**Image classification:**

Implement the image classification process using the IBM Cloud Visual Recognition API.

**AI-Generated Captions:**

Using natural language generation to create captions for the recognized images.

**Program:**

const form = document.querySelector('form');

const caption = document.querySelector('#caption');

form.addEventListener('submit', async (e) => {

e.preventDefault();

const formData = new FormData(form);

const imageFile = formData.get('image');

const response = await fetch('https://api.example.com/visual-recognition', {

method: 'POST',

body: formData

});

const data = await response.json();

caption.textContent = data.caption;

});

**USER INTERFACE:**

To design a simple web interface where users can upload images and view the AI-generated captions.

<!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>

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

</head>

<body>

<header>

<h1>Image Recognition System</h1>

</header>

<section>

<form enctype="multipart/form-data" action="upload" method="post">

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

<button type="submit">Upload</button>

</form>

</section>

<section id="result">

<h2>AI-Generated Caption:</h2>

<p id="caption">Waiting for image recognition...</p>

</section>

<footer>

<p>&copy; 2023 Image Recognition System. All rights reserved.</p>

</footer>

</body>

</html>

CSS: style.css

@import url(https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css);

@import url('https://fonts.googleapis.com/css?family=Roboto');

// Vars and Reset

//

$theme: #454cad;

$dark-text: #5f6982;

html,body,\* { box-sizing: border-box; font-size: 16px;}

html,body { height: 100%; text-align: center;}

body { padding: 2rem; background: #f8f8f8;}

h2 {

font-family: "Roboto", sans-serif;

font-size: 26px;

line-height: 1;

color: $theme;

margin-bottom: 0;

}

p {

font-family: "Roboto", sans-serif;

font-size: 18px;

color: $dark-text;

}

// Upload Demo

//

.uploader {

display: block;

clear: both;

margin: 0 auto;

width: 100%;

max-width: 600px;

label {

float: left;

clear: both;

width: 100%;

padding: 2rem 1.5rem;

text-align: center;

background: #fff;

border-radius: 7px;

border: 3px solid #eee;

transition: all .2s ease;

user-select: none;

&:hover {

border-color: $theme;

}

&.hover {

border: 3px solid $theme;

box-shadow: inset 0 0 0 6px #eee;

#start {

i.fa {

transform: scale(0.8);

opacity: 0.3;

}

}

}

}

#start {

float: left;

clear: both;

width: 100%;

&.hidden {

display: none;

}

i.fa {

font-size: 50px;

margin-bottom: 1rem;

transition: all .2s ease-in-out;

}

}

#response {

float: left;

clear: both;

width: 100%;

&.hidden {

display: none;

}

#messages {

margin-bottom: .5rem;

}

}

#file-image {

display: inline;

margin: 0 auto .5rem auto;

width: auto;

height: auto;

max-width: 180px;

&.hidden {

display: none;

}

}

#notimage {

display: block;

float: left;

clear: both;

width: 100%;

&.hidden {

display: none;

}

}

progress,

.progress {

// appearance: none;

display: inline;

clear: both;

margin: 0 auto;

width: 100%;

max-width: 180px;

height: 8px;

border: 0;

border-radius: 4px;

background-color: #eee;

overflow: hidden;

}

.progress[value]::-webkit-progress-bar {

border-radius: 4px;

background-color: #eee;

}

.progress[value]::-webkit-progress-value {

background: linear-gradient(to right, darken($theme,8%) 0%, $theme 50%);

border-radius: 4px;

}

.progress[value]::-moz-progress-bar {

background: linear-gradient(to right, darken($theme,8%) 0%, $theme 50%);

border-radius: 4px;

}

input[type="file"] {

display: none;

}

div {

margin: 0 0 .5rem 0;

color: $dark-text;

}

.btn {

display: inline-block;

margin: .5rem .5rem 1rem .5rem;

clear: both;

font-family: inherit;

font-weight: 700;

font-size: 14px;

text-decoration: none;

text-transform: initial;

border: none;

border-radius: .2rem;

outline: none;

padding: 0 1rem;

height: 36px;

line-height: 36px;

color: #fff;

transition: all 0.2s ease-in-out;

box-sizing: border-box;

background: $theme;

