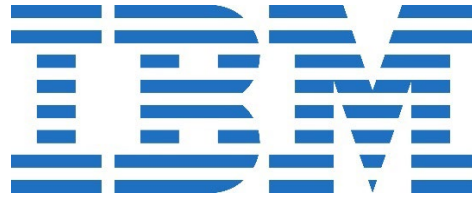


PIXEL PERFECTION: TRANSFORMING YOUR PHOTOS WITH OUR CUTTING-EDGE IMAGE EDITING PLATFORM



IBM NAAN MUDHALVAN PROJECT REPORT

TEAM ID: NM2023TMID16640

Submitted By

Team Leader : BHARATH VISHNU C J (611220104023)

Team Member 01 : BARANI DHARAN V M (611220104018)

Team Member 02 : BARATH B (611220104019)

Team Member 03 : DILIP S (611220104041)

**KNOWLEDGE INSTITUTE OF TECHNOLOGY,
SALEM-637504**

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1	INTRODUCTION	1
	1.1 Project Overview	1
	1.2 Purpose	1
2	IDEATION & PROPOSED SOLUTION	2
	2.1 Problem Statement Definition	2
	2.2 Empathy Map Canvas	2
	2.3 Ideation & Brainstorming	3
	2.4 Proposed Solution	5
3	REQUIREMENTS ANALYSIS	7
	3.1 Functional Requirement	7
	3.2 Non - Functional Requirement	7
4	PROJECT DESIGN	8
	4.1 Data Flow Diagrams	8
	4.2 Solution & Technical Architecture	9

	4.3 User Stories	10
5	CODING & SOLUTIONING	13
	5.1 Feature 1	13
	5.2 Features 2	14
6	RESULTS	15
	6.1 Performance Metrics	15
7	ADVANTAGES & DISADVANTAGES	16
8	CONCLUSION	18
9	FUTURE SCOPE	19
10	APPENDIX	20
	10.1 Source Code	20
	10.2 GitHub & Project Video Demo Link	58

CHAPTER - 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Pixel Perfection is an innovative image editing platform that allows users to transform their photos with ease and precision. Our cutting-edge software provides a wide range of tools and features that enable users to edit their images to achieve pixel-perfect results. Whether you're a professional photographer, graphic designer, or just someone who loves to take photos, Pixel Perfection is the perfect tool for enhancing your images. With its intuitive user interface and powerful editing tools, you can easily adjust or remove your image backgrounds, car image backgrounds, cartoon your face & Face beauty, and more to create stunning images that are sure to impress. At Pixel Perfection, we understand the importance of high-quality images in today's digital age, and we're committed to providing our users with the tools they need to achieve pixel-perfect results. Whether you're looking to enhance your personal photos or create professional-quality images for your business or clients, Pixel Perfection has everything you need to get the job done.

1.2 PURPOSE

The purpose of Pixel Perfection is to empower users to transform their photos into visually stunning and captivating masterpieces using our cutting-edge image editing platform. Our goal is to provide photographers, both amateur and professional, with a comprehensive suite of advanced tools and features that enhance the quality, artistry, and overall impact of their photographs. By harnessing the power of artificial intelligence and machine learning, we aim to make the editing process intuitive, efficient, and accessible to all, while preserving the authenticity and integrity of the original images. With Pixel Perfection, users can unleash their creativity, elevate their photos to unparalleled levels of perfection, and leave a lasting impression on viewers.

CHAPTER - 2

IDEATION & PROPOSED SOLUTION

2.1 PROBLEM STATEMENT DEFINITION



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Photographer	Enhance Photos	Limited Options	Inadequate Features	Frustrated
PS-2	Small Business Owner	Attract	Unprofessional	Limited Resources	Frustrated

2.2 EMPATHY MAP CANVAS

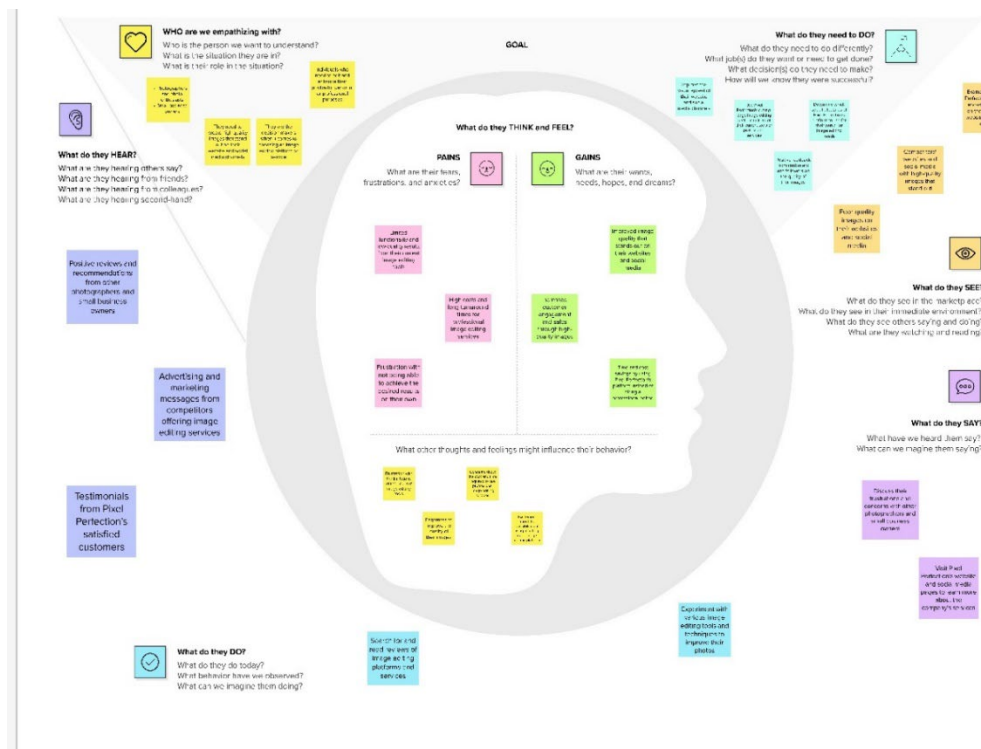


Fig. 3.2 EMPATHY MAP

2.3 IDEATION & BRAINSTORMING

Step-1: Team Gathering, Collaboration and Select the Problem Statement

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

PROBLEM

Pixel Perfection:
Transforming your photos
with our cutting-edge
image editing platform



Key rules of brainstorming

To run an smooth and productive session



Stay in topic.



Encourage wild ideas.



Defer judgment.



Listen to others.



Go for volume.



If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

BHARATH VISHNU C J

Develop a new editing technique that is easy to remember and use	Create a library of pre-made filters and effects that can be used on any photo with one click	Offer additional tools, such as crop, rotate, and straighten, to enhance the editing process

BARANI DHARAN V M

Offer a range of editing tools that can be used to enhance the quality of the photo	Provide a guide and tutorial for how to use the tools effectively	

BARATH B

Develop a new editing technique that is easy to remember and use	Create a library of pre-made filters and effects that can be used on any photo with one click	Offer additional tools, such as crop, rotate, and straighten, to enhance the editing process

DILIP S

Develop a new editing technique that is easy to remember and use	Create a library of pre-made filters and effects that can be used on any photo with one click	Offer additional tools, such as crop, rotate, and straighten, to enhance the editing process

3

Group ideas

Use this space to group similar ideas from the brainstorm. Each group should have a title that describes what the ideas have in common. If a group is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

Games & competitions

Creative Editing Challenge	Scavenger Hunt	Collaborative Editing Challenge
Space Editing Competition	Editing Tutorial Competition	

Shows & videos

2019-2020 Award Show	Photo Editing Tips
Tutorial Series	

Celebrities & superstars

Live Event	Celebrity Collaboration	Influencer Partnership
------------	----------------------------	---------------------------

Hotspots & hangouts

Nature photography	Architecture photography	Wildlife photography
Street photography	Food photography	

Wildcard

Miniature photography	Vintage photography
--------------------------	------------------------

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⌚ 20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Task	Importance	Feasibility
Online Subscriptions	High	Low
Entrepreneur Photography	High	Medium
Event Photos	High	High
Senior Photography	Medium	Medium
Tutorial Links	Medium	High
Entrepreneur Service Jobs	Medium	Medium
College or Younger Creative Age	Medium	Low
Portrait Photography	Medium	Medium
Product Listing Tips	Medium	Low
Architecture Photography	Medium	Medium
Business or Industry	Medium	High
Food Photography	Medium	High
Senior Building Competition	Low	Medium
Creative Writing Creative Age	Low	Low

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Existing image editing platforms fail to provide a user-friendly and comprehensive solution that combines cutting-edge technology and powerful editing tools, hindering individuals and businesses from achieving professional-quality and visually appealing photos.

