Web Application Development Assignment

Subject: Web Application Development

Name: Jayesh Gorakh Kande

Roll no.: 29

Q.1) Illustrate the need for GraphicsMagick package

What is GraphicsMagick?

GraphicsMagick is a powerful, lightweight image processing tool used for editing, converting, resizing, and manipulating images in a fast and efficient way.

★ Why Do We Need GraphicsMagick?

1. Efficient Image Processing

GraphicsMagick is designed for **high performance**. It can handle large batches of images quickly, which is essential for applications like:

- Web servers that process user-uploaded images
- Batch resizing of image assets for apps or games
- Command-line automation of image edits

2. Lightweight Alternative to ImageMagick

It is a **fork of ImageMagick**, but is often preferred because:

- It's faster for many operations
- More stable and consistent results

• Requires less memory

3. Supports Multiple Formats

It supports over **88 major image formats** including PNG, JPEG, TIFF, BMP, GIF, and more — making it highly versatile for developers.

4. Scripting and Automation

GraphicsMagick works well in **shell scripts** or integrated into backend logic using Node.js, Python, PHP, etc. For example:

bash

CopyEdit

gm convert input.jpg -resize 300x300 output.jpg

5. Used in Real-Time Applications

It's widely used in real-world tools:

- Image processing in CMS platforms
- Automation in DevOps pipelines
- Image handling in e-commerce applications (thumbnails, compression)

Q.2) Implement small application which will make use of mentioned package (Input/Output)

Great, Jayesh! Let's implement a **small Node.js application** using the gm package (GraphicsMagick wrapper) to **resize an image**. The app will take an input image and output a resized version.

✗ Step-by-Step: GraphicsMagick Image Resizer (Node.js)

1. Install Dependencies

First, make sure you have **GraphicsMagick installed** on your system.

For Ubuntu:

sudo apt-get install graphicsmagick

For Windows/Mac, download from:

https://www.graphicsmagick.org/download.html

Then, initialize a Node.js project and install the gm package:

npm init -y
npm install gm

2. Project Structure

3. Code: app.js

```
const gm = require('gm').subClass({ imageMagick: false }); // false for
GraphicsMagick
const inputPath = 'input.jpg';
const outputPath = 'output_resized.jpg';
gm(inputPath)
 .resize(300, 300) // Resize to 300x300
 .write(outputPath, function (err) {
  if (err) {
   console.log('X Error resizing image:', err);
  } else {
   console.log(' ✓ Image resized successfully! Check output_resized.jpg');
 });
```

4. Run the App

Place any image in the project folder named input.jpg, then run:

node app.js

You'll see:

Image resized successfully! Check output_resized.jpg

Input/Output

Input (input.jpg) → Output (output_resized.jpg)

1024x768 resolution 300x300 resized image

Q.3) Illustrate the need for code of ethics

Example 2 Code of Ethics for Image Processing Application using GraphicsMagick

1. Respect User Privacy

- Do **not use or process user-uploaded images** without explicit consent.
- Avoid storing unnecessary personal or sensitive image data.
- Implement proper file deletion after processing if not required.

🛂 2. Avoid Misuse of Images

- Do not manipulate images for **misleading**, **harmful**, **or deceptive** purposes (e.g., deepfakes or misinformation).
- Ensure image transformations do not violate copyrights or digital rights.

3. Ensure Fair Use

• Only use the application for **authorized and lawful** purposes.

 Do not automate or mass-process copyrighted images without permission.

4. Secure the Application

- Implement secure file upload mechanisms to prevent malicious file uploads.
- Prevent buffer overflow or directory traversal attacks when processing files.

5. Promote Accessibility and Transparency

- Inform users clearly about:
 - What will happen to their images
 - What data is stored and for how long
- Provide an option to delete data if needed.

6. Be Accountable

- Monitor application logs and handle errors or bugs responsibly.
- If any ethical violation or misuse is reported, take immediate corrective action.

>> 7. Respect Open Source Licenses

• GraphicsMagick is open source. Always **acknowledge and respect its license terms** when using or distributing your application.

Example Ethical Notice (You can add this to your app's UI or README):

This application processes images using GraphicsMagick. User images are processed securely and are not stored permanently. We respect your privacy and expect fair and responsible use of this tool.