



# Generative Prompt Image Classification on Promptails Website

Version	: 4.0.0
Page	: 49 pages
Status	: Final
Confidentiality	: Public

**JUNE 30, 2024**

INFORMATICS ENGINEERING BATAM STATE  
POLYTECHNIC

Compiled By: **PBL IF 23-2-16**

# Identitas Proyek

ID Number	:	IF 23-2-16
Project Proponent	:	Agung Riyadi, S.Si. M.Kom
Project Manager	:	Agung Riyadi, S.Si. M.Kom
Co Manpro	:	-
Project Title	:	Generative Prompt Image Classification on Promptails Website
Output	:	1. Classification Model 2. Dataset Processed 3. System Implementation
Clients/Customers	:	-
Steering Committee (Lecturer & Laboratory of PBL course)	:	1. Rina Yulius,S.PD., M.Eng [Industrial Software Project – IF419] 2. Mir'atul Khusna Mufida,S.ST., M.Sc [Artificial Intelligence – IF420] 3. Agus Fatulloh,S.T., M.T & Noper Ardi, S.Pd., M.Eng [Software Installation and Maintenance – IF421] 4. Adhiguna,S.Kom. & Muhammad Idris, S.Tr. [Software Testing – IF422] 5. Suwarno ,S.S., M.Pd & Siti Noor Chayati, S.T., M.Sc [English for Business – IF424] 6. Ir.Farouki Dinda Rassarandi,S.T., M.Eng. & Jhon Hericson Purba,S.pd.,M.Pd [Indonesian Language Education – PK4IF]
Student Team Member	:	1 3312211055 – Samuel Parsaoran Tambunan 2 3312211004 – Sechan Faradila Sahab 3 3312211006 – Muhamad Rafiansyah 4 3312211008 – Putra Ramadhan 5 3312211073 – Alpaber Pardomuan Tambunan

---

# Table of Contents

Project Identity	2
Table of Contents	3
Document History	4
List of Image	5
Table List	6
List of Attachments	7
Project Work History	8
System Specifications	13
A. General Description	13
B. User Stories	14
C. System Modeling	16
D. Database Design	18
E. Interface Design	21
Implementation Results	25
A. Application Implementation	25
B. Database Implementation	34
C. Application Testing and Deployment	36
Conclusion	45
Summary	45
Lesson Learned	46
Bibliography	47
APPENDIX	49

---

# Document History

Table 1. Document History

Version	Date	Author	Description
1.0.0	10/03/2024	Samuel Parsaoran Tambunan	Writing Project Identity, Table of Contents, List of Figures, List of Tables, List of Appendices, Project Work History, System Specifications.
2.0.0	24/03/2024	Samuel Parsaoran Tambunan	Table of Contents, List of Figures, List of Tables, List of Appendices, Project History, System Specifications.
3.0.0	11/06/2024	Samuel Parsaoran Tambunan	Table of Contents, List of Figures, List of Tables, List of Appendices, Project History, System Specifications.
4.0.0 (Final)	30/06/2024	Samuel Parsaoran Tambunan	Table of Contents, List of Figures, List of Tables, List of Appendices, Project History, System Specifications, Implementation Results, Closing, Bibliography, Appendices.

---

# List of Image

Image 1, 2 ..... 11

Image 3, 4, 5 ..... 12

Image 6 ..... 13

Image 7 ..... 17

Image 8 ..... 18

Image 9 ..... 21

Image 10, 11 ..... 22

Image 12 ..... 23

Image 13, 14 ..... 24

Image 15 ..... 25

Image 16, 17 ..... 26

Image 18, 19 ..... 27

Image 20 ..... 28

Image 21, 22 ..... 29

Image 23, 24 ..... 30

Image 25, 26 ..... 31

Image 27 ..... 34

Image 28, 29 ..... 35

Image 30, 31 ..... 41

Image 32 ..... 42

---

# Table List

Table 1 ..... 4

Table 2 ..... 8

Table 3, 4 ..... 9

Table 5 ..... 10

Table 6, 7 ..... 14

Table 8 ..... 15

Table 9 ..... 36

Table 10 ..... 43

Table 11 ..... 44

---

# List of Attachments

Lampiran 1 .....	48
------------------	----

## Mandatory attachments:

- a. Links to external documents are:
  - i. Full version logbook
  - ii. Application download link
  - iii. Presentation Videos
  - iv. Demo video (on completion)
  - v. Poster (on completion)
- b. Application Testing Documentation/photos
- c. Minutes of application handover to the proposer
- d. IPR proposal document (draft)
- e. Documentation of PBL work and group discussions and manpro at least 5 photos for 1 semester