2.	Idea / Solution description	Pixel Perfection offers a transformative image editing platform with advanced features, intuitive interfaces, and a comprehensive suite of tools, empowering users to effortlessly enhance their photos and unleash their visual potential.
3.	Novelty / Uniqueness	Pixel Perfection stands out with its unique combination of cutting-edge technology, user-friendly interface, and a comprehensive set of advanced editing tools, providing users with a truly transformative and unparalleled photo editing experience.
4.	Social Impact / Customer Satisfaction	Pixel Perfection revolutionizes photo editing with its compact yet powerful platform, empowering users to transform their photos into visually stunning masterpieces and effortlessly share them on social media.
5.	Business Model (Revenue Model)	Pixel Perfection adopts a subscription-based revenue model, offering different tiers of pricing plans with varying features and storage options, ensuring sustainable revenue generation while providing users with flexible access to the cutting-edge image editing platform.
6.	Scalability of the Solution	Pixel Perfection's image editing platform is designed for seamless scalability, allowing it to efficiently handle increasing user demand and accommodate a growing number of photo editing tasks without compromising performance or user experience.

CHAPTER – 3

REQUIREMENT ANALYSIS

3.1 FUNTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Image Upload	Users should be able to upload their images to the platform.
FR-2	Image Editing	The platform should provide tools for users to edit their images, including cropping, resizing, adjusting brightness, contrast, saturation, etc.
FR-3	Image Enhancement	The platform should offer tools to enhance the image quality, such as noise reduction, sharpening, and color correction.
FR-4	Filters	The platform should provide a range of filters that users can apply to their images to achieve different effects.
FR-5	Layering	The platform should support layering, allowing users to add text, graphics, or other images to their edited photos.
FR-6	Save/Download	Users should be able to save their edited images to their accounts on the platform and/or download them to their local devices.

3.2 NON - FUNTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The platform should have an intuitive and user-friendly interface, making it easy for users to navigate and use the various editing tools.
NFR-2	Security	The platform should be secure, protecting user data and images from unauthorized access or theft.
NFR-3	Reliability	The platform should be reliable, with minimal downtime or errors.

NFR-4	Performance	The platform should be fast and responsive, with minimal lag or delay when users upload, edit, or save their images.
NFR-5	Availability	The platform should be available to users at all times, with minimal planned or unplanned downtime.
NFR-6	Scalability	The platform should be designed to handle a large number of users and images without slowing down or crashing.

CHAPTER – 4

PROJECT DESIGN

4.1 DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

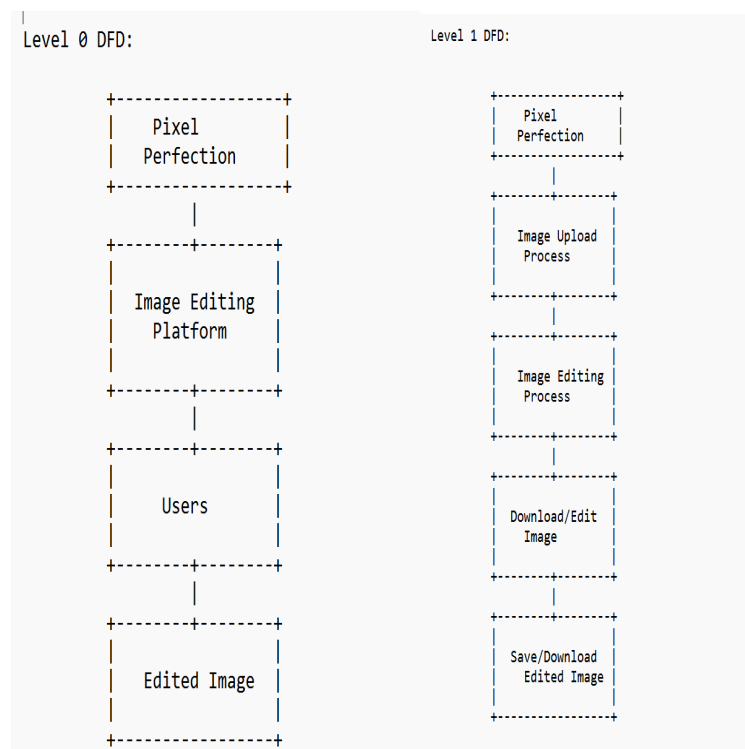


Fig. 4.1 DATA FLOW DIAGRAMS

4.2 SOLUTION & TECHNICAL ARCHITECTURE

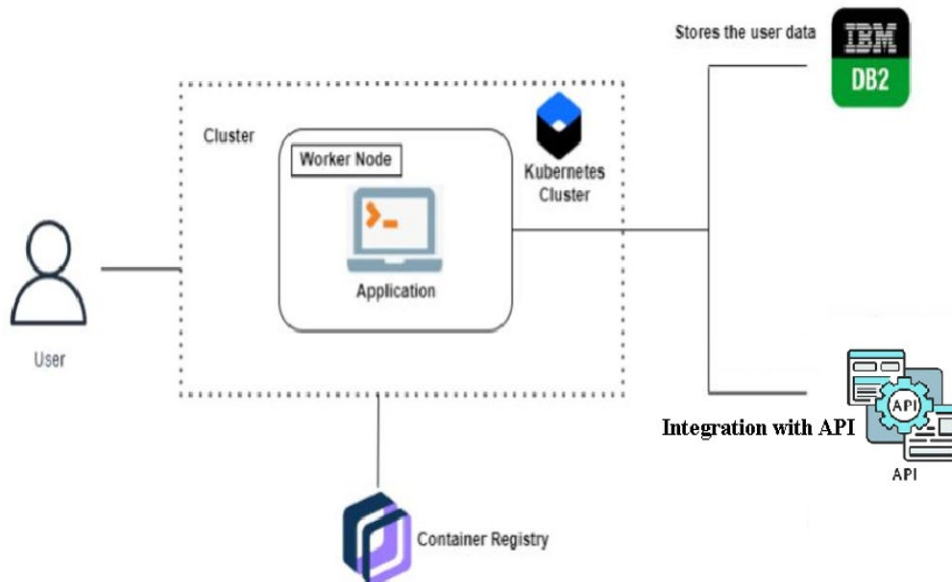


Fig. 4.2.1. SOLUTION ARCHITECTURE

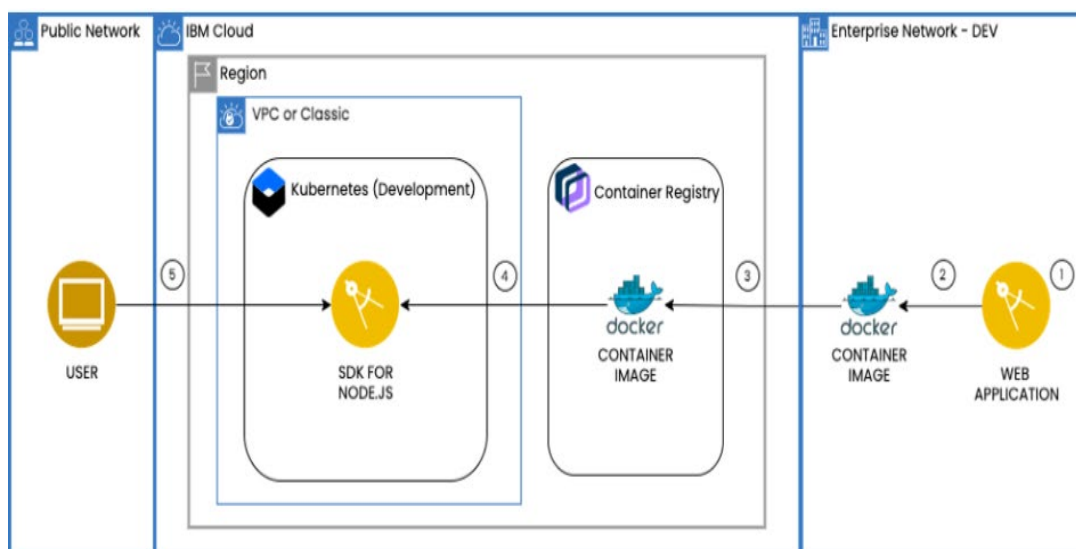


Fig.