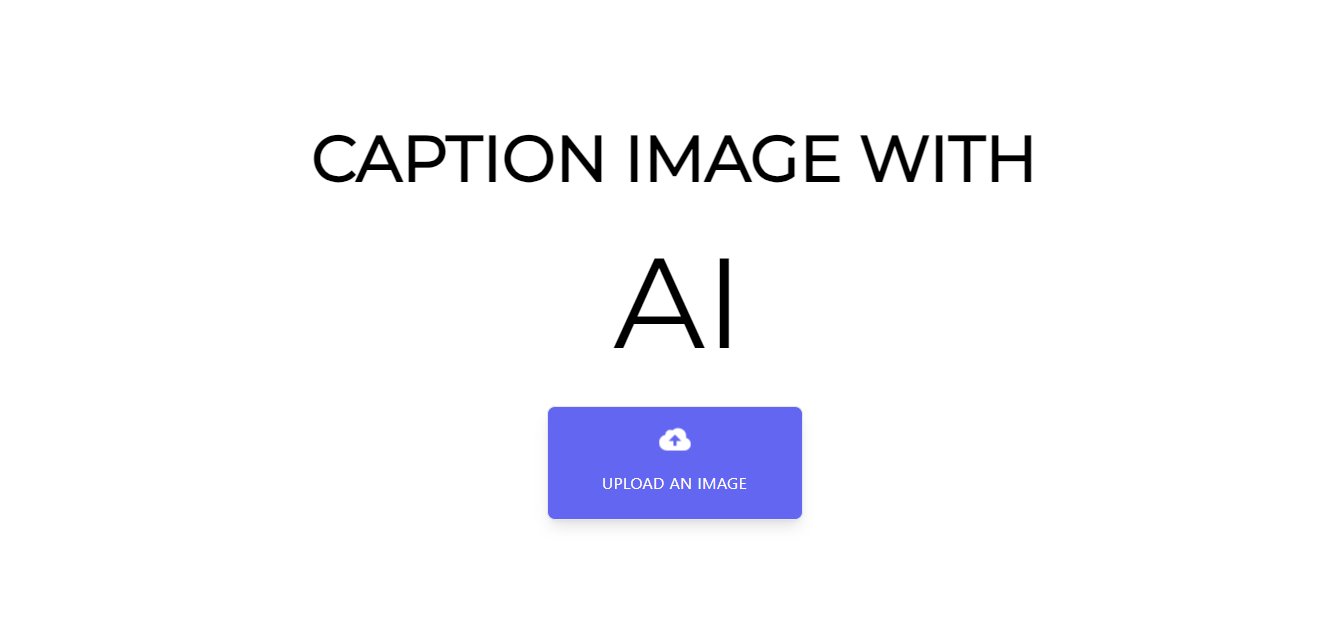
border-color: $theme;

cursor: pointer;

