# Image Compressor Web Application

**This is a simple and interactive web-based image compressor. Users can upload images, and the application will display a preview of the original image alongside compressed versions with different levels of compression.**

**Features**

* **Drag and Drop Upload: Users can drag and drop an image file directly into the upload box or click to browse.**
* **Real-time Preview: Displays the uploaded image and its compressed versions.**
* **Multiple Compression Levels: Automatically generates compressed images at varying levels (75%, 50%, and 25%).**
* **Download Option: Users can download the compressed images.**
* **Error Handling: Provides error messages for unsupported file types or files below the minimum size.**

**Technologies Used**

* **HTML5: Structure of the web page.**
* **CSS3: Styling with a cheerful and vibrant theme.**
* **JavaScript: Handles file input, image processing, and UI interactions.**

**How the Algorithm Works**

1. **Image Loading:**
   * **When the user selects or drops an image, it is read using the FileReader API.**
   * **The original image is displayed immediately for comparison.**
2. **Canvas Drawing:**
   * **The selected image is drawn onto an HTML5 <canvas> element.**
   * **This allows direct pixel manipulation and resizing of the image.**
3. **Binary Search for Compression Quality:**
   * **The algorithm aims to find the optimal compression quality for each target size (75%, 50%, and 25%).**
   * **A binary search approach is used:**
     + **It starts with an initial quality of 0.5.**
     + **The image is compressed using canvas.toBlob() with the current quality.**
     + **If the resulting file size is larger than the target size, the quality is decreased; otherwise, it is increased.**
     + **This process repeats until the file size is close to the target or a maximum number of attempts is reached.**
4. **Result Generation:**
   * **Once the optimal quality is determined, the compressed image is displayed in a card with details such as dimensions, file size, and percentage savings.**
   * **A download button is provided for the user to save the compressed image.**

**How to Use**

1. **Open the web application in a browser.**
2. **Drag and drop an image into the upload area or click the box to select an image from your device.**
3. **Wait for the image to be processed.**
4. **Preview the original image and its compressed versions.**
5. **Click the "Download" button to save the compressed image to your device.**

**File Structure**

* **index.html: Contains the main HTML structure and content.**
* **style.css: Defines the styles for the web page.**
* **script.js: Contains JavaScript code for handling file uploads, processing images, and updating the UI.**

**Customization**

**Changing the Color Theme**

**You can modify the color scheme by editing the CSS variables defined in the :root selector:**

**:root {**

**--primary: #4caf50;**

**--primary-dark: #388e3c;**

**--background: #f0f4c3;**

**--card-bg: #ffffff;**

**--text: #333333;**

**--text-secondary: #616161;**

**--border: #c5e1a5;**

**--shadow: rgba(0, 0, 0, 0.1);**

**}**

**Adjusting the Minimum File Size**

**The minimum file size for uploads can be changed by modifying the MIN\_FILE\_SIZE constant in script.js:**

**const MIN\_FILE\_SIZE = 2 \* 1024 \* 1024; // 2MB**

**Browser Compatibility**

**This web application is compatible with modern web browsers, including:**

* **Google Chrome**
* **Mozilla Firefox**
* **Microsoft Edge**
* **Safari**

Name : Hager Tarek Anter

ID: 20230331943