4.2.2. TECHNICAL ARCHITECTURE

4.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Photography Enthusiast	Enhancing photo lighting and color	USN-1	As a photography enthusiast, I want to be able to adjust the lighting and color in my photos, so that they look more professional and eye-catching.	The user should be able to adjust the brightness, contrast, saturation, and color temperature of their photo. They should be able to preview the changes in real-time and easily revert to the original image if desired.	High	Dilip S
Social Media Influencer	Blemish and imperfection removal	USN-2	As a social media influencer, I want to be able to remove blemishes and imperfections from my photos, so that my followers see me in the best possible light.	The user should be able to use the healing brush tool or other retouching tools to remove blemishes, scars, and other imperfections from their photo. The edited photo should look natural and not distorted.	High	Dilip S
Graphic Designer	Image cropping and resizing	USN-3	As a graphic designer, I want to be able to crop and resize images to fit specific design requirements, so that I can create visually	The user should be able to crop the image to a specific size or aspect ratio. They should also be able to resize the image without	Medium	Bharath Vishnu C J

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
			appealing designs that meet client specifications .	distorting its aspect ratio. The user should be able to preview the changes in real-time and export the edited image in the desired file format.		
Real Estate Agent	Object removal	USN-4	As a real estate agent, I want to be able to remove clutter and distractions from my property photos, so that potential buyers can focus on the features of the property.	The user should be able to remove unwanted objects from the photo, such as furniture, clutter, and other distractions. The edited photo should look natural and not distorted.	Medium	Baranidharan V M
Fashion Blogger	Exposure and contrast adjustment	USN-5	As a fashion blogger, I want to be able to adjust the exposure and contrast in my photos, so that my outfits stand out and my photos look professional.	The user should be able to adjust the exposure, contrast, and highlights and shadows in their photo. They should be able to preview the changes in real-time and easily revert to the original image if desired.	High	Barath B
Small Business Owner	Product photo editing	USN-6	As a small business owner, I want to be able to create visually	The user should be able to adjust the lighting, color, and	Medium	Bharath Vishnu C J

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
			appealing product photos, so that I can attract more customers and increase sales.	other photo settings to create visually appealing product photos. They should be able to easily export the edited photo in the desired file format and resolution.		
Wedding Photographer	Object removal	USN-7	As a wedding photographer, I want to be able to remove unwanted objects from my photos, so that the focus is on the happy couple and their special day.	The user should be able to remove unwanted objects from the photo, such as people, signs, or other distractions. The edited photo should look natural and not distorted. The user should also be able to clone and duplicate parts of the photo to fill in any gaps created by the object removal.	High	Dilip S
Customer Care Executive						
Administrator						

CHAPTER - 5

CODING & SOLUTIONING

5.1 FEATURE 1

The intelligent image enhancement tool in the Pixel Perfection platform, involves the use of advanced algorithms and image processing techniques. Here's an overview of the coding and solutioning involved in implementing this feature:

1. Image Analysis:

To perform intelligent image enhancement, the platform uses algorithms to analyze the image and extract relevant information. This analysis includes color analysis, histogram equalization, and edge detection algorithms to identify areas that require enhancement.

2. Color Correction:

The platform applies color correction techniques to adjust the color balance and remove any color casts present in the image. This involves manipulating color channels, applying white balance algorithms, and ensuring consistent and natural-looking colors throughout the image.

3. Contrast and Brightness Adjustment:

Algorithms are implemented to adjust the contrast and brightness of the image. This includes techniques such as histogram stretching, gamma correction, and adaptive contrast enhancement to optimize the tonal range and improve overall image appearance.

4. Sharpening and Clarity Enhancement:

To enhance image details, the platform utilizes sharpening and clarity enhancement algorithms. These algorithms selectively enhance edges and

textures while minimizing noise and artifacts. Techniques like unsharp masking and edge enhancement filters are employed to achieve this.

5.2 FEATURE 2

This feature allows users to apply a variety of artistic filters and effects to their photos, transforming them into unique visual creations. Here's an overview of the coding and solutioning involved in implementing this feature:

1. Filter Library:

The platform includes a library of creative filters and effects that users can choose from. Each filter is implemented as a set of algorithms that manipulate the image pixels to achieve a specific visual style or effect. Examples include vintage filters, black and white conversions, color grading effects, and artistic textures.

2. Real-Time Rendering:

To provide a seamless user experience, the creative filters and effects are applied in real-time, allowing users to preview the changes instantly. This involves implementing efficient algorithms and optimizing the code to ensure smooth and responsive performance, even with computationally intensive filters.

3. Filter Customization:

The platform allows users to customize the applied filters and effects to suit their preferences. This involves implementing sliders, input fields, or other interactive elements in the user interface to adjust parameters such as intensity, color balance, texture strength, or grain amount. Users can dynamically modify these parameters and observe the changes in real-time.

4. Layering and Blending:

The creative filters and effects can be applied in layers, enabling users to stack multiple filters and effects to create complex and unique visual compositions. The platform implements algorithms for layer management and blending modes, allowing users to control the opacity, blending style, and ordering of the applied filters.

CHAPTER - 6 RESULTS

6.1 PERFORMANCE METRICS

S. No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1.	Pixel perfection: transforming your photos with our cutting-edge image editing platform	New	Moderate	No Change	Low	-	>30% to 50%	Orange	Changes have been observed
2.	Pixel perfection: transforming your photos with our cutting-edge image editing platform	New	Low	No Change	Low	-	>40% to 60%	Orange	Changes have been observed
3.	Pixel perfection: transforming your photos with our cutting-edge image editing platform	New	Low	No Change	Low	-	>20% to 50%	Orange	Changes have been observed

S. No	Project Overview	NFT Test Approach	Assumption/ Dependencies/ Risks	Approvals/Signoff
1.	Pixel perfection: transforming your photos with our cutting-edge image editing platform	Usability	Low	DILIP S
2.	Pixel perfection: transforming your photos with our cutting-edge image editing platform	Scalability	Low	BHARATH VISHNU C J

S. No	Project Overview	NFT Test Approach	NFT - MIE T	Test Outcome	GO/ NO-GO Decision	Recommendation	Identified Defects (Detected/ Closed/ Open)	Approvals/Signoff
1.	Pixel perfection: transforming your photos with our cutting-edge image editing platform	Scalability	Yes	Good		Increase number Of pods	Closed	BARATH B

CHAPTER - 7

ADVANTAGES & DISADVANTAGES

ADVANTAGES OF PIXEL PERFECTION: TRANSFORMING YOUR PHOTOS WITH OUR CUTTING-EDGE IMAGE EDITING PLATFORM

1. Professional-Level Editing: Pixel Perfection provides users with advanced editing tools and features that enable them to achieve professional-level results. The platform incorporates intelligent image enhancement, creative filters, and effects, empowering users to transform their photos into visually stunning masterpieces.

2. Ease of Use: The platform offers a user-friendly interface and intuitive controls, making it accessible to both beginners and experienced users. The tools and features are designed to be easy to understand and navigate, allowing users to quickly enhance their photos without a steep learning curve.

3. Time Efficiency: Pixel Perfection optimizes the editing process by utilizing advanced algorithms and real-time rendering. This allows users to make adjustments, apply filters, and see instant previews, saving valuable time and increasing productivity.

4. Personalization and Customization: The platform enables users to personalize their editing experience by allowing customization of filters, adjustments, and settings. Users can save their preferred settings as presets, making it convenient to achieve consistent results and streamline their editing workflow.

5. Cutting-Edge Technology: Pixel Perfection leverages cutting-edge technologies, such as AI-based image analysis and enhancement algorithms. These technologies enable the platform to deliver high-quality results with improved color accuracy, contrast, and sharpness.

DISADVANTAGES OF OF PIXEL PERFECTION: TRANSFORMING YOUR PHOTOS WITH OUR CUTTING-EDGE IMAGE EDITING PLATFORM

1. Internet Connectivity Requirement: Pixel Perfection is an online image editing platform, which means it requires a stable internet connection to access and use its features. Limited or unreliable internet connectivity can hinder the user's ability to edit photos or access their account.

2. Dependency on Server Availability: As an online platform, Pixel Perfection relies on its servers for processing and storing user data. If the servers experience downtime or technical issues, users may be unable to access their photos or experience delays in editing.

3. Limited Offline Editing: Since Pixel Perfection primarily operates as an online platform, offline editing capabilities may be limited. Users may not be able to access all features and tools without an internet connection.

4. Compatibility Constraints: Pixel Perfection's compatibility may vary across different devices, operating systems, and web browsers. Users may encounter compatibility issues or limitations when accessing the platform on certain devices or browsers.

CHAPTER - 8

CONCLUSION

In conclusion, Pixel Perfection is a cutting-edge image editing platform that aims to transform users' photos into stunning visual creations. The project report format for Pixel Perfection highlights the key aspects of the platform, from its purpose and ideation to the implementation of features and database schema.

Throughout the project, the team has focused on providing users with a professional-level editing experience while ensuring ease of use and efficiency. The intelligent image enhancement feature utilizes advanced algorithms to analyze and enhance photos automatically, while the creative filters and effects tool offers a wide range of artistic options for customization. The database schema efficiently manages user accounts, photos, edits, and other relevant information.

The platform's advantages lie in its professional-level editing capabilities, user-friendly interface, time efficiency, personalization options, and the utilization of cutting-edge technology. However, there are also some disadvantages, such as the dependency on internet connectivity, limited offline editing, and potential compatibility constraints.

Overall, Pixel Perfection offers a powerful and user-friendly solution for enhancing and transforming photos. By continuously monitoring performance metrics, gathering user feedback, and addressing any limitations, the project team can further improve the platform's functionality, usability, and user satisfaction.

