objects within images.

● Develop the full-scale image recognition system using the chosen approach. Integrate it with IBM Cloud Visual Recognition, ensuring seamless operation.

● Measure the system's recognition accuracy against predefined criteria. Collect user feedback to make any necessary adjustments.

● Based on evaluation results, refine the solution further. Continuously seek opportunities for improvement in accuracy, speed, and user experience.

## 3.DEVELOPMENT PHASES:

### PHASE 1:

1. **Sign Up for IBM Cloud Account:**
    - If you don't already have an IBM Cloud account, you'll need to create one. Go to the IBM Cloud website (https://cloud.ibm.com/registration) and sign up for a free or paid account.
2. **Log In to IBM Cloud:**
    - Log in to your IBM Cloud account using your registered credentials .
3. **Create or Access a Watson Visual Recognition Service:**
    - Once you're logged in, navigate to the IBM Cloud Dashboard.
    - You can either create a new service or access an existing one.
    - If you're creating new one, search for "Visual Recognition" and create the service.
4. **Get Your API Key:**

    After you've created or accessed your Visual Recognition service, you should be able to find your API key in the service's dashboard.

    - It might be under "Service Credentials" or something similar.
    - Click on "View credentials" to see your API key(s).
5. **Save Your API Key:**
    - It's essential to keep your API key secure. Save it in a safe place.
    - If you're using the API key in code, consider using environment variables to store it securely.
6. **Using the API Key:**
    - You can now use the API key to authenticate your requests to IBM Watson Visual Recognition. When you make requests to the Watson Visual Recognition API, you need to include the API key as an authentication token in the headers of your HTTP requests.

**Imgr.html:**

```
<!DOCTYPE html>

<html>

<head>

            <style>

    /* style.css */


/* Global Styles */

body {

   font-family: Arial, sans-serif;

   background-color: #567795;

}


h1 {

   color: #333;

}
```

```
/* Image Upload Styles */
#imageInput {
    margin: 10px 0;
}

/* Button Styles */
button {
    background-color: #007BFF;
    color: #fff;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
}

button:hover {
    background-color: #0056b3;
}
    input{

      background-color: orange;
      }

/* Results Styles */
#results {
    margin: 20px 0;
    font-weight: bold;
}

  </style>
    <title>Image Recognition</title>
</head>
<body>
    <h1 >Image Recognition</h1>
    <input type="file" id="imageInput" accept="image/*">
    <button onclick="recognizeImage()">Recognize</button>
    <div id="results"></div>

    <script>
        function recognizeImage() {
            const fileInput = document.getElementById('imageInput');
            const resultsDiv = document.getElementById('results');

            if (fileInput.files.length === 0) {
                resultsDiv.innerHTML = "Please select an image.";
                return;
            }

            const imageFile = fileInput.files[0];
            const formData = new FormData();
            formData.append('image', imageFile);

            fetch('/recognize', {
                method: 'POST',
                body: formData
            })
            .then(response => response.json())
            .then(data => {
                resultsDiv.innerHTML = `Recognition Result: ${data.result}`;
            })
            .catch(error => {
                resultsDiv.innerHTML = 'Error: ' + error;
            });
        }
    </script>
</body>
</html>
```

