

Got it 

Here's a **simple report text** you can copy-paste directly into Word and format as needed.

Project Report

Name: Paluvari Karthik

Roll No: 23675A7305

Subject: Computer Vision

Course: AIML-A

Title

Image Processing Toolkit – GUI Based Application using Streamlit and OpenCV

Objective

The main objective of this project is to design and implement a GUI-based application in Python that demonstrates fundamental image processing operations using **OpenCV**.

The application provides an interactive interface where users can upload images, apply different transformations, and view results side by side in real-time.

GUI Layout

1. Menu Bar

- Open → Upload an image
- Save → Save processed image
- Exit → Close app

2. Sidebar (Operations)

- Image Info (Resolution, Format, Channels, DPI)
- Color Conversions (RGB, HSV, YCbCr, Grayscale)
- Transformations (Rotation, Scaling, Translation, Affine, Perspective)
- Filtering & Morphology (Gaussian, Median, Sobel, Laplacian, Erosion, Dilation)
- Enhancement (Histogram Equalization, Contrast Stretching, Sharpening)
- Edge Detection (Sobel, Canny, Laplacian)
- Compression (JPG, PNG, BMP)

3. Main Display Area

- Left: Original Image

- Right: Processed Image

4. Status Bar

- Dimensions, File format, File size, DPI
-

❖ Implementation Phases

- **Phase 1** – Setup base Streamlit app, sidebar, and upload widget.
 - **Phase 2** – Implement image fundamentals and color conversions.
 - **Phase 3** – Add transformations (rotate, scale, translate).
 - **Phase 4** – Add filtering & morphology operations.
 - **Phase 5** – Add enhancement and edge detection methods.
 - **Phase 6** – Add compression & file handling.
 - **Phase 7** – Polish GUI with vibrant colors, download button, and comparison view.
-

📁 Deliverables

- **Codebase:** app.py (Streamlit GUI app)
 - **Notebook:** ImageToolkit_<roll-no>.ipynb (theory + practice tasks)
 - **Report:** Documentation with screenshots and explanations
 - **Final Demo:** Interactive Streamlit GUI
-

🚀 Bonus Features

- Sliders for kernel size, rotation angle, scaling factor, Canny thresholds
 - Split-screen comparison (original vs processed)
 - Real-time video mode with webcam
 - Download button for processed images
-

✅ Conclusion

This project successfully demonstrates how **fundamental image processing techniques** can be integrated into a user-friendly GUI using **Python, Streamlit, and OpenCV**.

It provides both theoretical learning and practical implementation for Computer Vision applications.

Do you want me to also **add diagrams/screenshots placeholders** (like “Insert Screenshot of GUI here”) so it looks like a proper Word assignment?