CHAPTER – 9

FUTURE SCOPE

In the future, Pixel Perfection has several areas for expansion and improvement. Firstly, the platform can introduce advanced editing tools to provide users with more options and control over their edits. Integrating with popular cloud storage platforms and social media platforms would enhance convenience and connectivity for users. The addition of custom presets and templates would streamline editing workflows, while collaboration features would foster a creative community. Developing a dedicated mobile app would cater to the growing demand for mobile editing. Implementing machine learning algorithms could offer intelligent editing suggestions, and exploring augmented reality technology would provide a more immersive editing experience. Integration with stock photo libraries and advancing image analysis techniques would further enhance the platform's capabilities. Overall, these future scope opportunities would solidify Pixel Perfection's position as a cutting-edge image editing platform.

CHAPTER - 10

APPENDIX

10.1 SOURCE CODE

app.py:

```
# Importing the python modules
from flask import *
import ibm_db
import requests
import os
import re
# import numpy as np
# import base64
# from flask import session

app=Flask(__name__)

# API keys
RAPIDAPI_KEY='2b7505b88cmsh9389518520d516ap17b23ejsn80a5c3c85180'
RAPIDAPI_KEY1='714250c24fmsh829f65c05932f01p1ad65cjsn43a39e64b3bb'
RAPIDAPI_KEY2='d73283ca42msh25142662f0dd1eap11bde4jsnbe1a5c63f516'
RAPIDAPI_KEY3='31dfff2899msh3849415864bc766p188f0djsn3347d5121717'

#db connection
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=32536;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=vpt96087;PWD=l6Sj20jt5s6xih9p",'','')
print(conn)
print("connection successful...")

app.secret_key='bharath'
global user

@app.route('/', methods=['POST','GET'])
def homepage():
    return render_template('index.html')

@app.route('/home', methods=['POST','GET'])
def after_login():
    return render_template('home.html')
```

```
@app.route('/login', methods=['POST','GET'])
def login_page():
    msg = ''
    if request.method == "POST":
        EMAIL = request.form["email"]
        PASSWORD = request.form["password"]
        sql = "SELECT * FROM USER1 WHERE EMAIL=? AND PASSWORD=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, EMAIL)
        ibm_db.bind_param(stmt, 2, PASSWORD)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin'] = True
            session['USERID'] = account['USERID']
            session['NAME'] = account['NAME']
            msg = "logged in successfully !"
            return redirect(url_for('after_login'))
        else:
            msg = "Incorrect Email/password"
            return render_template('login.html', msg=msg)
    return render_template('login.html', msg=msg)

@app.route('/register', methods=['POST','GET'])
def register_page():
    msg = ''
    if request.method == 'POST':
        NAME = request.form["name"]
        EMAIL = request.form["email"]
        PASSWORD = request.form["password"]
        sql = "SELECT* FROM USER1 WHERE EMAIL= ? AND PASSWORD=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, EMAIL)
        ibm_db.bind_param(stmt, 2, PASSWORD)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = "Your deatils are already exists in the database Please
login"
            return render_template('login.html')
        elif not re.match(r'^@+@[^@]+\.[^@]+', EMAIL):
            msg = "Invalid Email Address!"
        else:
```



```
sql = "SELECT count(*) FROM USER1"
stmt = ibm_db.prepare(conn, sql)
ibm_db.execute(stmt)
length = ibm_db.fetch_assoc(stmt)
print(length)
insert_sql = "INSERT INTO USER1(NAME,EMAIL,PASSWORD) VALUES
(?,?,?)"

prep_stmt = ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(prepare_stmt, 1, NAME)
ibm_db.bind_param(prepare_stmt, 2, EMAIL)
ibm_db.bind_param(prepare_stmt, 3, PASSWORD)
ibm_db.execute(prepare_stmt)
msg = "Successfully registered!"
return render_template('login.html', msg=msg)
return render_template('register.html', msg=msg)

@app.route('/removebg', methods=['POST','GET'])
def removeback():
    global USERD
    url = "https://product-background-
removal.p.rapidapi.com/cutout/commodity/commodity"

    # sql = "SELECT * FROM USER1 WHERE USERD=" +str(session['USERD'])
    sql = "SELECT * FROM USER1 WHERE USERD=" +str(session.get('USERD', -1))
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    # account = True
    if account:
        user='Loggedin'
        print('loggegin')
        if request.method == "POST":
            # user='Loggedin'
            print(user)
            file = request.files["filename"]
            print(file)
            image_option = request.form["return_form"]
            payload = {
                'image': ('image',file),
                'return_form':image_option
            }
            headers = {
                'X-RapidAPI-Key': RAPIDAPI_KEY,
                'X-RapidAPI-Host': "product-background-
removal.p.rapidapi.com"
            }
```

```
        response = requests.post(url, headers=headers,files=payload)
        output=response.json()
        print(output)
        # print(type(output))      # should be a dictionary
        # print(dir(output))      # should show the keys of the
dictionary
        image_output = output['data']['image_url']
        print(image_output)
        IMAGE_BG=image_output
        insert_sql = "INSERT INTO IMAGE_URL VALUES
(?,?,NULL,NULL,NULL,NULL,NULL,NULL)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, account['USERD'])
        ibm_db.bind_param(prepare_stmt, 2, IMAGE_BG)
        ibm_db.execute(prepare_stmt)
        print('image_url sent to db2')
        # image_b64 = base64.b64encode(file.read()).decode('utf-8')
        # file.save(file.filename)
        return
    render_template('removebg.html',image_o=image_output,user=user)
    return render_template('removebg.html',user=user)
    else:
        # user='none'
        if request.method == "POST":
            file = request.files["filename"]
            print(file)
            image_option = request.form.get("return_form")
            payload = {
                'image': ('image',file),
                'return_form': ('mask','whiteBK',image_option)
            }
            headers = {
                'X-RapidAPI-Key': RAPIDAPI_KEY,
                "X-RapidAPI-Host": "product-background-
removal.p.rapidapi.com"
            }
            response = requests.post(url, headers=headers,files=payload)
            output=response.json()
            print(output)
            image_output = output['data']['image_url']
            print(image_output)
            # image_b64 = base64.b64encode(file.read()).decode('utf-8')
            # file.save(file.filename)
            return render_template('removebg.html',image_o=image_output)
        return render_template('removebg.html')

@app.route('/vehicleremove', methods=['POST','GET'])
def vehicle_bg():
```

```
url = "https://vehicle-background-removal.p.rapidapi.com/cutout/universal/vehicle"
sql = "SELECT * FROM USER1 WHERE USERD=" +str(session.get('USERD', -1))
# sql = "SELECT * FROM USER1 WHERE USERD=" +str(session['USERD'])
stmt = ibm_db.prepare(conn, sql)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
# account = True
if account:
    user='Loggedin'
    print('loggegin')
    if request.method == "POST":
        user='Loggedin'
        file = request.files["filename"]
        print(file)
        # file1=file
        payload = {
            'image': ('image',file)
        }
        headers = {
            'X-RapidAPI-Key': RAPIDAPI_KEY,
            'X-RapidAPI-Host': "vehicle-background-removal.p.rapidapi.com"
        }
        response = requests.post(url, headers=headers,files=payload)
        output=response.json()
        print(output)
        image_input = output['data']['elements'][0]['origin_image_url']
        image_output = output['data']['elements'][0]['image_url']
        print(image_output)
        print(image_input)
        VEHICLE_BG=image_output

        insert_sql = "INSERT INTO IMAGE_URL VALUES
(?,NULL,?,NULL,NULL,NULL,NULL)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, account['USERD'])
        ibm_db.bind_param(prepare_stmt, 2, VEHICLE_BG)
        ibm_db.execute(prepare_stmt)
        print('image_url sent to db2')
        # image_b64 = base64.b64encode(file.read()).decode('utf-8')
        # file1.save(file1.filename)
        return
render_template('vehicleremove.html',image_o=image_output,image_i=image_input,
user=user)
    return render_template('vehicleremove.html',user=user)
else:
    if request.method == "POST":
```

```
        user='Loggedin'
        file = request.files["filename"]
        print(file)
        # file1=file
        payload = {
            'image': ('image',file)
        }
        headers = {
            'X-RapidAPI-Key': RAPIDAPI_KEY,
            "X-RapidAPI-Host": "vehicle-background-removal.p.rapidapi.com"
        }
        response = requests.post(url, headers=headers,files=payload)
        output=response.json()
        print(output)
        image_input = output['data']['elements'][0]['origin_image_url']
        image_output = output['data']['elements'][0]['image_url']
        print(image_output)
        print(image_input)
        # image_b64 = base64.b64encode(file.read()).decode('utf-8')
        # file1.save(file1.filename)
        return
render_template('vehicleremove.html',image_o=image_output,image_i=image_input,
user=user)

        return render_template('vehicleremove.html')

@app.route('/beauty_img', methods=['POST','GET'])
def beauty_image():
    url = "https://ai-skin-beauty.p.rapidapi.com/face/editing/retouch-skin"
    sql = "SELECT * FROM USER1 WHERE USERD=" +str(session.get('USERD', -1))
    # sql = "SELECT * FROM USER1 WHERE USERD=" +str(session['USERD'])
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
        # if account['USERD']== session['USERD']:
        user='Loggedin'
        print('loggegin')
        if request.method == "POST":
            file = request.files["filename"]
            Retouch_degree=request.form['number']
            Whitening_degree=request.form['number1']
            print(file)
            # option = request.form.get("return_form")
            payload = {
                'image': ('image',file),
                'Retouch_degree': ('',Retouch_degree),
```

```
        'Whitening_degree': ('',Whitening_degree)
    }
    headers = {
        'X-RapidAPI-Key': RAPIDAPI_KEY,
        'X-RapidAPI-Host': "ai-skin-beauty.p.rapidapi.com"
    }
    response = requests.post(url, headers=headers,files=payload)
    output=response.json()
    print(output)
    image_output = output['data']['image_url']
    print(image_output)
    SKIN_BEAUTY =image_output
    insert_sql = "INSERT INTO IMAGE_URL VALUES
(?,NULL,NULL,NULL,?,NULL,NULL,NULL)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, account['USERD'])
    ibm_db.bind_param(prepare_stmt, 2, SKIN_BEAUTY )
    ibm_db.execute(prepare_stmt)
    print('image_url sent to db2')
    # image_b64 = base64.b64encode(file.read()).decode('utf-8')
    # file.save(file.filename)
    return
render_template('beauty_img.html',image_o=image_output,user=user)
    return render_template('beauty_img.html',user=user)
else:
    if request.method == "POST":
        file = request.files["filename"]
        Retouch_degree=request.form['number']
        Whitening_degree=request.form['number1']
        print(file)
        # option = request.form.get("return_form")
        payload = {
            'image': ('image',file),
            'Retouch_degree': ('',Retouch_degree),
            'Whitening_degree': ('',Whitening_degree)
        }
        headers = {
            'X-RapidAPI-Key': RAPIDAPI_KEY,
            'X-RapidAPI-Host': "ai-skin-beauty.p.rapidapi.com"
        }
        response = requests.post(url, headers=headers,files=payload)
        output=response.json()
        print(output)
        image_output = output['data']['image_url']
        print(image_output)
        # image_b64 = base64.b64encode(file.read()).decode('utf-8')
        # file.save(file.filename)
        return render_template('beauty_img.html',image_o=image_output)
```

```
return render_template('beauty_img.html')

@app.route('/cartoon_img', methods=['POST','GET'])
def cartoon_image():

    url = "https://cartoon-yourself.p.rapidapi.com/facebody/api/portrait-
animation/portrait-animation"

    sql = "SELECT * FROM USER1 WHERE USERD=" +str(session.get('USERD', -1))
    # sql = "SELECT * FROM USER1 WHERE USERD=" +str(session['USERD'])
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
        user='Loggedin'
        print('loggegin')
        if request.method == "POST":
            # user='Loggedin'
            file = request.files["filename"]
            print(file)
            option = request.form["return_form"]
            print(option)
            payload = {
                'image': (file),
                'type': (option)
            }
            headers = {
                'X-RapidAPI-Key': RAPIDAPI_KEY3,
                'X-RapidAPI-Host': "cartoon-yourself.p.rapidapi.com"
            }
            response = requests.post(url, headers=headers,files=payload)
            print(response)
            output=response.json()
            print(output)
            # try:
            #     output = response.json()
            # except json.decoder.JSONDecodeError as e:
            #     print(f"Failed to decode response: {e}")
            #     output = {}
            print(type(output))
            print(dir(output))
            image_output = output['data']['image_url']
            print(image_output)
            CARTOON_IMG=image_output
            insert_sql = "INSERT INTO IMAGE_URL VALUES
(?,NULL,NULL,?,NULL,NULL,NULL,NULL)"
```

```
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, account['USERD'])
        ibm_db.bind_param(prepare_stmt, 2, CARTOON_IMG)
        ibm_db.execute(prepare_stmt)
        print('image_url sent to db2')
        return render_template('cartoon_img.html',image_o=image_output,
user=user)
    return render_template('cartoon_img.html',user=user)
else:
    if request.method == "POST":
        user='Loggedin'
        file = request.files["filename"]
        print(file)
        option = request.form["return_form"]
        payload = {
            'image': ('image',file),
            'return_form': ('anime',option)
        }
        headers = {
            'X-RapidAPI-Key': RAPIDAPI_KEY3,
            'X-RapidAPI-Host': "cartoon-yourself.p.rapidapi.com"
        }
        response = requests.post(url, headers=headers,files=payload)
        output=response.json()
        print(output)
        image_output = output['data']['image_url']
        print(image_output)
        # image_b64 = base64.b64encode(file.read()).decode('utf-8')
        # file.save(file.filename)
        return render_template('cartoon_img.html',image_o=image_output,
user=user)
    return render_template('cartoon_img.html')

@app.route('/beauty_images')
def beauty_img_ai():
    user='login'
    # user='loggedin'
    sql = "SELECT * FROM IMAGE_URL WHERE USERD=" +str(session['USERD'])
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    row = []
    while True:
        data = ibm_db.fetch_assoc(stmt)
        if not data:
            break
        else:
            row.append(data)
    print('rows: ', row)
```

```
        return render_template("My_images.html", rows=row, user1=user)

@app.route('/remove_bg_images')
def remove_bg_images():
    user1='login'
    # user='loggedin'
    sql = "SELECT * FROM IMAGE_URL WHERE USERD=" +str(session['USERD'])
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    row = []
    while True:
        images_url = ibm_db.fetch_assoc(stmt)
        if not images_url:
            break
        else:
            row.append(images_url)
    print('rows: ', row)
    return render_template("My_images.html", row1=row, user1=user1)

@app.route('/vehicle_images')
def vehicle_img_ai():
    user1='login'
    # user='loggedin'
    sql = "SELECT * FROM IMAGE_URL WHERE USERD=" +str(session['USERD'])
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    row = []
    while True:
        data = ibm_db.fetch_assoc(stmt)
        if not data:
            break
        else:
            row.append(data)
    print('rows: ', row)
    return render_template("My_images.html", row2=row, user1=user1)

@app.route('/cartoon_images')
def cartoon_img_ai():
    user1='login'
    # user='loggedin'
    sql = "SELECT * FROM IMAGE_URL WHERE USERD=" +str(session['USERD'])
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    row = []
    while True:
        data = ibm_db.fetch_assoc(stmt)
        if not data:
            break
```



```
        else:
            row.append(data)
        print('rows: ', row)
        return render_template("My_images.html", row3=row, user1=user1)

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('USERID', None)
    return render_template('index.html')

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)
```

