



DERMATOLOGY MADE EASY


SHURIDA BINTE RASHED - 21047377

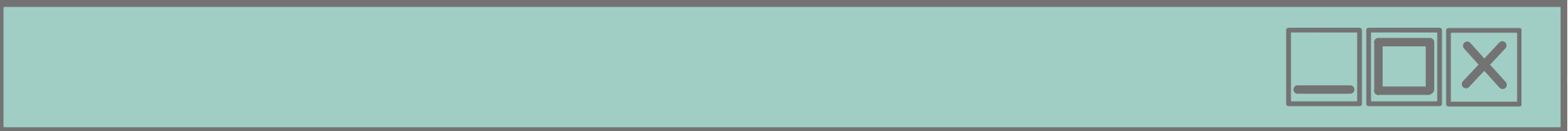


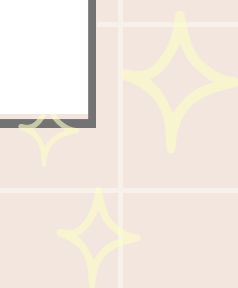
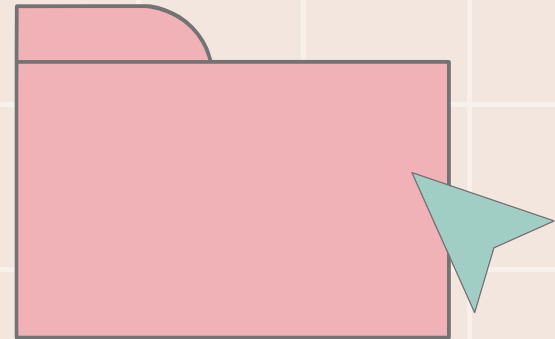
A LITTLE BACK-STORY



TABLE OF CONTENTS_



- 
- _01 INTRODUCTION
 - _02 OBJECTIVES & REQUIREMENTS
 - _03 PLANNING & DESIGN
 - _04 DEVELOPMENT BREAKDOWN
 - _05 TESTING & OUTPUT
 - _06 REFLECTION & CONCLUSION



INTRODUCTION_

Problem & Motivation:

- 📌 Skin cancer affects millions; early detection saves lives
- 📌 Manual diagnosis is time-consuming and expertise-bound

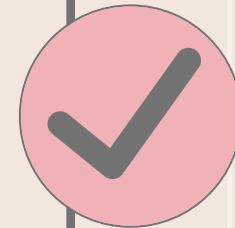
Our goal:

To build a model capable of classifying different types of skin diseases from images

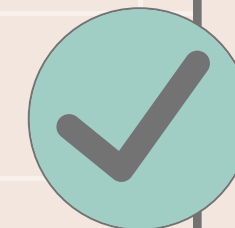


Assignment Brief:

- Build an image-classification pipeline for skin lesions
- Use CNN + deep learning framework (TensorFlow / Keras)
- Deliver accuracy $\geq 80\%$ on validation set
- Validate system with standard metrics - accuracy, sensitivity, specificity and F1 score



OBJECTIVES & REQUIREMENTS_



Personal Goals:

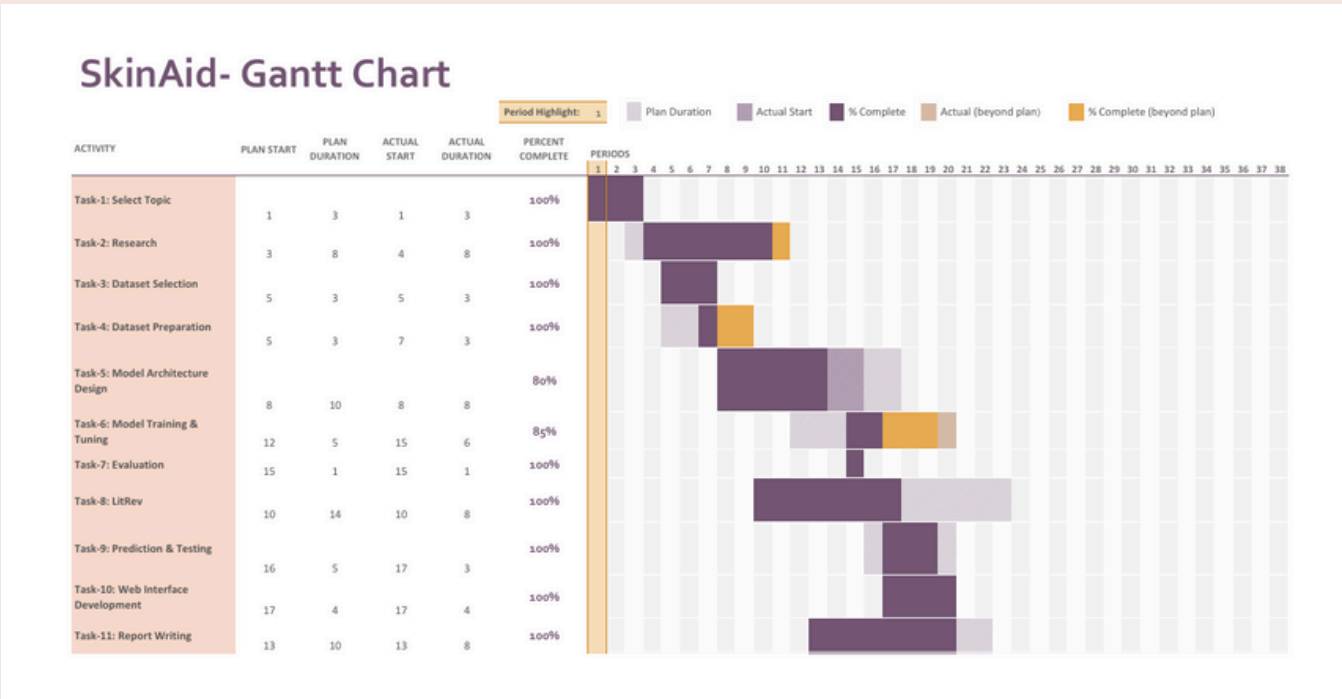
- Modular, maintainable code structure
- Data-augmentation to improve generalization
- Simple web application for live uploads



