

**Hackathon Submission Template (Level-1-Solution)**

**Use Case Title:** Color Detection from Images

**Student Name:** S. Nasreen Farzana

**Register Number:** 511723104007

**Institution:** Pallavan College of Engineering

**Department:** B.E. (CSE)

**Date of Submission:** 17-05-2025

# 1.Problem Statement

Designers, developers, and digital artists often need to identify and reuse specific colors from images. However, manually detecting these colors and matching them with known names and RGB values is inefficient. A utility tool that allows users to upload an image and click on it to detect the color can save time and enhance productivity.

# 2.Proposed Solution

We propose a simple, interactive web-based application using Python and Streamlit that enables users to upload an image, click anywhere on it, and instantly view the name and RGB values of the selected pixel color. The application uses OpenCV for image processing and a CSV dataset to match the pixel's RGB value to the closest known color name. The interface displays both the name and RGB values and shows a color box for easy reference.

# 3.Technologies & Tools Considered

* Python
* OpenCV (for image reading and pixel detection)
* Pandas (for dataset handling)
* Streamlit (for web-based interface)
* colors.csv (dataset with color names and RGB values)
* GitHub (for version control and documentation)

# 4.Architecture & Workflow

**Workflow:**

1. User uploads an image using Streamlit UI.
2. The image is displayed on the web page.
3. User clicks on a point in the image.
4. OpenCV captures the coordinates and fetches the RGB value.
5. The RGB value is matched against a color dataset to find the closest color name.
6. The app displays the color name, RGB value, and a color box in real time.

Architecture Diagram:



User Uploads



Display Color Name +



Match with Dataset



Fetch RGB Value at



OpenCV Captures Pixel



User Clicks on



Image Displayed in App



# 5.Feasibility & Challenges

* **Feasibility:**
* Highly practical and easy to implement using standard Python libraries.
* Streamlit simplifies web deployment.
* Suitable for beginners and intermediate developers.

**Challenges:**

* Matching exact colors can be tricky due to lighting and shading differences.
* Ensuring responsive UI for various image sizes.
* Handling invalid or corrupted image files gracefully.

# 6.Expected Outcome & Impact

The application will:

* Allow quick and accurate color detection from images.
* Improve workflow for designers and developers.
* Be accessible via a simple web interface.
* Be open source, enabling others to enhance and reuse the tool. Beneficiaries:
* Designers, web developers, artists, accessibility engineers, and students.

# 7.Future Enhancements

* Add HEX color value support.
* Enable zooming in for precision pixel selection.
* Allow saving selected colors to a palette.
* Include support for image drag-and-drop.
* Add accessibility features like high-contrast view.