TEMPLATES

index.html:

```
<!DOCTYPE html>
<html lang="en">

  <head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;50
0;600;700;800;900&display=swap" rel="stylesheet">

    <title>Pixel Perfection: Transforming your photos with our cutting-edge
image editing platform</title>

    <link href="static/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">

    <link rel="stylesheet" href="static/assets/css/fontawesome.css">
    <link rel="stylesheet" href="static/assets/css/templatemo-574-mexant.css">
    <link rel="stylesheet" href="static/assets/css/owl.css">
    <link rel="stylesheet" href="static/assets/css/animate.css">
```

```
<link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-  
bundle.min.css">  
</head>  
  
<body>  
  
<header class="header-area header-sticky">  
  <div class="container">  
    <div class="row">  
      <div class="col-12">  
        <nav class="main-nav">  
          <ul class="nav">  
            <li class="scroll-to-section"><a href="#top"  
class="active">Home</a></li>  
            <li class="scroll-to-section"><a  
href="#about">About</a></li>  
  
            {% if user == 'Loggedin' %}  
            <li><a href="/logout">logout</a></li>  
            {% else %}  
            <li><a href="/login">login</a></li>  
            {% endif %}  
          </ul>  
          <a class='menu-trigger'>  
            <span>Menu</span>  
          </a>  
        </nav>  
      </div>  
    </div>  
  </div>  
</header>  
  
<div class="swiper-container" id="top">  
  <div class="swiper-wrapper">  
    <div class="swiper-slide">  
      <div class="slide-inner" style="background-  
image:url(https://images.pexels.com/photos/2085998/pexels-photo-  
2085998.jpeg)">  
        <div class="container">  
          <div class="row">  
            <div class="col-lg-8">  
              <div class="header-text">  
                <h2><em>Pixel</em> Perfection</h2>>  
                <h4> Transforming Your Photos Of Our Cutting-Edge Image  
<em>Editing Platform </em></h4>  
  
              <div class="div-dec"></div>
```