---

# Project Work History

## Output Work History

Table 2. Output Work History

Task	Completion Date	Outputs produced
Planning	5 February 2024 – 26 February 2024	Project Implementation Plan, Project Plan, Report Version 1.0.0, PPT Presentation Version 1.0.0
Analysis	27 February 2024 – 24 March 2024	Report Version 2.0.0, Presentation .PPT Version 2.0.0
Design	27 February 2024 – 24 March 2024	Desain UI/UX, Wire Frame, ER Diagram, Diagram Alir, Use Case, Class Diagram.
Implementation	6 May 2024 – 22 June 2024	Data preprocessing, pattern recognition, model evaluation, model implementation, prediction system, classification system, CSV file upload system, web-based application.
Testing	23 June 2024 – 30 June 2024	Testing and Validation Results, Test Report.



## Application Version History

Table 3. Application Version History

Version	Release Date	Version Details
<b>1.0.0</b>	<b>10 Maret 2024</b>	<b>Planning Project Display.</b>
<b>2.0.0</b>	<b>24 March 2024</b>	<b>Initial prototyping of the app, implementation of initial UI/UX.</b>
<b>3.0.0</b>	<b>11 June 2024</b>	<b>Implementation of text classification model, integration of model with user interface.</b>
<b>4.0.0 (Akhir)</b>	<b>30 June 2024</b>	<b>Final application testing and validation, bug fixes, performance optimisation, application finalisation</b>

## Member Contributions

Table 4. Group member Contributions

Member Name	Contribution	Outputs produced
<b>Samuel Parsaoran Tambunan</b>	<ul style="list-style-type: none"> <li>• Scheduling</li> <li>• Project Implementation Plan Creation</li> <li>• Project Documentation</li> <li>• Front End Programming</li> <li>• Back End Programming</li> <li>• Application Testing</li> <li>• Application Training</li> </ul>	<ul style="list-style-type: none"> <li>• Project Implementation Plan Report</li> <li>• Project Creation Report</li> <li>• Application Testing Report</li> <li>• Logbook</li> <li>• Presentation Report</li> <li>• Presentation Video</li> <li>• Application Page View</li> <li>• SQLite Database</li> </ul>
<b>Sechan Faradila Sahab</b>	<ul style="list-style-type: none"> <li>• Usecase Diagram</li> <li>• Usecase Scenario</li> <li>• Functional &amp; non functional</li> <li>• Dataset</li> </ul>	<ul style="list-style-type: none"> <li>• Diagram Usecase</li> <li>• Usecase Scenario</li> <li>• Functional &amp; Non functional Documents</li> </ul>

		<ul style="list-style-type: none"> <li>• Data Cleaning</li> </ul>
<b>Muhamad Rafiansyah</b>	<ul style="list-style-type: none"> <li>• Creating UI/UX Design</li> <li>• Wireframe Creation</li> <li>• Back End Programming</li> <li>• Application Programming</li> <li>• Application Training</li> <li>• Application Testing</li> </ul>	<ul style="list-style-type: none"> <li>• Wireframe View</li> <li>• Integration of Artificial Intelligence System into Front End</li> <li>• Integration of Prediction System into Front End</li> <li>• Modelling</li> <li>• Application Manual Book</li> </ul>
<b>Putra Ramadhan</b>	<ul style="list-style-type: none"> <li>• Project Planning</li> <li>• Application Mock up</li> </ul>	<ul style="list-style-type: none"> <li>• Project Planning Report</li> <li>• Mock Up Display</li> </ul>
<b>Alpaber Pardomuan Tambunan</b>	<ul style="list-style-type: none"> <li>• Logbook</li> </ul>	<ul style="list-style-type: none"> <li>• Logbook</li> </ul>

### Communication History with Client/Proposer

Table 5. Communication with Clients / Project Proponent

Date	Client Name	Media	Discussion
<b>12/02/2024</b>	<b>Mr. Agung Riyadi, S.Si. M.Kom</b>	<b>Zoom Meeting</b>	<b>Discussion of initial application specifications</b>
<b>28/02/2024</b>	<b>Mr. Agung Riyadi, S.Si. M.Kom</b>	<b>Zoom Meeting</b>	<b>Discussion on making a Project Implementation Plan</b>
<b>04/03/2024</b>	<b>Mr. Agung Riyadi, S.Si. M.Kom</b>	<b>Zoom Meeting</b>	<b>Project Planning Discussion</b>
<b>06/03/2024</b>	<b>Mr. Agung Riyadi, S.Si. M.Kom</b>	<b>Zoom Meeting</b>	<b>Confirmation of application specifications</b>
<b>13/03/2024</b>	<b>Mr. Agung Riyadi, S.Si. M.Kom</b>	<b>Zoom Meeting</b>	<b>Discussion of UI/UX Design</b>