PLANNING & DESIGN_

TASK-BASED
EXECUTION
STRATEGY

PRODUCT
DESIGN



Tools, visual plans, and structural logic behind the software system

Visual artefacts such as, diagrams, wireframe and design mock-ups



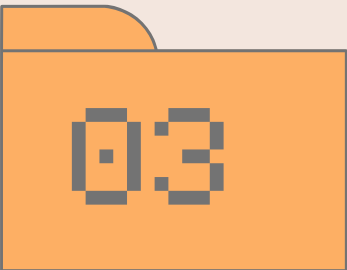
TECH STACK



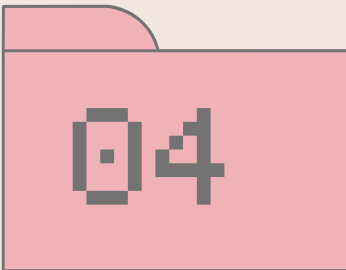
COLAB GPU



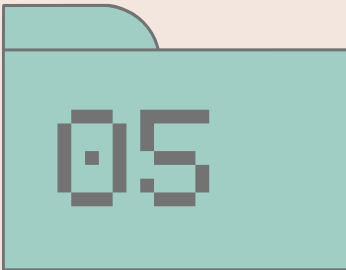
PYTHON3



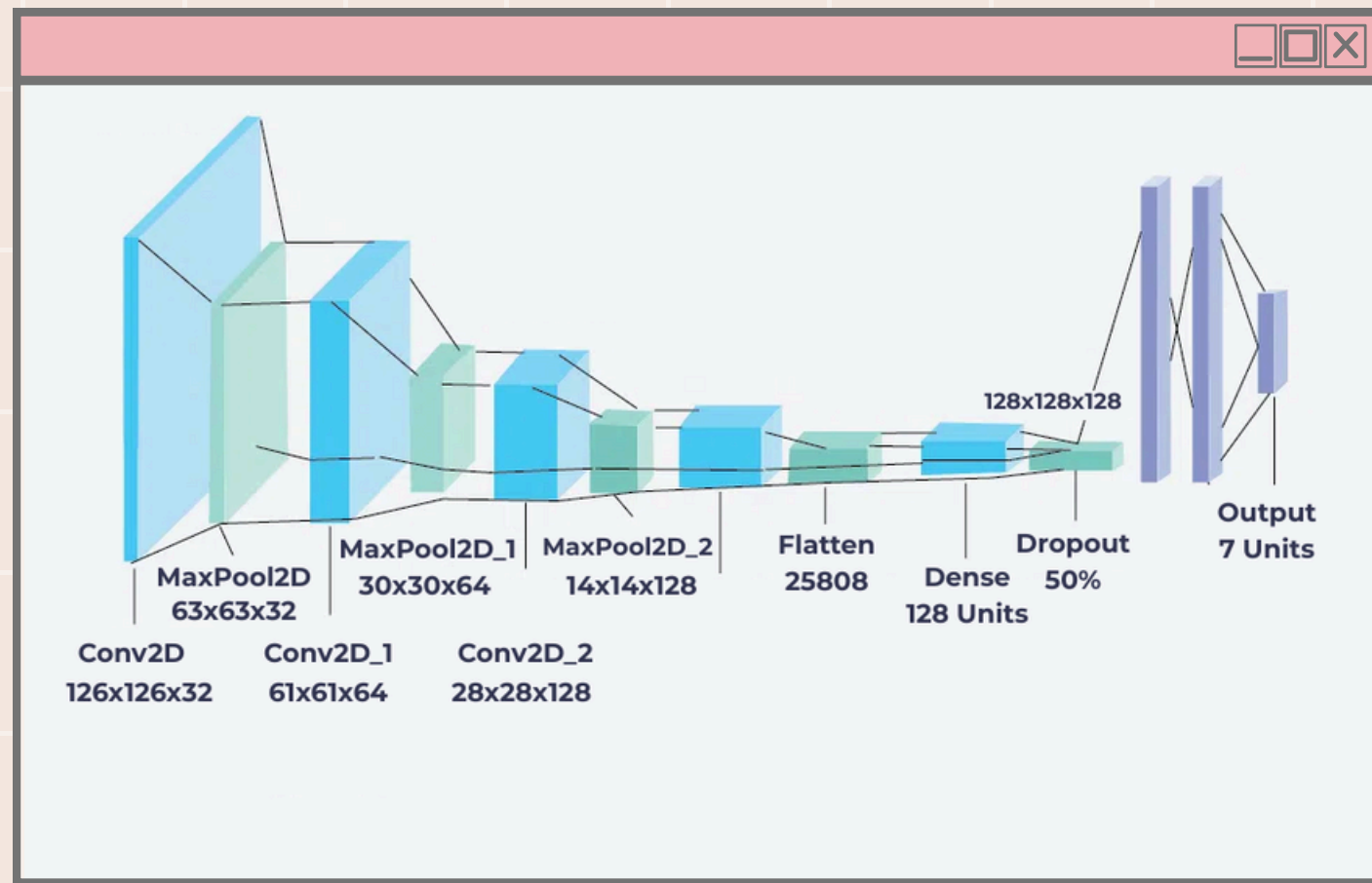