`<p>Pixel Perfection is an innovative image editing platform that allows users to transform their photos with ease and precision. Our cutting-edge software provides a wide range of tools and features that enable users to edit their images to achieve pixel-perfect results. Whether you're a professional photographer, graphic designer, or just someone who loves to take photos, Pixel Perfection is the perfect tool for enhancing your images. With its intuitive user interface and powerful editing tools, you can easily adjust or remove your image backgrounds, car image backgrounds, Cartoon your face & Face beauty, and more to create stunning images that are sure to impress.</p>`

```
    </div>
  </div>
</div>
</div>
</div>
</div>
```

```
  </div>
</div>
<div class="swiper-button-next swiper-button-white"></div>
<div class="swiper-button-prev swiper-button-white"></div>
</div>
```

```
<section class="simple-cta">
  <div class="container">
    <div class="row">
      <div class="col-lg-5">
        <h4>Please <em>Login</em> to use our <strong>Editing
Platform</strong> </h4>
```

```
      </div>
      <div class="col-lg-7">
        <div class="buttons">
```

```
          <div class="orange-button">
            <a href="/login">login</a>
          </div>
```

```
        </div>
      </div>
    </div>
  </div>
</section>
```

```
<section class="about-us" id="about">
  <div class="container">
    <div class="row">
```

```
<div class="col-lg-6 offset-lg-3">
  <div class="section-heading">
    <h6>About Us</h6>
    <h4>Know Us Better</h4>
  </div>
</div>
<div class="col-lg-4">
  <div class="right-content">
    <h4>Team Mates</h4>
    <p>Team Lead - BHARATH VISHNU C J<br>Team Member 01 : BARANI
DHARAN V M<br>Team member 02 - BARATH B<br>Team member 03 - DILIP S</p>

  </div>
</div>
<div class="col-lg-8">
  <div class="naccs">
    <div class="tabs">
      <div class="row">
        <div class="col-lg-12">
          <div class="menu">
            <div class="gradient-border"><span>Cartoon your
images</span></div>
            <div class="active gradient-border"><span>Remove
Background</span></div>
            <div class="gradient-border"><span>AI Image
Generator</span></div>
          </div>
        </div>
        <div class="col-lg-12">
          <ul class="nacc">
            <li>
              <div>
                <div class="main-list">
                  

                </div>
              </div>
            </li>
            <li class="active">
              <div>
                <div class="main-list">
                  

                </div>
              </div>
            </li>
            <li>
```

34

```
mousewheelControl: true,
keyboardControl: true,
navigation: {
  nextEl: ".swiper-button-next",
  prevEl: ".swiper-button-prev"
},
on: {
  progress: function() {
    var swiper = this;
    for (var i = 0; i < swiper.slides.length; i++) {
      var slideProgress = swiper.slides[i].progress;
      var innerOffset = swiper.width * interleaveOffset;
      var innerTranslate = slideProgress * innerOffset;
      swiper.slides[i].querySelector(".slide-inner").style.transform =
        "translate3d(" + innerTranslate + "px, 0, 0)";
    }
  },
  touchStart: function() {
    var swiper = this;
    for (var i = 0; i < swiper.slides.length; i++) {
      swiper.slides[i].style.transition = "";
    }
  },
  setTransition: function(speed) {
    var swiper = this;
    for (var i = 0; i < swiper.slides.length; i++) {
      swiper.slides[i].style.transition = speed + "ms";
      swiper.slides[i].querySelector(".slide-inner").style.transition
=
      speed + "ms";
    }
  }
}
};

var swiper = new Swiper(".swiper-container", swiperOptions);
</script>
</body>
</html>
```

login.html:

```
<!DOCTYPE html>
<html lang="en">

<head>

<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">
<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;50
0;600;700;800;900&display=swap" rel="stylesheet">

<title>Pixel Perfection: Transforming your photos with our cutting-edge
image editing platform</title>

<link href="static/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">

<link rel="stylesheet" href="static/assets/css/fontawesome.css">

<link rel="stylesheet" href="static/assets/css/owl.css">
<link rel="stylesheet" href="static/assets/css/animate.css">
<link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-
bundle.min.css">
<link rel="stylesheet" href="static/assets/css/templatemo-575-mexant.css">
</head>
<body>

<header class="header-area header-sticky">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <nav class="main-nav">

          <ul class="nav">
            <li class="scroll-to-section"><a href="/"
class="active">Home</a></li>
            <li class="scroll-to-section"><a
href="#about">About</a></li>

            <li><a href="/register">Register</a></li>
          </ul>
          <a class='menu-trigger'>
            <span>Menu</span>
          </a>

        </nav>
      </div>
    </div>
  </div>
</header>
```

```
<section id="register" class="calculator" style="cursor: pointer;">
  <div class="container">
    <div class="row">
      <div class="col-lg-7">
        <div class="left-image">

        </div>
      </div>
      <div class="col-lg-5">
        <div class="section-heading">

        <h4>Login here</h4>
        </div>
        <form id="calculate" action="/login" method="POST" style="cursor:
context-menu">
          <div class="row">
            <div class="col-lg-12">
              <fieldset>
                <label for="msg">{{msg}}</label>
              </fieldset>
            </div>
            <div class="col-lg-12">
              <fieldset>
                <label for="email">Enter Your Email</label>
                <input type="text" name="email" id="email" pattern="^[^
@]*@[^ @]*" placeholder="youremail@gmail.com" required="">
              </fieldset>
            </div>
            <div class="col-lg-12">
              <fieldset>
                <label for="password">Enter Your Password</label>
                <input type="passowrd" name="password" id="subject"
placeholder="">
              </fieldset>
            </div>
            <div class="col-lg-12">
              </fieldset>
            </div>
            <div class="col-lg-6">

            <button type="submit" id="form-submit" class="orange-
button">Submit Now</button>

            </div><br><br>
            <div class="col-lg-12">
              <fieldset>
```



```
        <label for="login">You don't Have an account ?<a
href="/register">Click</a></label>

        </fieldset>
    </div>
</div>
<div class="msg">{{ msg }}</div>
</form>
</div>
</div>
</div>
</section>

<script src="static/vendor/jquery/jquery.min.js"></script>
<script src="static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="static/assets/js/isotope.min.js"></script>
<script src="static/assets/js/owl-carousel.js"></script>

<script src="static/assets/js/tabs.js"></script>
<script src="static/assets/js/swiper.js"></script>
<script src="static/assets/js/custom.js"></script>
<script>
    var interleaveOffset = 0.5;

    var swiperOptions = {
        loop: true,
        speed: 1000,
        grabCursor: true,
        watchSlidesProgress: true,
        mousewheelControl: true,
        keyboardControl: true,
        navigation: {
            nextEl: ".swiper-button-next",
            prevEl: ".swiper-button-prev"
        },
        on: {
            progress: function() {
                var swiper = this;
                for (var i = 0; i < swiper.slides.length; i++) {
                    var slideProgress = swiper.slides[i].progress;
                    var innerOffset = swiper.width * interleaveOffset;
                    var innerTranslate = slideProgress * innerOffset;
                    swiper.slides[i].querySelector(".slide-inner").style.transform =
                        "translate3d(" + innerTranslate + "px, 0, 0)";
                }
            }
        }
    },
```

```
touchStart: function() {
    var swiper = this;
    for (var i = 0; i < swiper.slides.length; i++) {
        swiper.slides[i].style.transition = "";
    }
},
setTransition: function(speed) {
    var swiper = this;
    for (var i = 0; i < swiper.slides.length; i++) {
        swiper.slides[i].style.transition = speed + "ms";
        swiper.slides[i].querySelector(".slide-inner").style.transition
=
        speed + "ms";
    }
}
};

var swiper = new Swiper(".swiper-container", swiperOptions);
</script>
</body>

</html>
```

Register.html:

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;50
0;600;700;800;900&display=swap" rel="stylesheet">

    <title>Pixel Perfection: Transforming your photos with our cutting-edge
image editing platform</title>

    <link href="static/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">
```

```
<link rel="stylesheet" href="static/assets/css/fontawesome.css">
<link rel="stylesheet" href="static/assets/css/templatemo-575-mexant.css">
<link rel="stylesheet" href="static/assets/css/owl.css">
<link rel="stylesheet" href="static/assets/css/animate.css">
<link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-
bundle.min.css">
</head>

<body>

<header class="header-area header-sticky">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <nav class="main-nav">

          <ul class="nav">
            <li class="scroll-to-section"><a href="/"
class="active">Home</a></li>
            <li class="scroll-to-section"><a
href="#about">About</a></li>

            <li><a href="/login">Login</a></li>
          </ul>
          <a class='menu-trigger'>
            <span>Menu</span>
          </a>

        </nav>
      </div>
    </div>
  </div>
</header>

<section id="register" class="calculator" style="cursor: pointer;">
  <div class="container">
    <div class="row">
      <div class="col-lg-7">
        <div class="left-image">
          
        </div>
      </div>
      <div class="col-lg-5">
        <div class="section-heading">

          <h4>Register here</h4>
        </div>
      </div>
    </div>
  </div>
</section>
```

```
<form id="calculate" action="/register" method="POST" style="cursor:
context-menu">
  <div class="row">
    <div class="col-lg-12">
      <fieldset>
        <label for="msg">{{msg}}</label>
      </fieldset>
    </div>
    <div class="col-lg-12">
      <fieldset>
        <label for="email">Enter Your Name</label>
        <input type="text" name="name" id="name" required="">
      </fieldset>
    </div>
    <div class="col-lg-12">
      <fieldset>
        <label for="email">Enter Your Email</label>
        <input type="text" name="email" id="email" pattern="[^\
@]*@[^\ @]*" placeholder="youremail@gmail.com" required="">
      </fieldset>
    </div>
    <div class="col-lg-12">
      <fieldset>
        <label for="password">Enter Your Password</label>
        <input type="passowrd" name="password" id="subject"
placeholder="">
      </fieldset>
    </div>
    <div class="col-lg-12">
      </fieldset>
    </div>
    <div class="col-lg-6">
      <button type="submit" id="form-submit" class="orange-
button">Submit Now</button>