			<b>Specifications, Front End, Back End</b>
--	--	--	--

## Documentation of Communication with Client

Here is a picture of evidence of PBL Team 23-2-16 Communication with Project Manager/Project Proponent

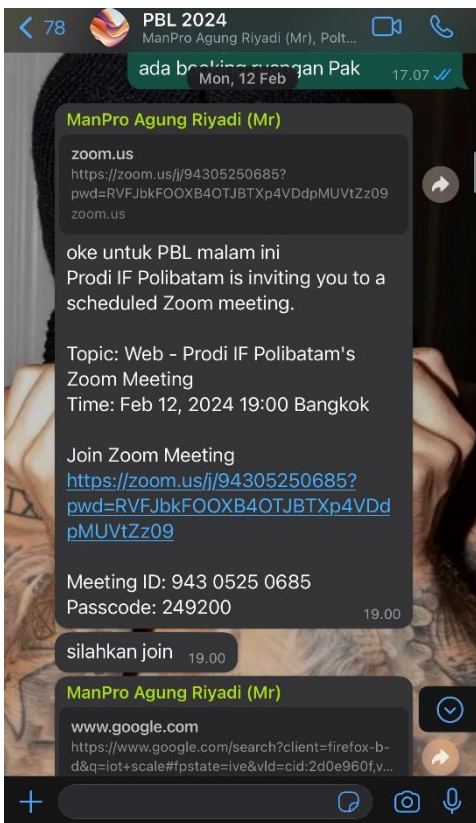


Image 1

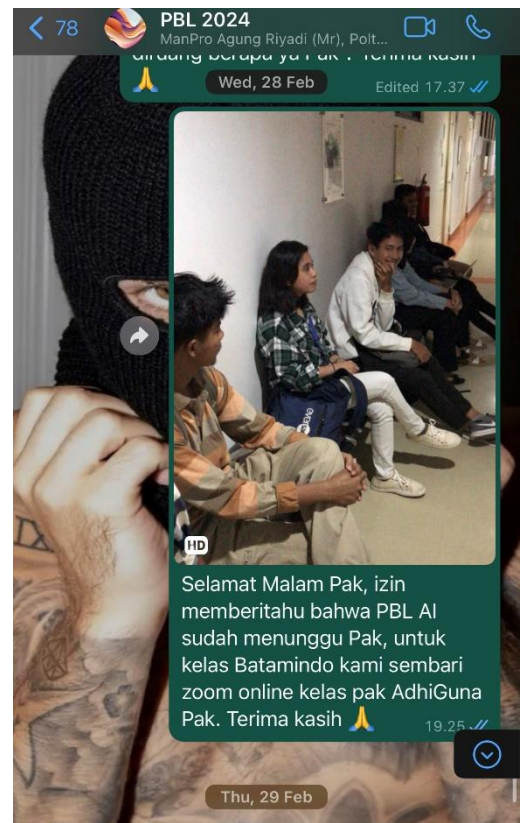


Image 2

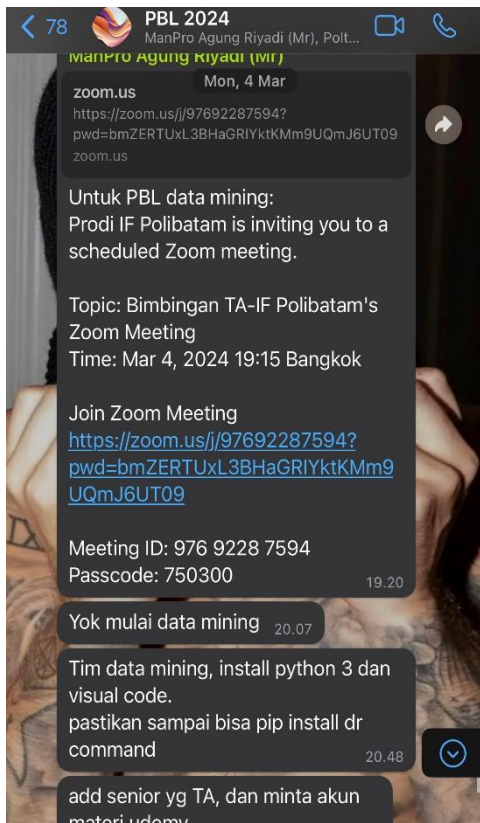


Image 3

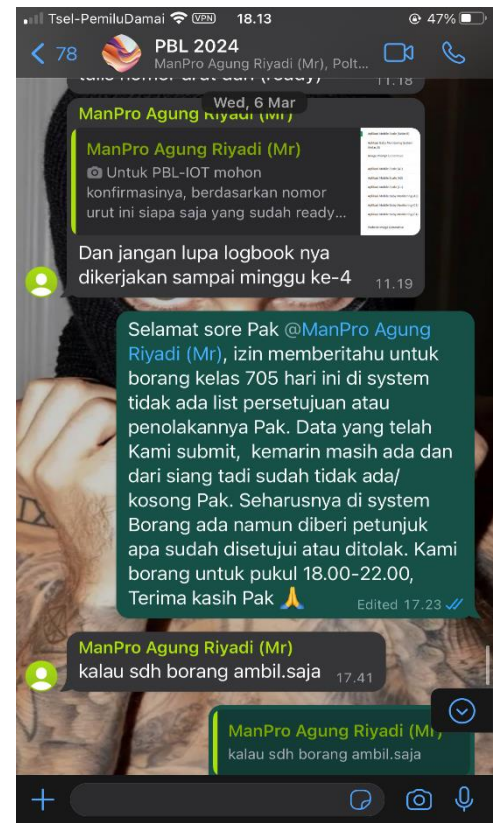


Image 4

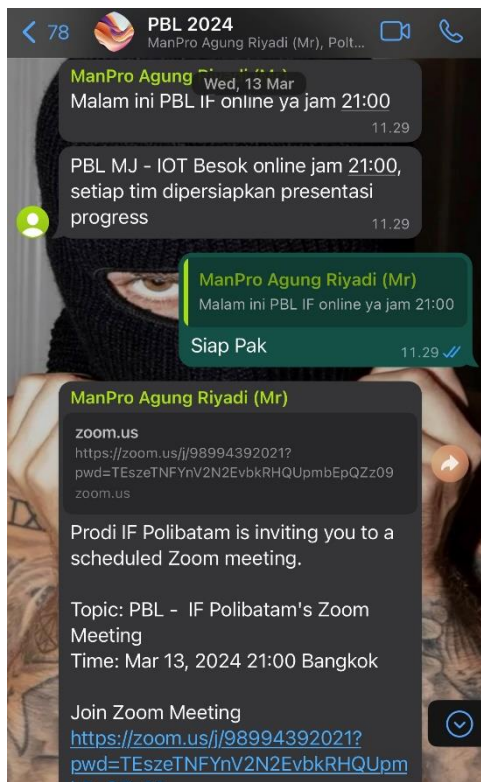


Image 5

# System specifications

## A. General description

The Promptails website has more than 30 million generative artificial intelligence-generated prompt data. Of this data, some can already be classified into several style-based categories such as isometric, gravity, abstract, fractal, oil paintings and others. However, there are still many that have not been classified. Therefore, using data mining, we learnt to model the identified data so that the unclassified data can be classified properly. We also calculated the error for this classification result.

Here is an overview of the system we will build:

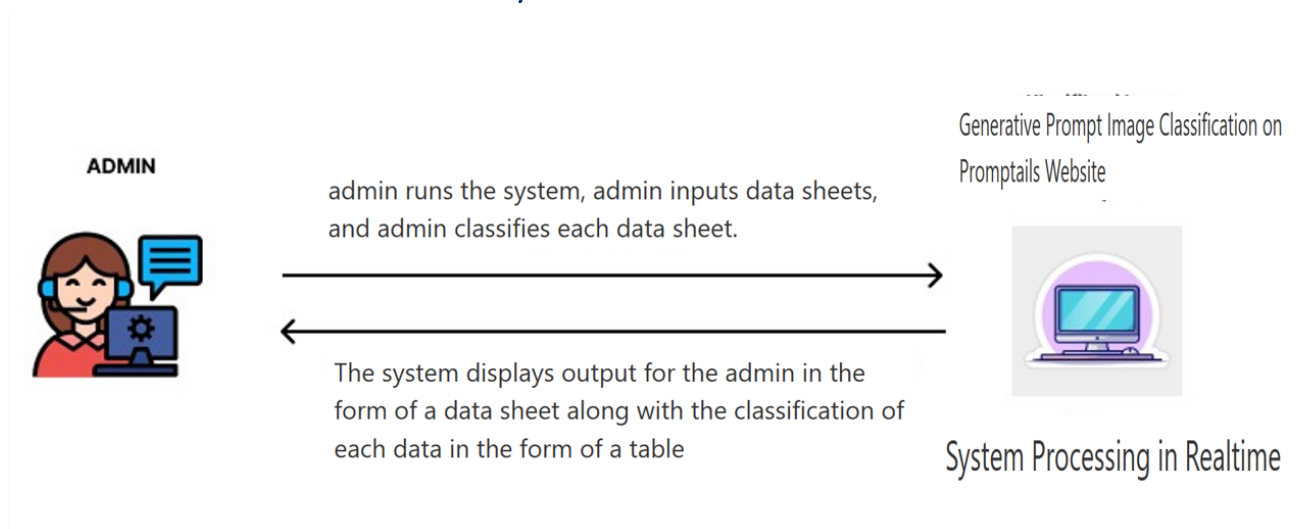


Image 6

Based on the picture above, the features that can be run by the admin on the system are:

- \* Admin can input data sheets
- \* Admin can perform classification for each data sheet after running the system.

While the generative prompt image classification system on the promptails website to the admin issues output in the form of data sheets along with the classification of each data in the form of images.

## B. User Stories

*User Stories are short descriptions that explain user needs in software development. This tool helps us understand what users want from the system or product we are developing.*

Table6

Title: Generative Prompt Image Classification on Promptails Website	
User Access	Access
User	As a user, I would like to see the classification results of new images on the Promptails website.
Administrator	As an administrator, I want to be able to classify the resultant prompt from the AI-based Image Generative Prompt according to the predefined classification.

### ➤ Functional Requirements

Functional requirements describe the specific functions that must be performed by the system to be built. These needs determine what the Generative Prompt Image Classification system can do on the Promptails Website to be built. Here are some of the functional requirements required by the system:

Table 7

No	Functional Requirements
1.	Admin can input datasets into the model.
2.	Admin can train datasets on the model.
3.	Admin can test datasets on the model

---

4.	<b>Admin can perform prompt classification from AI-based Image Generative Prompt.</b>
5.	<b>Admin can save the prompt classification results into the database</b>

➤ **Non-Functional Requirements**

Non-functional requirements describe the characteristics of the system to be built but are not directly related to the main functions, but are important for the performance, security, and usability of the system. The following are some of the non-functional requirements needed by the system as a supporting function of the system's main function:

Table 8

No	Kebutuhan Non-Fungsional
1.	<b>Fast page load time (estimated to be less than 15 seconds)</b>
2.	<b>Security, The system must have adequate security layers to protect sensitive data.</b>
3.	<b>Scalability, The system must be able to increase its capacity horizontally or vertically when required.</b>
4.	<b>Availability, The system should be available on an almost 24-hour basis to users.</b>
5.	<b>Comply with laws and regulations related to privacy, security, and accessibility. As specified in Law No.27 of 2022 on Personal Data Protection</b>



---

## C. System modelling

- Use Case Diagram

*Use case diagram is one type of UML (Unified Modelling Language) diagram that describes the interaction relationship between the system and actors. Use cases can describe the type of interaction between system users and the system. Here is the purpose of making use cases:*

- Shows the sequence of activities in the system.
- Describes business processes, even showing the sequence of activities in a process.
- Become a bridge between the creator and the consumer to describe a system.

The following is the Use case of the Generative Image Prompt Classification application website software system that will be built:



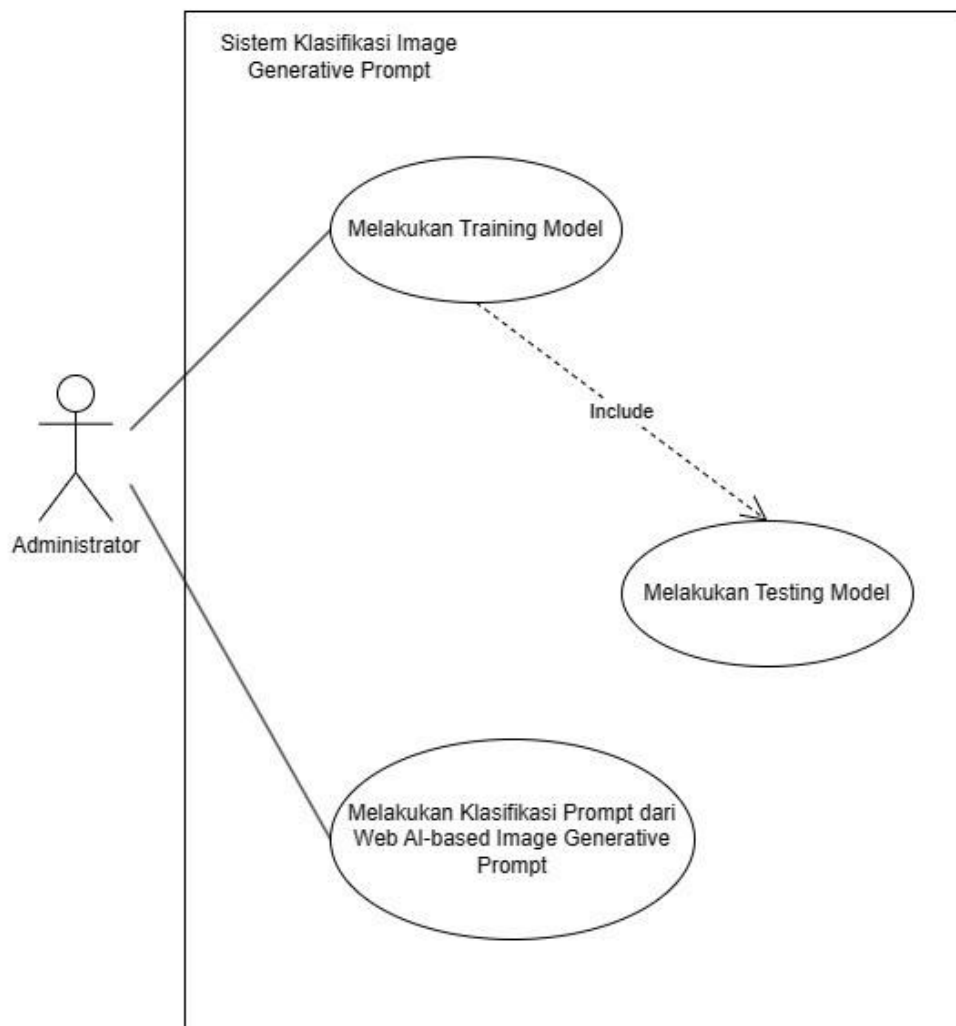


Image 7

In Figure 6, it can be concluded that the system to be built has one actor, namely the admin and two usecases or interactions between the admin and the system. Admin can do training and testing models, admin can also do Prompt Classification of Web AI based Image Generative Prompt.

## D. Database Design

### ➤ ER Diagram (Entity-Relationship Diagram)

An ER Diagram is a visual representation that shows data entities and the relationships between those entities. The following is the ER Diagram for the text classification system:

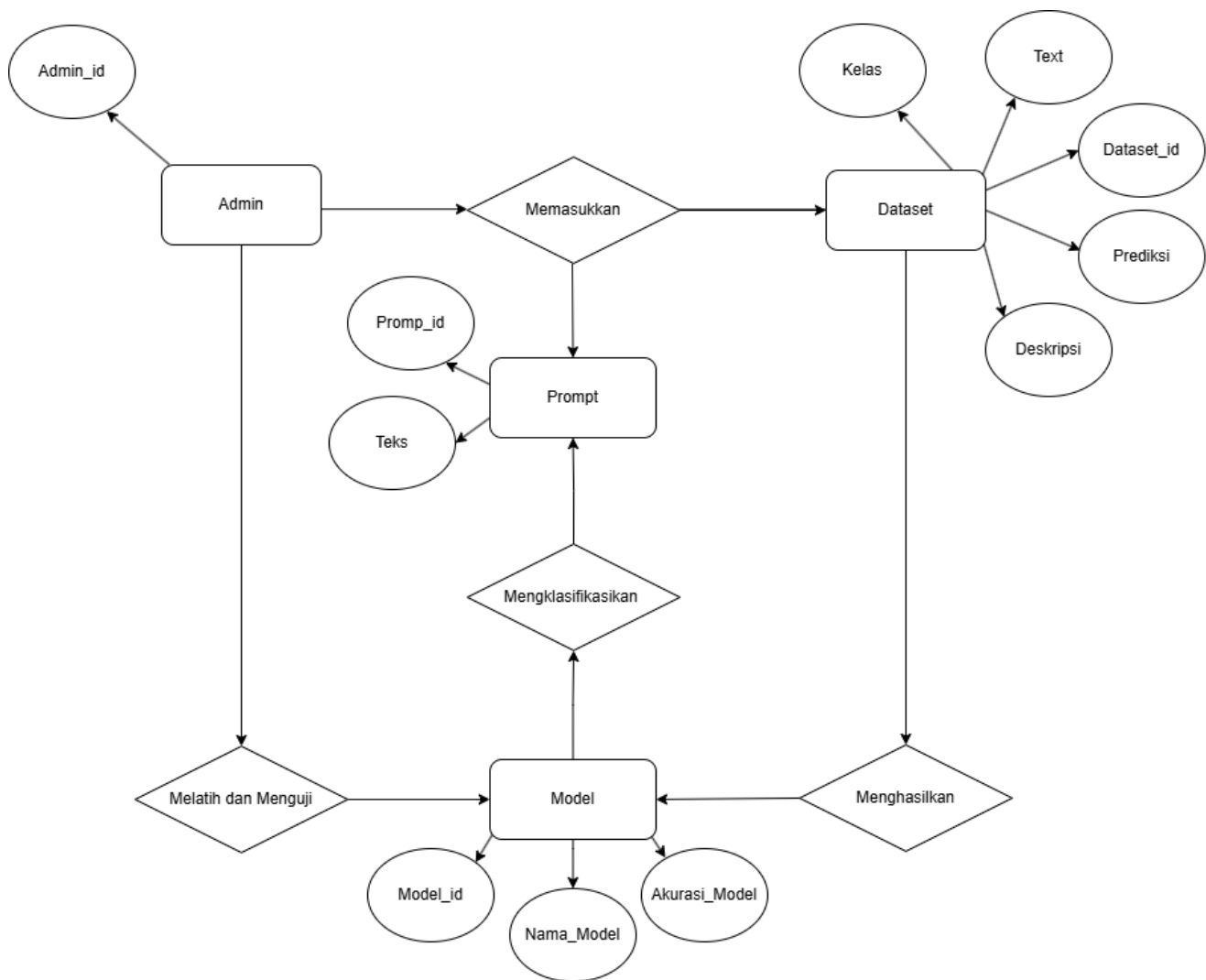


Image 8

---

The following is a detailed explanation of the ER diagram that we created:

- Admin

Attributes:

- 'Admin\_id': Unique identification for each admin.
- Relationship: - Admins have relationships with the 'Prompt' and 'Model' entities where they enter prompts, as well as train and test models.

- Prompt

Attributes:

- 'Prompt\_id': Unique identification for each prompt.
- 'Text': The text of the prompt entered by the admin.

Relationship:

- The prompt has a relationship with the 'Dataset' entity through the 'Classify' activity.
- The prompt also has a relationship with the 'Admin' who entered the prompt.

- Dataset

Attributes:

- 'Dataset\_id': Unique identification for each dataset.
- 'Class;': The class of the dataset used for classification.
- 'Text': The text in the dataset.
- 'Prediction': The prediction results of the model using this dataset.
- 'Description': Description of the dataset. Relationship: The dataset is entered by 'Admin' and generates the data used by 'Model'.

Relationship:

- Dataset entered by 'Admin' and generated data used by 'Model'.

- 
- Model  
Attributes:
    - 'Model\_id': Unique identification for each model.
    - 'Model\_name': The name of the model.
    - 'Accuracy\_Model': Accuracy of the model after training and testing.
  - Relationship:
    - The model is trained and tested by 'Admin'.
    - The model also classifies the prompts entered by 'Admin'.
  - Main Activities in the System
    - Input:  
'Admin' enters 'Dataset' and 'Prompt' into the system.
    - Classifying:  
The 'Prompt' is classified using the trained 'Model'.
    - Train and Test:  
'Admin' trains and tests the 'Model' using the 'Dataset'.
    - Generate:  
The 'Dataset' generates the prediction data used by the 'Model'.

## Process Flow

1. Admin enters the dataset and prompt into the system.
2. The model is trained and tested using the dataset by the admin.
3. The prompts entered by the admin are then classified by the trained model.
4. The model generates predictions based on the prompt and dataset used.