TENSOR-
FLOW



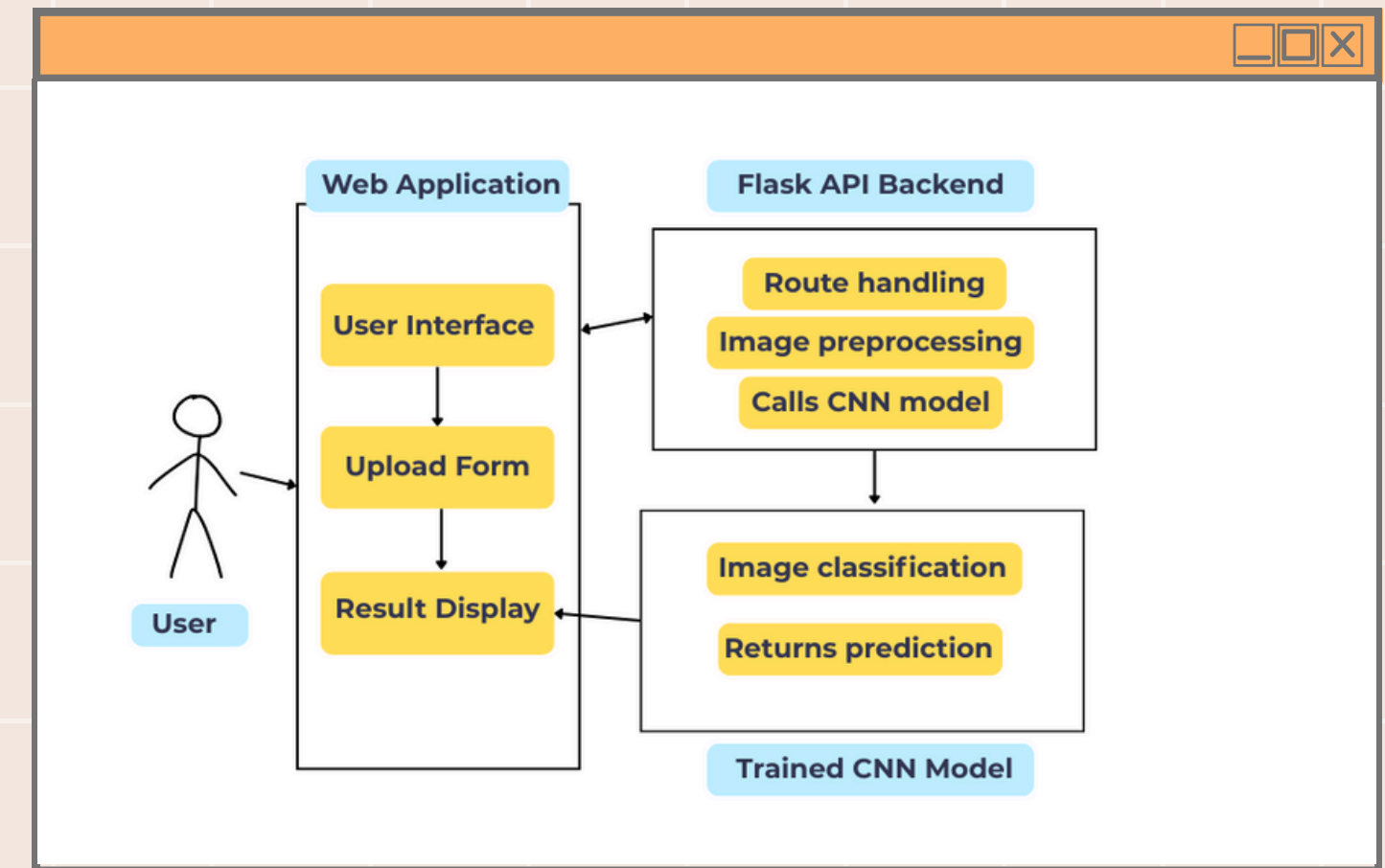
KERAS



FLASK



CNN model architecture

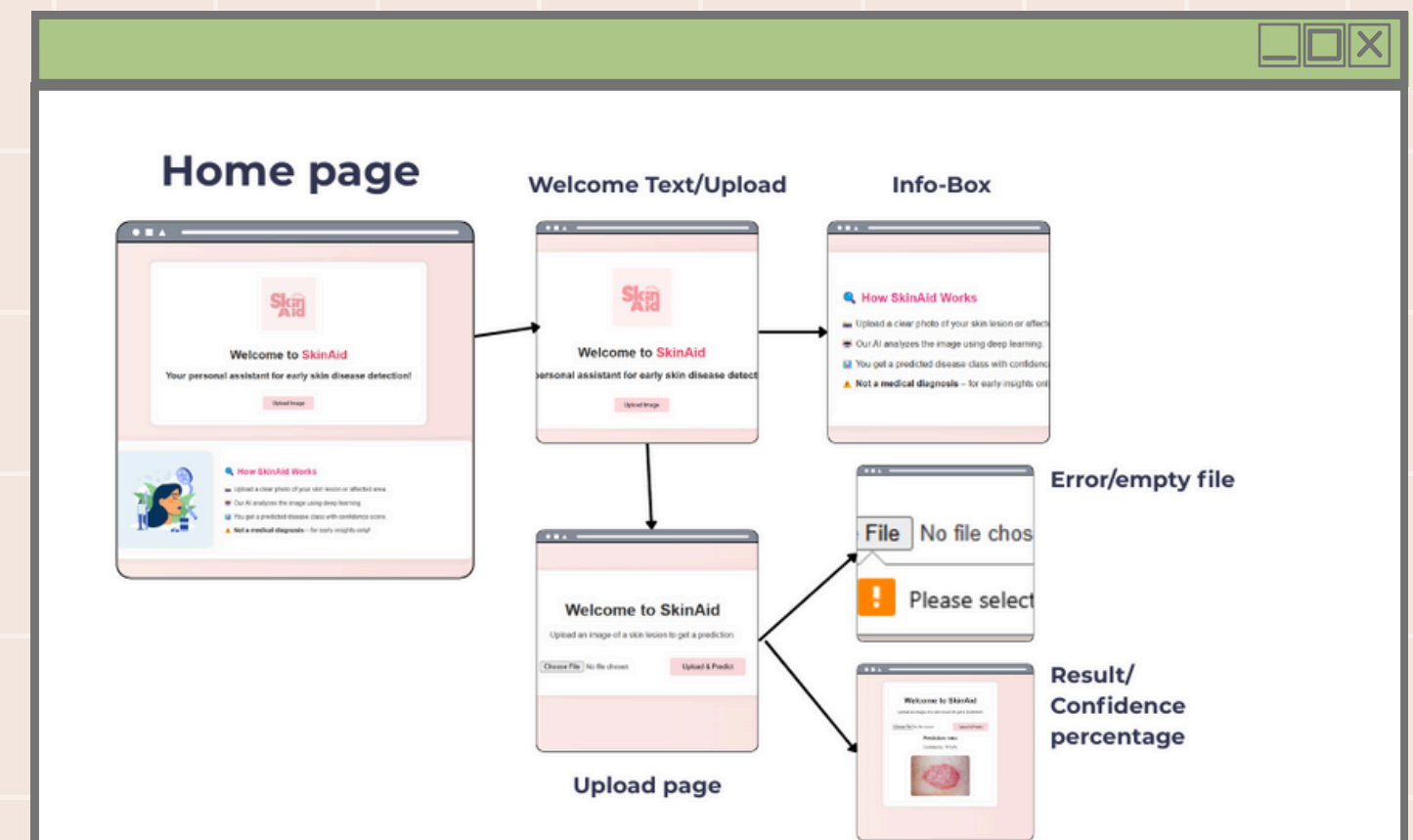
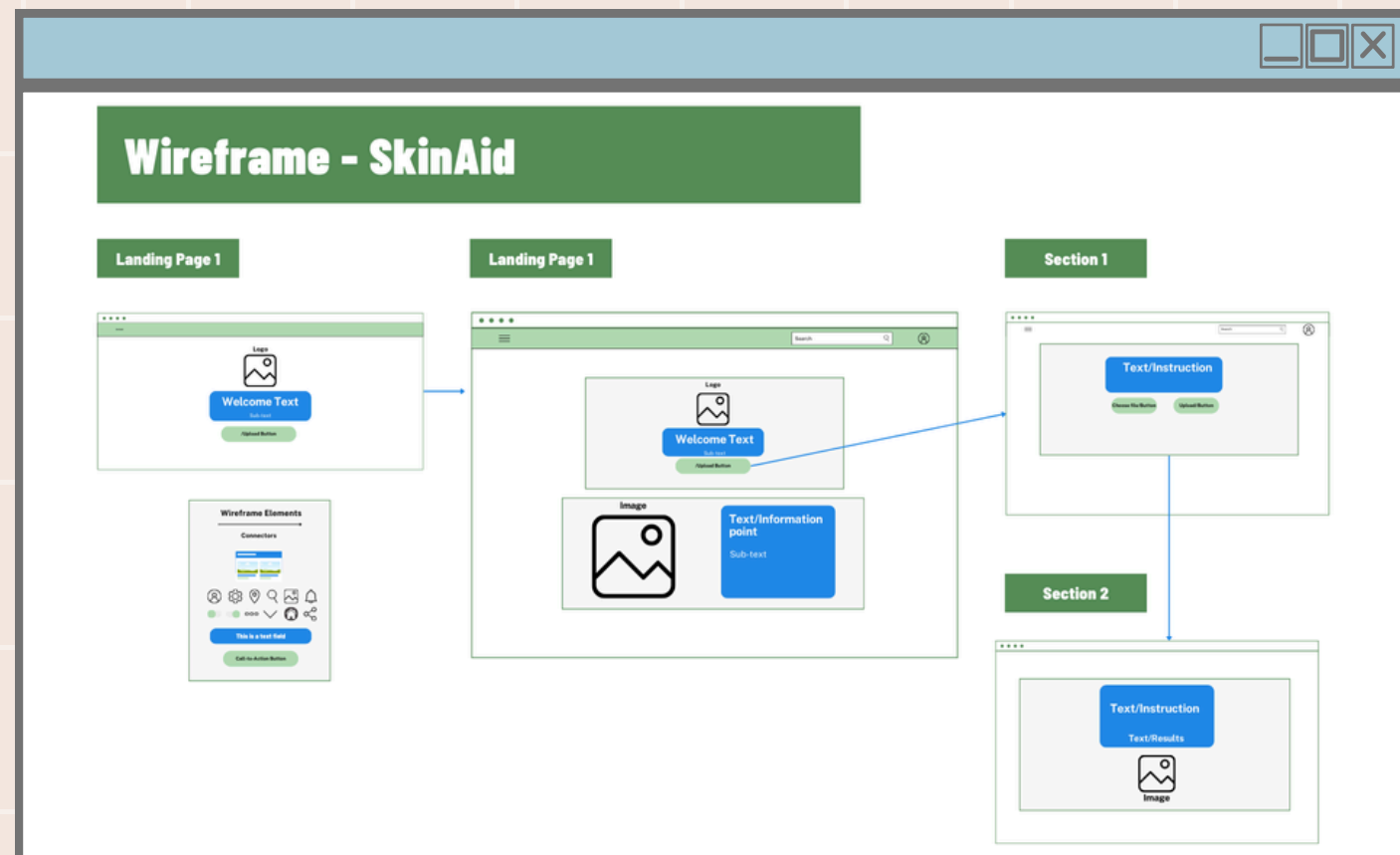