    </div><br><br>
    <div class="col-lg-12">
      <fieldset>
        <label for="login">Already have an account?<a
href="/login">Click</a></label>

      </fieldset>
    </div>
  </div>
</form>
</div>
```

```
</div>
</div>
</section>

<script src="static/vendor/jquery/jquery.min.js"></script>
<script src="static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="static/assets/js/isotope.min.js"></script>
<script src="static/assets/js/owl-carousel.js"></script>

<script src="static/assets/js/tabs.js"></script>
<script src="static/assets/js/swiper.js"></script>
<script src="static/assets/js/custom.js"></script>
<script>
    var interleaveOffset = 0.5;

    var swiperOptions = {
        loop: true,
        speed: 1000,
        grabCursor: true,
        watchSlidesProgress: true,
        mousewheelControl: true,
        keyboardControl: true,
        navigation: {
            nextEl: ".swiper-button-next",
            prevEl: ".swiper-button-prev"
        },
        on: {
            progress: function() {
                var swiper = this;
                for (var i = 0; i < swiper.slides.length; i++) {
                    var slideProgress = swiper.slides[i].progress;
                    var innerOffset = swiper.width * interleaveOffset;
                    var innerTranslate = slideProgress * innerOffset;
                    swiper.slides[i].querySelector(".slide-inner").style.transform =
                        "translate3d(" + innerTranslate + "px, 0, 0)";
                }
            },
            touchStart: function() {
                var swiper = this;
                for (var i = 0; i < swiper.slides.length; i++) {
                    swiper.slides[i].style.transition = "";
                }
            },
            setTransition: function(speed) {
                var swiper = this;
                for (var i = 0; i < swiper.slides.length; i++) {
                    swiper.slides[i].style.transition = speed + "ms";
                }
            }
        }
    };
    new Swiper(".swiper", swiperOptions);
</script>
```

```
        swiper.slides[i].querySelector(".slide-inner").style.transition
=
        speed + "ms";
    }
}
};

var swiper = new Swiper(".swiper-container", swiperOptions);
</script>
</body>
</html>
```

removebg.html:

```
<!DOCTYPE html>
<html lang="en">
<div class="my-class">
  <head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;50
0;600;700;800;900&display=swap" rel="stylesheet">

    <title>Pixel Perfection: Transforming your photos with our cutting-edge
image editing platform</title>

    <link href="static/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">

    <link rel="stylesheet" href="static/assets/css/fontawesome.css">
    <link rel="stylesheet" href="static/assets/css/templatemo-574-mexant.css">
    <link rel="stylesheet" href="static/assets/css/owl.css">
    <link rel="stylesheet" href="static/assets/css/animate.css">
    <link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-
bundle.min.css">

  </head>

<body>
```

```
<header class="header-area header-fixed background-header">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <nav class="main-nav">

          <ul class="nav">
            <li class="scroll-to-section"><a href='/',
class="active">Home</a></li>
            {% if user == 'Loggedin' %}
            <li class="scroll-to-section"><a
href="/remove_bg_images">My Images</a></li>
            {% endif %}

            <li class="scroll-to-section"><a
href="#about">About</a></li>
            <li class="has-sub">
              <a href="javascript:void(0)">IMAGE AI</a>
              <ul class="sub-menu">
                <li><a href="/removebg">Remove
Background</a></li>
                <li><a href="/vehicleremove">Vehicle
Background</a></li>
                <li><a href="/cartoon_img">Cartoon
yourself</a></li>
                <li><a href="/beauty_img">AI Skin
Beauty</a></li>

              </ul>
            </li>
            {% if user == 'Loggedin' %}
            <li><a href="/logout">logout</a></li>

            {% else %}
            <li><a href="/login">login</a></li>
            {% endif %}
          </ul>
          <a class='menu-trigger'>
            <span>Menu</span>
          </a>

        </nav>
      </div>
    </div>
  </div>
</header>
```

```
<section class="main-services">
  <div class="container">

    <div class="row">
      <div class="col-lg-12">
        <div class="service-item">
          <div class="row">
            <div class="col-lg-6">

              <div class="left-image">
                <br><br>
                
              </div>
            </div>
            <div class="col-lg-6 align-self-center">
              <div class="right-text-content">
                <i class="fa fa-upload"></i>
                <h4>Upload Image</h4>
                <form action="/removebg" method="POST"
enctype="multipart/form-data">
                  <input type="file" id="myFile" name="filename"><br><br>
                  <p>NOTE:- Image format: JPEG, JPG, PNG(8-bit, 16-bit, 64-
bit PNG not supported), BMP, WEBP. <br>
                    Image size: no more than 3 MB.<br>
                    Image resolution: less than 1280 x 1280 pixels.</p><br>
                  <label for="return_form">Choose:</label>
                  <select id="remove_bg_img" name='return_form'>
                    <option value="mask">mask</option>
                    <option value="whiteBK">whiteBK</option>
                  </select><br><br>
                  <input type="submit">
                </form><br>
                <p>NOTE:- <br>
                  mask: Returns a single channel mask.<br>
                  whiteBK: Return to white background image.</p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>

<section class="simple-cta">
  <div class="container">
```


46

```
        nextEl: ".swiper-button-next",
        prevEl: ".swiper-button-prev"
    },
    on: {
        progress: function() {
            var swiper = this;
            for (var i = 0; i < swiper.slides.length; i++) {
                var slideProgress = swiper.slides[i].progress;
                var innerOffset = swiper.width * interleaveOffset;
                var innerTranslate = slideProgress * innerOffset;
                swiper.slides[i].querySelector(".slide-inner").style.transform =
                    "translate3d(" + innerTranslate + "px, 0, 0)";
            }
        },
        touchStart: function() {
            var swiper = this;
            for (var i = 0; i < swiper.slides.length; i++) {
                swiper.slides[i].style.transition = "";
            }
        },
        setTransition: function(speed) {
            var swiper = this;
            for (var i = 0; i < swiper.slides.length; i++) {
                swiper.slides[i].style.transition = speed + "ms";
                swiper.slides[i].querySelector(".slide-inner").style.transition
=
                speed + "ms";
            }
        }
    }
};

var swiper = new Swiper(".swiper-container", swiperOptions);
</script>
</body>
</div>
</html>
```

beauty_img.html:

```
<!DOCTYPE html>
<html lang="en">
<div class="my-class">

    <head>

        <meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">
<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;50
0;600;700;800;900&display=swap" rel="stylesheet">

<title>Pixel Perfection: Transforming your photos with our cutting-edge
image editing platform</title>

<link href="static/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">

<link rel="stylesheet" href="static/assets/css/fontawesome.css">
<link rel="stylesheet" href="static/assets/css/templatemo-574-mexant.css">
<link rel="stylesheet" href="static/assets/css/owl.css">
<link rel="stylesheet" href="static/assets/css/animate.css">
<link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-
bundle.min.css">

</head>

<body>

<header class="header-area header-fixed background-header">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <nav class="main-nav">

          <ul class="nav">
            <li class="scroll-to-section"><a href='/',
class="active">Home</a></li>
            {% if user == 'LoggedIn' %}
            <li class="scroll-to-section"><a
href="/beauty_images">My Images</a></li>
            {% endif %}
            <li class="scroll-to-section"><a
href="#about">About</a></li>
            <li class="has-sub">
              <a href="javascript:void(0)">IMAGE AI</a>
              <ul class="sub-menu">
                <li><a href="/removebg">Remove
Background</a></li>
                <li><a href="/vehicleremove">Vehicle
Background</a></li>
```

```
                                <li><a href="/cartoon_img">Cartoon
yourself</a></li>
                                <li><a href="/beauty_img">AI Skin
Beauty</a></li>

                                </ul>
                            </li>

                                {% if user == 'Loggedin' %}
                                <li><a href="/logout">logout</a></li>
                                {% else %}
                                <li><a href="/login">login</a></li>
                                {% endif %}
                            </ul>
                            <a class='menu-trigger'>
                                <span>Menu</span>
                            </a>

                        </nav>
                    </div>
                </div>
            </div>
        </header>

        <section class="main-services">
            <div class="container">
                <div class="row">
                    <div class="col-lg-12">
                        <div class="service-item">
                            <div class="row">
                                <div class="col-lg-6">
                                    <div class="left-image">
                                        <br><br>

                                    </div>
                                </div>
                                <div class="col-lg-6 align-self-center">
                                    <div class="right-text-content">
                                        <i class="fa fa-upload"></i>
                                        <h4>Upload Image</h4>
                                        <form action="/beauty_img" method="POST"
enctype="multipart/form-data">
                                            <input type="file" id="myFile" name="filename"><br><br>
                                            <p>NOTE:- Image format: Only JPG,JPEG,PNG format
Supported. <br>
                                            Image size: no more than 3 MB.<br>
                                            Image resolution: less than 1280 x 1280 pixels.</p><br>
```

```

                <p>Retouch_degree:-<input type="number" id="number"
name="number"  min="0" max="1.5" step="0.1"></p><br><br>
                <p>Whitening_degree:-<input type="number" id="number"
name="number1"  min="0" max="1.5" step="0.1"></p><br><br>

                <input type="submit">
            </form><br>

        </div>
    </div>
</div>
</div>
</div>
</div>

</div>
</div>
</section>

<section class="simple-cta">
    <div class="container">
        <div class="row">
            <div class="col-lg-5">
                <h4>For <em>Help</em> and <strong>Support</strong> Contact us !</h4>
            </div>
            <div class="col-lg-7">
                <div class="buttons">
                    <div class="green-button">
                        <a href="#">Discover More</a>
                    </div>
                    <div class="orange-button">
                        <a href="#">Contact Us</a>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>

<footer>
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <p>Copyright @The-Achievers All Rights Reserved.