**Server.js:**

```
                                const express = require('express');
```

```
const bodyParser = require('body-parser');
const VisualRecognitionV3 = require('ibm-watson/visual-recognition/v3');
const { IamAuthenticator } = require('ibm-watson/auth');

const app = express();
const port = process.env.PORT || 3000;

app.use(bodyParser.json());
app.use(express.static(__dirname));

const apiKey = 'YOUR_API_KEY'; // Replace with your Watson Visual Recognition API
key
const apiUrl = 'YOUR_API_URL'; // Replace with your Watson Visual Recognition API
URL

const visualRecognition = new VisualRecognitionV3({
  version: '2018-03-19',
  authenticator: new IamAuthenticator({
    apikey: apiKey,
  }),
  url: apiUrl,
});

app.post('/recognize', (req, res) => {
  const imageFile = req.files.image; // Assuming you're using Express.js and the `express-
fileupload` middleware
  const params = {
    images_file: imageFile.data,
  };

  visualRecognition.classify(params)
    .then(response => {
      const classes = response.result.images[0].classifiers[0].classes;
      const topClass = classes[0].class;
      res.json({ result: topClass });
    })
    .catch(error => {
      res.status(500).json({ error: error.message });
    });
});

app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

## PHASE 2:

1. **Sign up for IBM Watson Services:**
   - sign up for IBM Watson services and create an instance of Watson Visual Recognition on the IBM Cloud platform. Create an project by clicking “create” on the bottom.
2. **Collect and Prepare Your Data:**
   - Gather a dataset of images that you want to recognize and classify. Ensure that the dataset is well-organized and labeled.you're training a custom model, you should have images grouped into categories or classes

- • Select minimum of 10 images to upload and test.

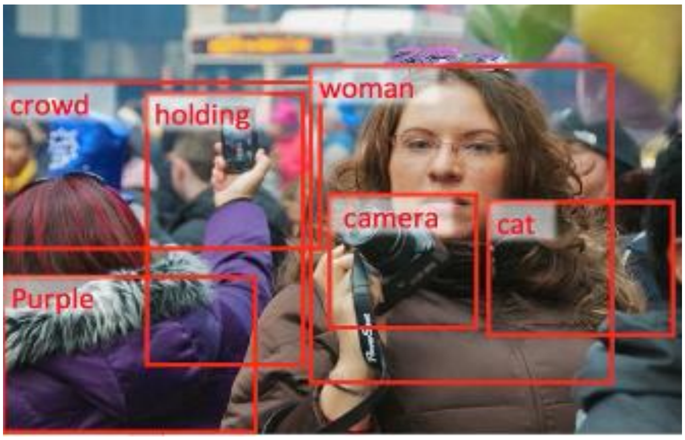
3. **Create a Custom Model (Optional):**
   - If you need to recognize specific objects or categories not covered by the pre-trained models, you can train your own custom model. This is useful for cases like recognizing your company's products or unique objects. - Go to your Watson Visual Recognition instance on the IBM Cloud. - Create a new project and upload your labeled dataset. - Train the custom model using the dataset**.**
4. **Use Pre-Trained Models (Optional):**
   - If your recognition needs align with the pre-trained models provided by Watson Visual Recognition, you can skip custom model training and use the pre-built models.click "add to project**".**
5. **Obtain API Credentials:**
   - You will need API credentials (API Key and URL) to access the Watson Visual Recognition service. You can find this information in the IBM Cloud dashboard for your Watson Visual Recognition instance.
6. **Interpret the Results and Application:**
   - The response from Watson Visual Recognition will contain information about the recognized objects or categories in the image. Depending on your use case, you can use this information to take specific actions or provide user feedback.
   - Integrate the image recognition code into your application or service, ensuring it can process images and return meaningful results**.**
7. **Test and Iterate:**
   - Test the image recognition system with various images and refine your model or code as needed. Iterate to improve accuracy and performance.
8. **Deploy:**
   - Finally, deploy your application or service with the integrated image recognition functionality to make it available to users. These steps should help you get started with implementing image recognition using Watson Visual Recognition. Customization and fine-tuning may be necessary to meet your specific requirements.

## 4.USER INTERFACE :

The user interface is designed with a clean and intuitive layout. Users can upload photos and view AI-generated captions. It features user profiles and social sharing capabilities.

## AI GENERATED CAPTION : USER ENGAGEMENT:

- AI-generated captions enhance user engagement by providing context and storytelling elements to photos.



- **Personalization:** AI tailors captions to the content of each user's photos, making the experience more personal and engaging.
- **Efficiency:** Users save time on caption creation, increasing their willingness to share photos and engage with the app.
- **Storytelling:** Captions turn individual images into narrative stories, making the content more compelling and relatable.
- **Accessibility:** AI-generated captions also benefit visually impaired users by describing the content of images.

**USER ENGAGEMENT:**

- enhanced engagement leads to increased user satisfaction, longer app usage, and potentially more sharing, driving the success of the application.