High-level diagram of the system

SYSTEM ARCHITECTURE

System wireframe

WebApp Mock-up





DEVELOPMENT BREAKDOWN_



DATA PREPARATION

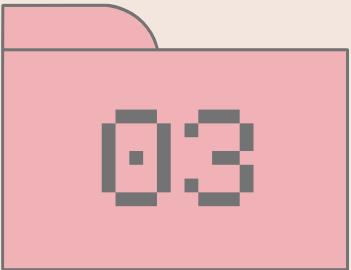
1. Dataset selection
2. Data Augmentation
3. Train-test split

1. Build layers
2. Conv2D
3. MaxPooling
4. Softmax activation

CNN MODEL DESIGN



02



03

TRAINING THE MODEL

1. Use of 10 epochs
2. Evaluation
3. Visualizing results

1. User friendly front-end
2. Integration of CNN model

DEPLOYMENT INTO WEB APP



04

TESTING & OUTPUT_

FUNCTIONAL TESTING

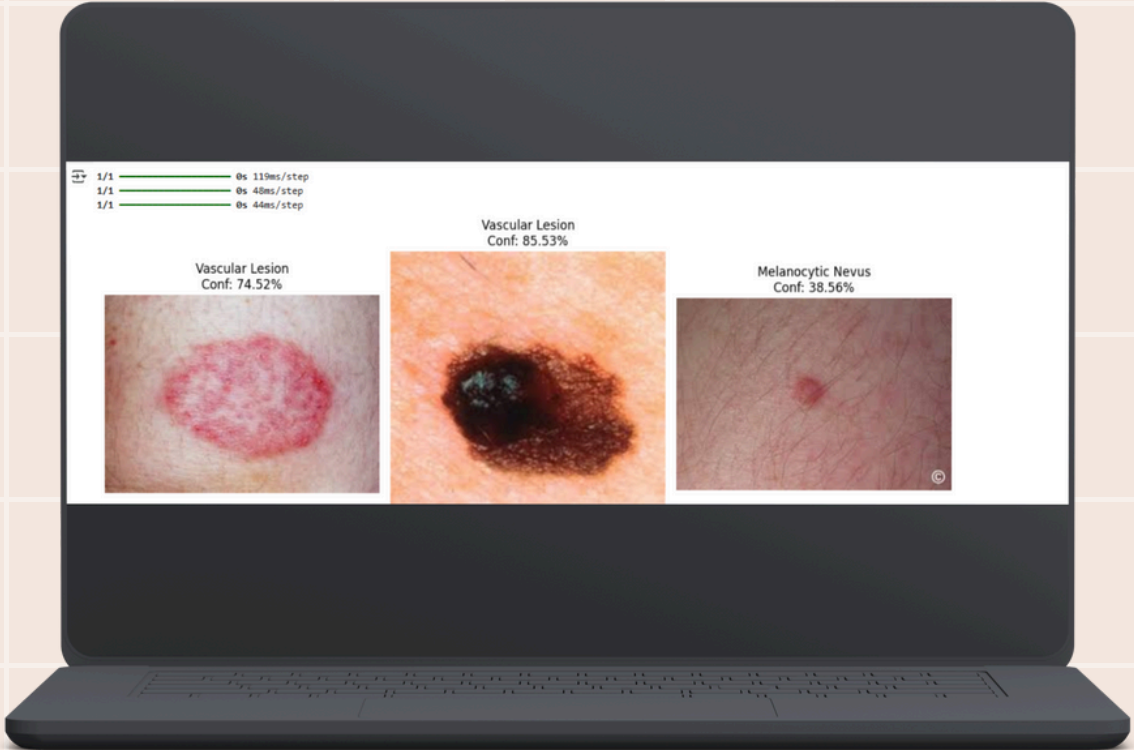
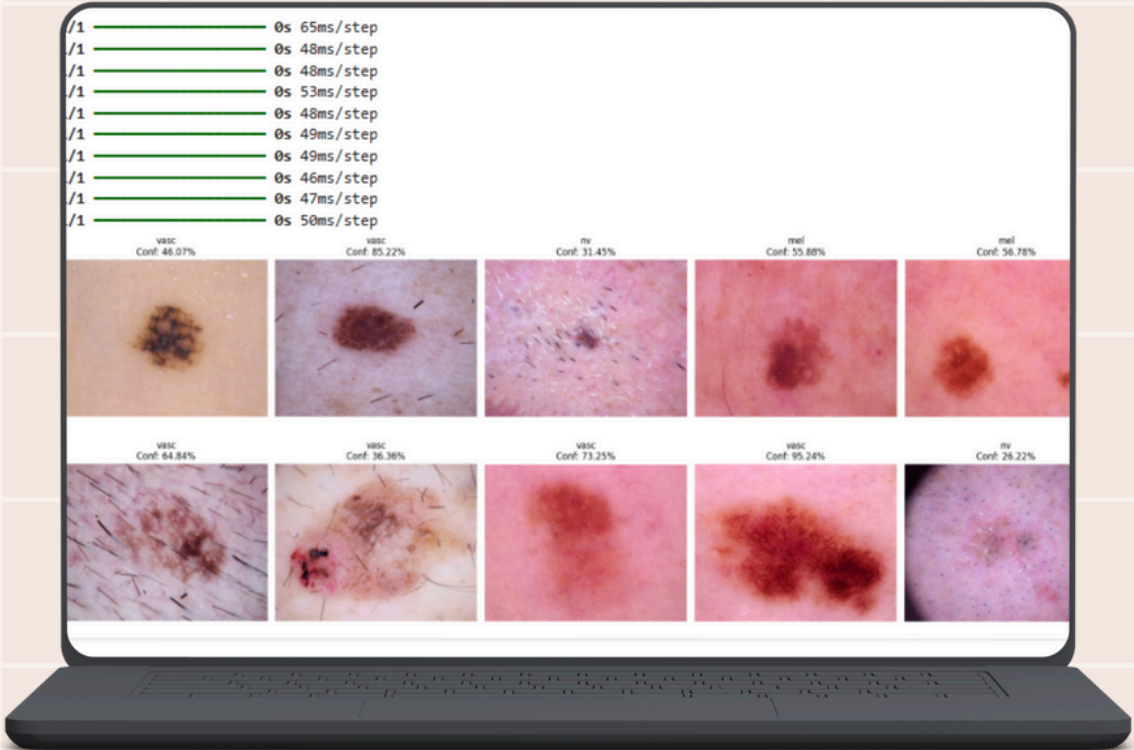
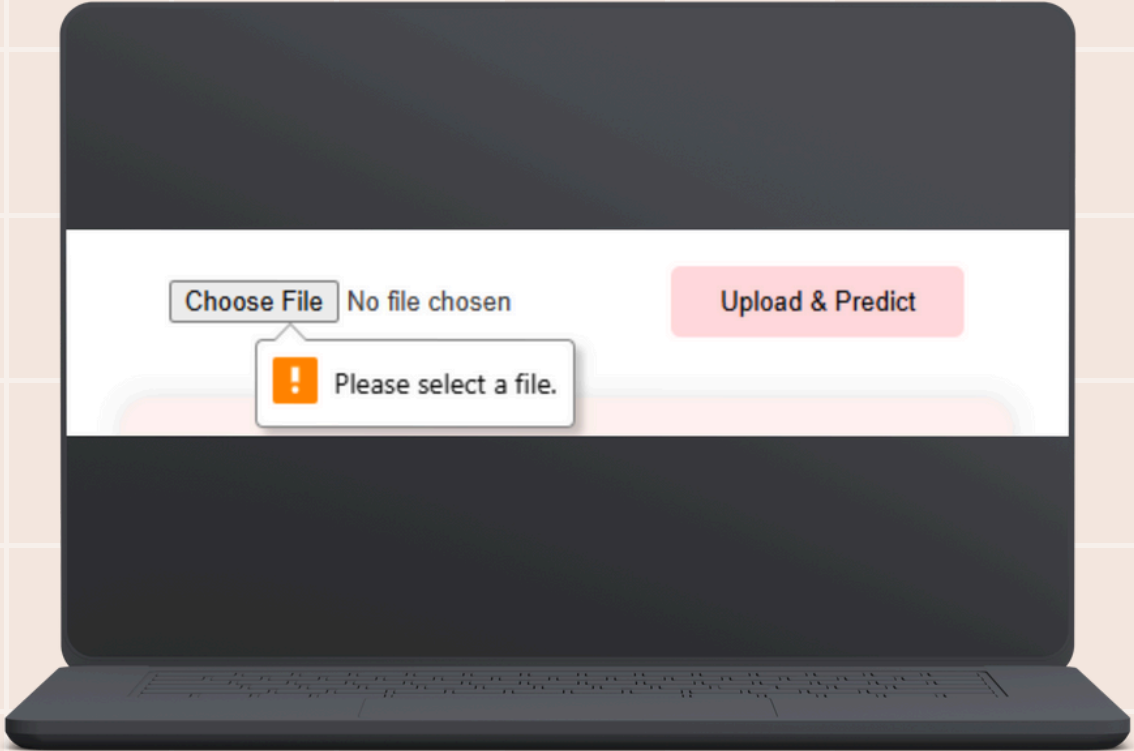
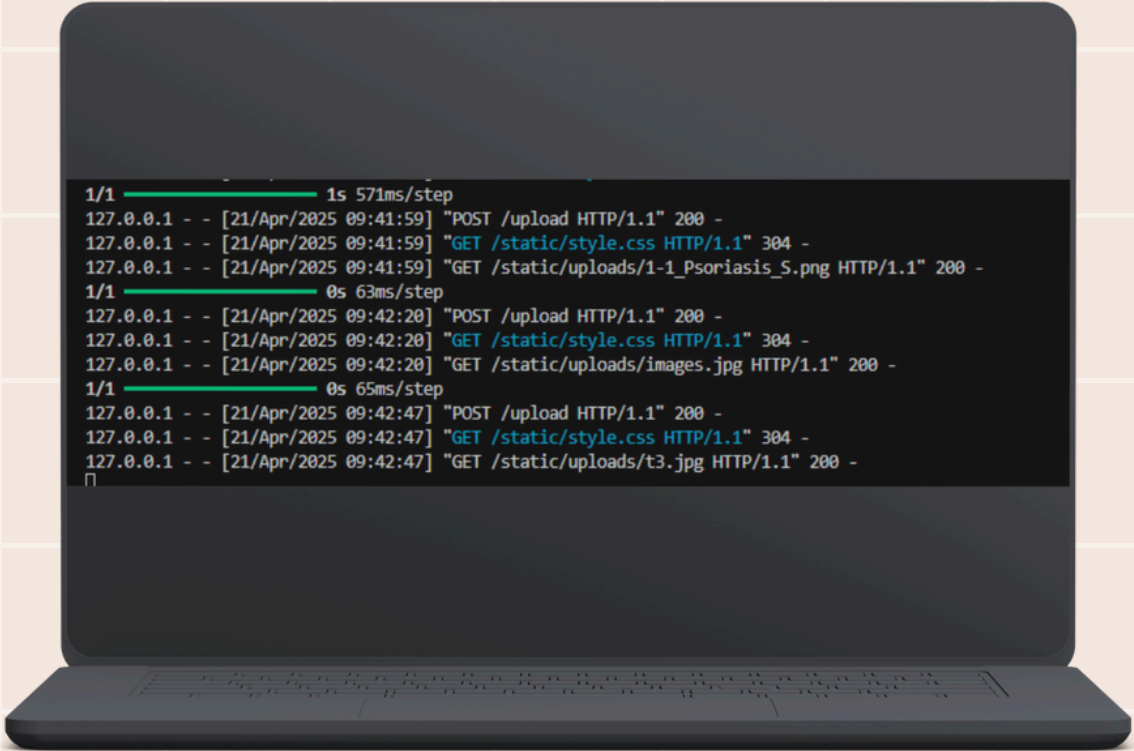
- Different format of images and their acceptance
- Error testing with no file
- Uploading any size of image