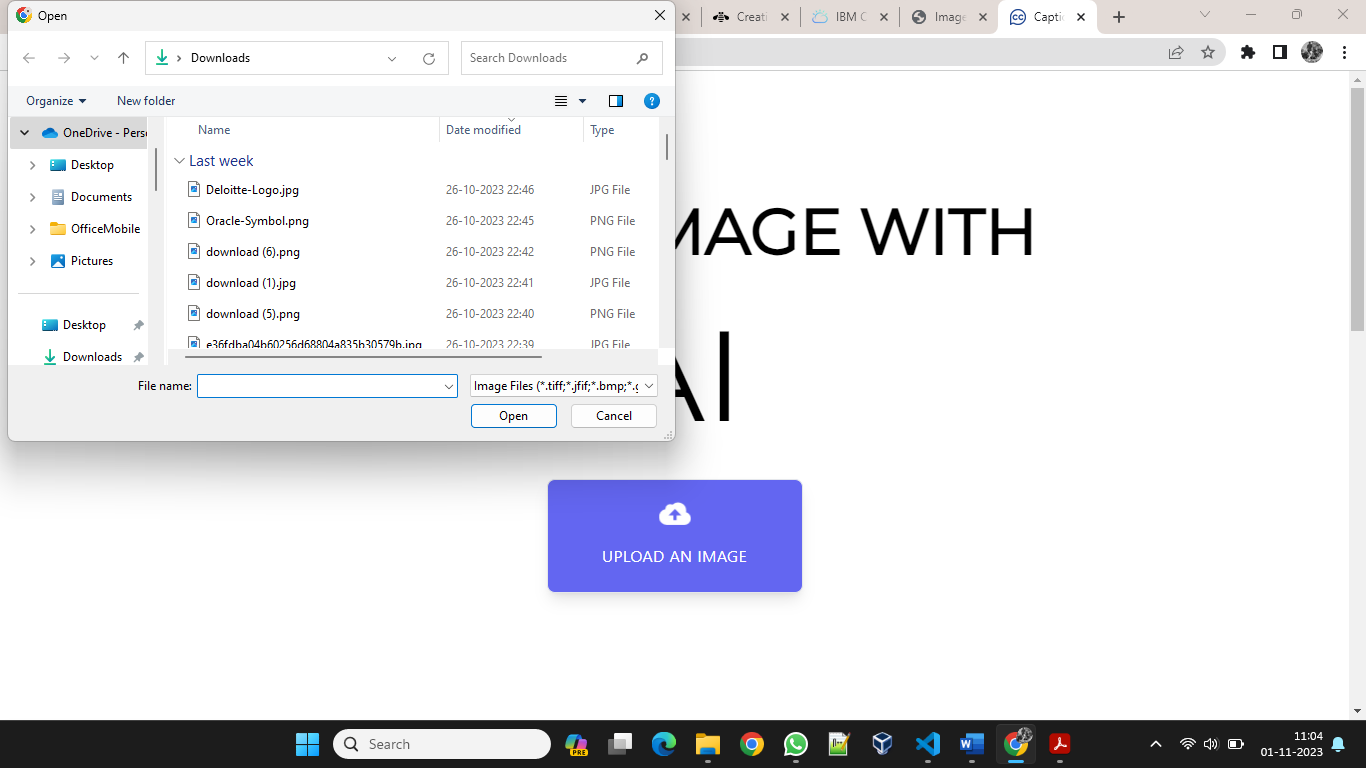
}

}

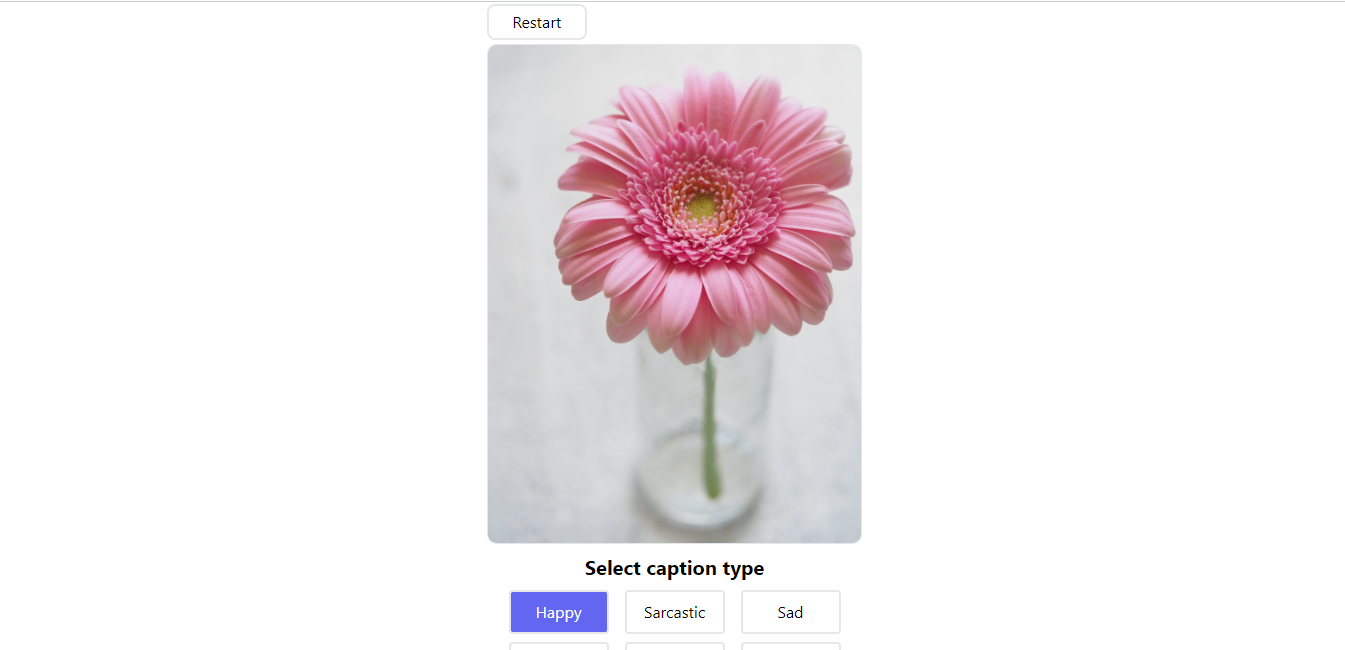
Some sample outputs are given:



Upload the images here..



Select the type of caption from the given options.



Multiple captions are generated for the given image and the specified type of the caption.