            </div>
        </div>
    </div>
</div>
```

```
</div>
</footer>

<script src="static/vendor/jquery/jquery.min.js"></script>
<script src="static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="static/assets/js/isotope.min.js"></script>
<script src="static/assets/js/owl-carousel.js"></script>

<script src="static/assets/js/tabs.js"></script>
<script src="static/assets/js/swiper.js"></script>
<script src="static/assets/js/custom.js"></script>
<script>
    var interleaveOffset = 0.5;

    var swiperOptions = {
        loop: true,
        speed: 1000,
        grabCursor: true,
        watchSlidesProgress: true,
        mousewheelControl: true,
        keyboardControl: true,
        navigation: {
            nextEl: ".swiper-button-next",
            prevEl: ".swiper-button-prev"
        },
        on: {
            progress: function() {
                var swiper = this;
                for (var i = 0; i < swiper.slides.length; i++) {
                    var slideProgress = swiper.slides[i].progress;
                    var innerOffset = swiper.width * interleaveOffset;
                    var innerTranslate = slideProgress * innerOffset;
                    swiper.slides[i].querySelector(".slide-inner").style.transform =
                        "translate3d(" + innerTranslate + "px, 0, 0)";
                }
            },
            touchStart: function() {
                var swiper = this;
                for (var i = 0; i < swiper.slides.length; i++) {
                    swiper.slides[i].style.transition = "";
                }
            },
            setTransition: function(speed) {
                var swiper = this;
                for (var i = 0; i < swiper.slides.length; i++) {
                    swiper.slides[i].style.transition = speed + "ms";
                }
            }
        }
    };
    new Swiper(".swiper", swiperOptions);
</script>
```

```
        swiper.slides[i].querySelector(".slide-inner").style.transition
=
        speed + "ms";
    }
}
};

var swiper = new Swiper(".swiper-container", swiperOptions);
</script>
</body>
</div>
</html>
```

cartoon_img.html:

```
<!DOCTYPE html>
<html lang="en">
<div class="my-class">
  <head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;50
0;600;700;800;900&display=swap" rel="stylesheet">

    <title>Pixel Perfection: Transforming your photos with our cutting-edge
image editing platform</title>

    <link href="static/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">

    <link rel="stylesheet" href="static/assets/css/fontawesome.css">
    <link rel="stylesheet" href="static/assets/css/templatemo-574-mexant.css">
    <link rel="stylesheet" href="static/assets/css/owl.css">
    <link rel="stylesheet" href="static/assets/css/animate.css">
    <link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-
bundle.min.css">

  </head>

<body>
```

```
<header class="header-area header-fixed background-header">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <nav class="main-nav">

          <ul class="nav">
            <li class="scroll-to-section"><a href='/',
class="active">Home</a></li>

            {% if user == 'Loggedin' %}
            <li class="scroll-to-section"><a
href="/cartoon_images">My Images</a></li>
            {% endif %}
            <li class="scroll-to-section"><a
href="#about">About</a></li>
            <li class="has-sub">
              <a href="javascript:void(0)">IMAGE AI</a>
              <ul class="sub-menu">
                <li><a href="/removebg">Remove
Background</a></li>
                <li><a href="/vehicleremove">Vehicl
Background</a></li>
                <li><a href="/cartoon_img">Cartoon
yourself</a></li>
                <li><a href="/beauty_img">AI Skin
Beauty</a></li>

              </ul>
            </li>

            {% if user == 'Loggedin' %}
            <li><a href="/logout">logout</a></li>
            {% else %}
            <li><a href="/login">login</a></li>
            {% endif %}
          </ul>
          <a class='menu-trigger'>
            <span>Menu</span>
          </a>

        </nav>
      </div>
    </div>
  </div>
</header>
```



```
<section class="main-services">
  <div class="container">
    <div class="row">
      <div class="col-lg-12">
        <div class="service-item">
          <div class="row">
            <div class="col-lg-6">
              <div class="left-image">
                <br><br>
                
              </div>
            </div>
            <div class="col-lg-6 align-self-center">
              <div class="right-text-content">
                <i class="fa fa-upload"></i>
                <h4>Upload Image</h4>
                <form action="/cartoon_img" method="POST"
enctype="multipart/form-data">
                  <input type="file" id="myFile" name="filename"><br><br>
                  <p>NOTE:- Image to be processed.
                    Image format: JPG, JPEG, PNG, BMP.
                    Image size: no more than 3 MB.<br>
                    Image resolution: less than 1280 x 1280
pixels.</p><br>
                  <label for="return_form">Choose:</label>
                  <select id="image_type" name='return_form'>
                    <option value="anime">anime</option>
                    <option value="pixar_plus">pixar plus</option>
                    <option value="3d_cartoon">3d cartoon</option>
                    <option value="angel">angel</option>

                    <option value="handdrawn">handdrawn</option>
                    <option value="demon">demon</option>
                  </select><br><br>
                  <input type="submit">
                </form><br>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

<section class="simple-cta">
  <div class="container">
```

```
<div class="row">
  <div class="col-lg-5">
    <h4>For <em>Help</em> and <strong>Support</strong> Contact us !</h4>
  </div>
  <div class="col-lg-7">
    <div class="buttons">
      <div class="green-button">
        <a href="#">Discover More</a>
      </div>
      <div class="orange-button">
        <a href="#">Contact Us</a>
      </div>
    </div>
  </div>
</div>
</div>
</section>

<footer>
  <div class="container">
    <div class="row">
      <div class="col-lg-12">
        <p>Copyright @The-Achievers All Rights Reserved.

      </div>
    </div>
  </div>
</footer>

<script src="static/vendor/jquery/jquery.min.js"></script>
<script src="static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="static/assets/js/isotope.min.js"></script>
<script src="static/assets/js/owl-carousel.js"></script>

<script src="static/assets/js/tabs.js"></script>
<script src="static/assets/js/swiper.js"></script>
<script src="static/assets/js/custom.js"></script>
<script>
  var interleavedOffset = 0.5;

  var swiperOptions = {
    loop: true,
    speed: 1000,
    grabCursor: true,
    watchSlidesProgress: true,
```

```
mousewheelControl: true,  
keyboardControl: true,  
navigation: {  
  nextEl: ".swiper-button-next",  
  prevEl: ".swiper-button-prev"  
},  
on: {  
  progress: function() {  
    var swiper = this;  
    for (var i = 0; i < swiper.slides.length; i++) {  
      var slideProgress = swiper.slides[i].progress;  
      var innerOffset = swiper.width * interleaveOffset;  
      var innerTranslate = slideProgress * innerOffset;  
      swiper.slides[i].querySelector(".slide-inner").style.transform =  
        "translate3d(" + innerTranslate + "px, 0, 0)";  
    }  
  },  
  touchStart: function() {  
    var swiper = this;  
    for (var i = 0; i < swiper.slides.length; i++) {  
      swiper.slides[i].style.transition = "";  
    }  
  },  
  setTransition: function(speed) {  
    var swiper = this;  
    for (var i = 0; i < swiper.slides.length; i++) {  
      swiper.slides[i].style.transition = speed + "ms";  
      swiper.slides[i].querySelector(".slide-inner").style.transition  
=  
        speed + "ms";  
    }  
  }  
}  
};  
  
var swiper = new Swiper(".swiper-container", swiperOptions);  
</script>  
</body>  
</div>  
</html>
```

KIOT

[illegible][illegible]

10.2 GITHUB & PROJECT VIDEO DEMO LINK

GITHUB LINK:

<https://github.com/naanmudhalvan-SI/IBM--13556-1682326385>

PROJECT VIDEO DEMO LINK:

<https://www.youtube.com/watch?v=NHZjcwoqmu4>

DEPLOYMENT LINK: <http://159.122.181.148:30472/>