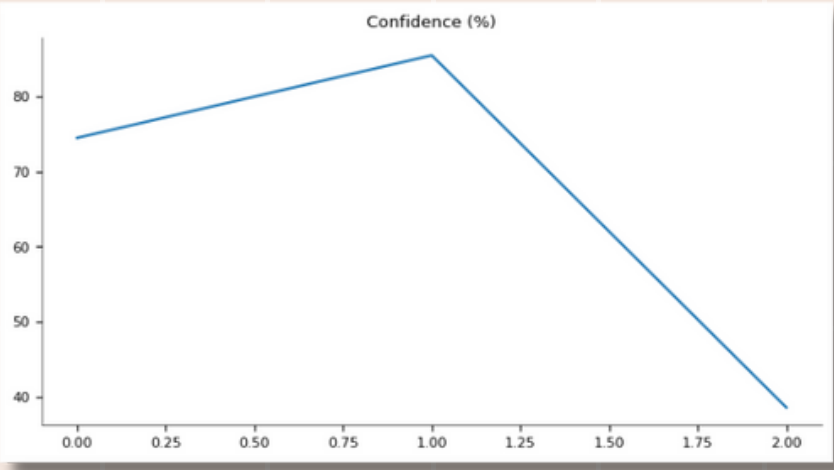
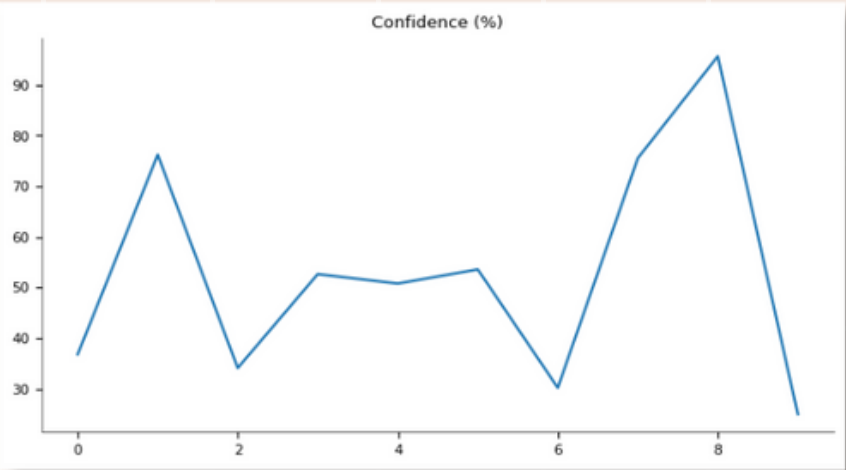
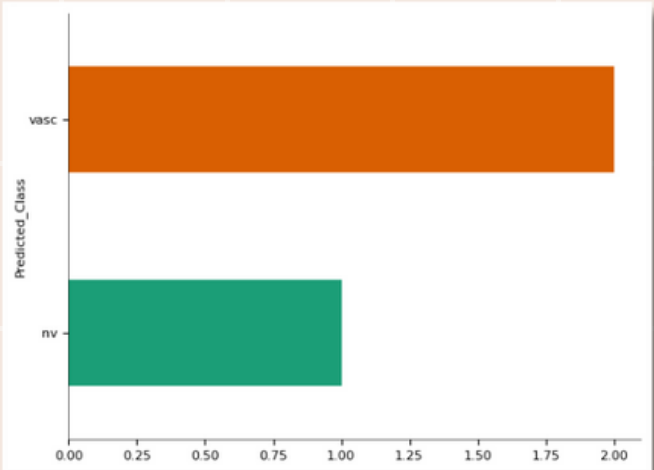
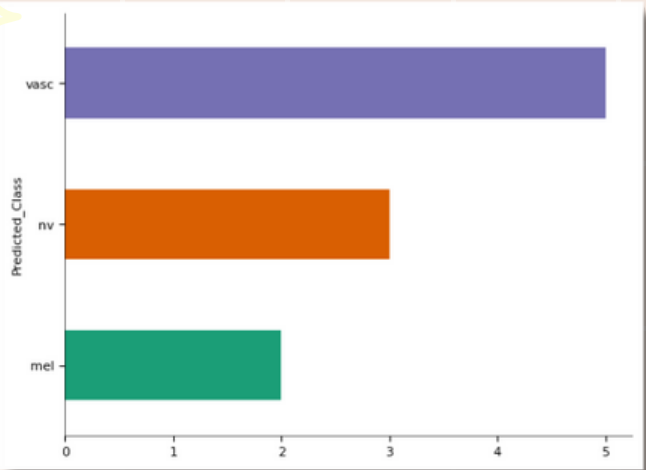
MODEL TESTING

- Validation between different batch of images
- Validation between metadata and unseen data

SYNC TO WEB-APP

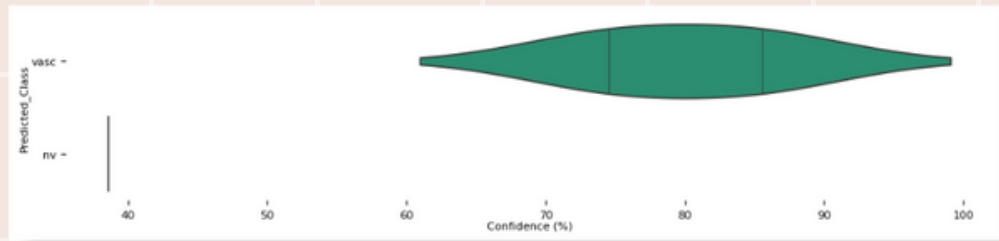
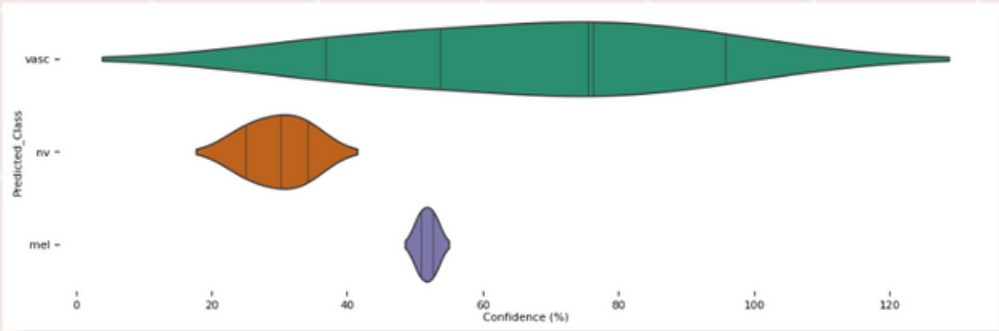
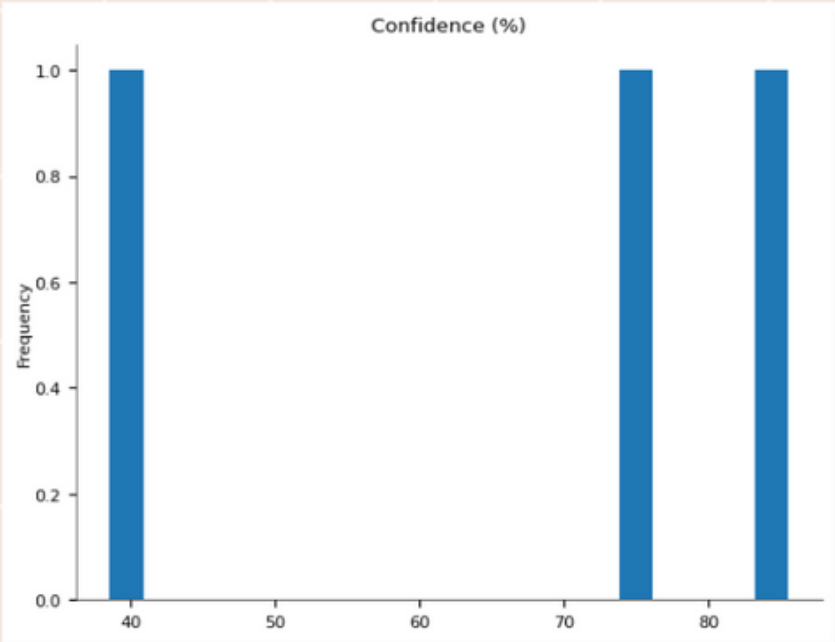
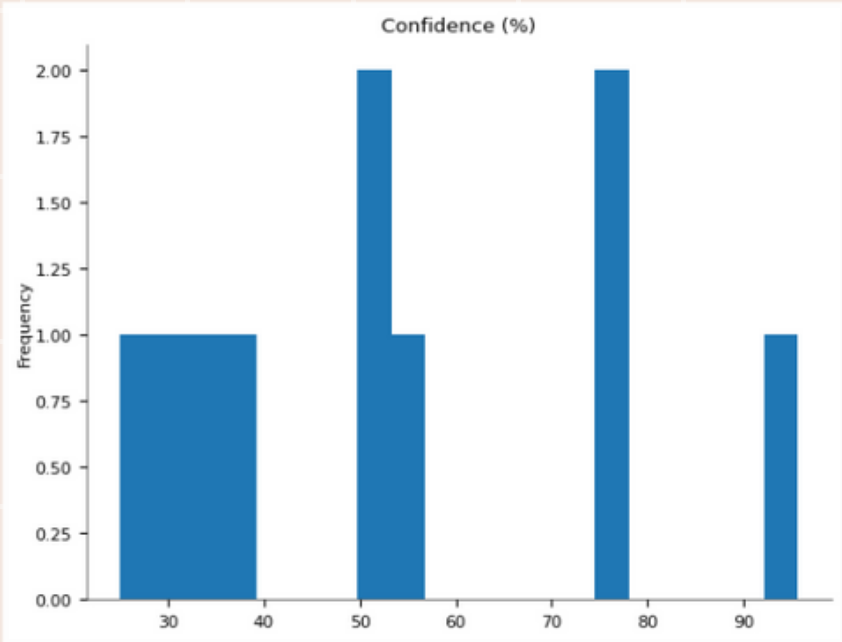
- Comparing the output from CNN model and Web App





TEST- BAR DIAGRAM

TEST- LINE CHART



TEST- BAR GRAPH

TEST- VIOLIN PLOT



REFLECTION_

- **Objectives Met:** CNN → $\geq 80\%$ accuracy & Flask UI → real-time image classification
- **Methodology Wins:** Agile + Fine tuning iterations drove validation from $\sim 71\%$ to $\sim 83\%$
- **Technical Skills:** End-to-end DL pipeline, data-augmentation, model tuning
- **Soft Skills:** Sprint planning, feedback integration (PiP), balancing ML & Web Dev



CONCLUSION_

- **SkinAid Delivered:** A portable .keras model + Flask app for fast (< 3 s) skin-disease support
- **Real-World Promise:** Scalable decision-support in low-resource or early-screening scenarios
- **Future Work:** Add explainability (Grad-CAM/saliency), mobile app, broader clinical validation & diverse datasets
- **Final Thought:** AI-driven dermatology can democratize care - SkinAid is a strong first step



THANK YOU!