---

## E. Interface Design

### 1. *Wireframe*

A wireframe is a simple visual representation of a user interface or web page that shows the arrangement of key elements without specific design or content details. It is an initial blueprint used in the design process to plan the layout and structure of a digital product, such as a website or app. Wireframes are usually sketches or rough drawings that display the relative positions of elements such as text, images, buttons and input fields. Here is how the wireframe looks like on the project we are creating:

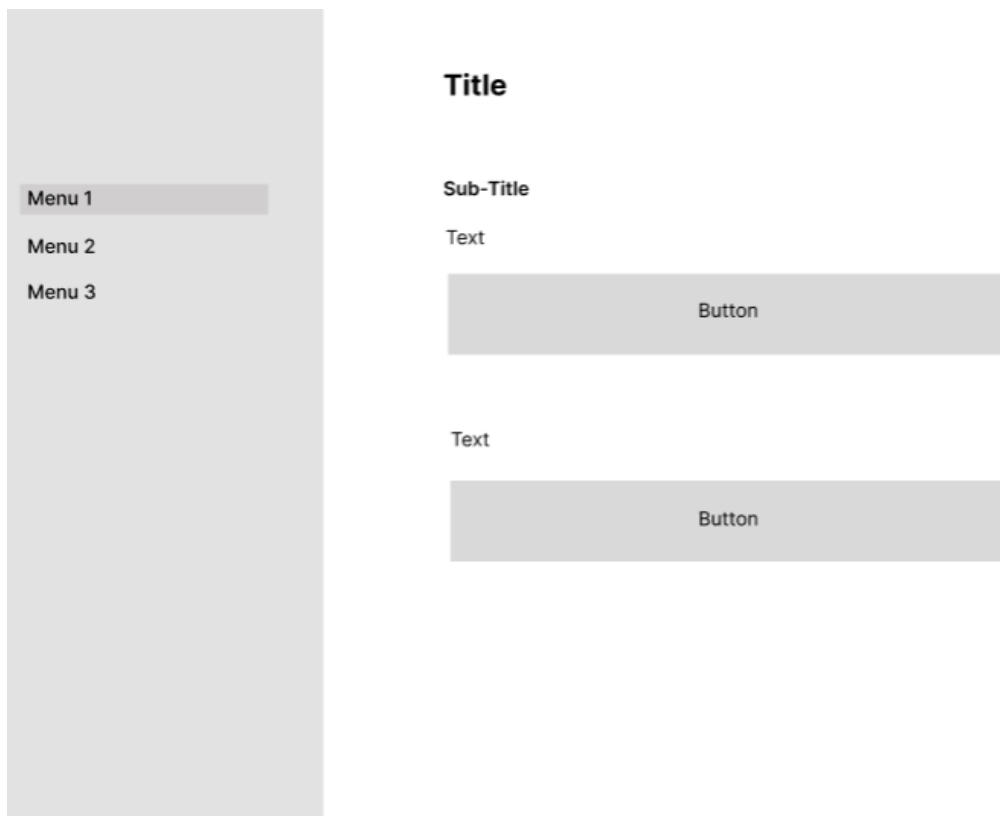
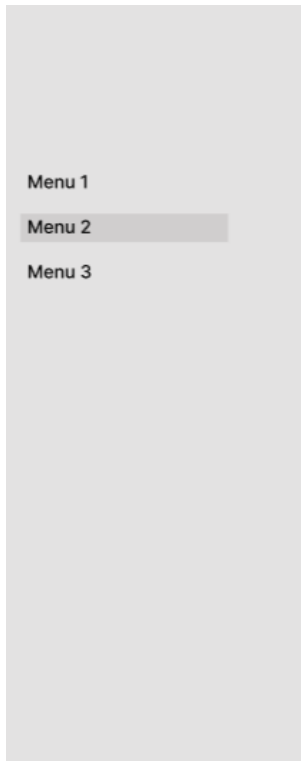


Image 9



## Title

Sub-Title

Text

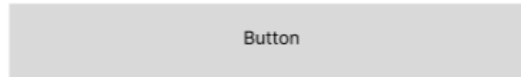
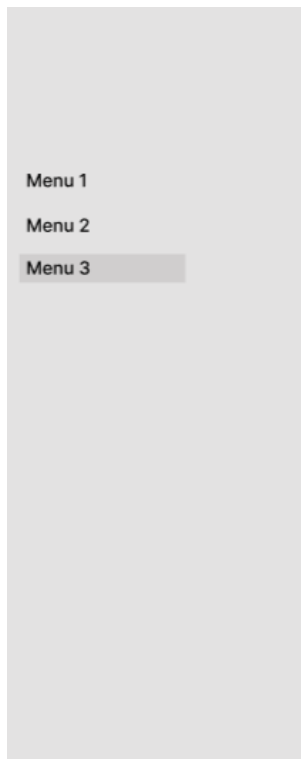


Image 10



## Title

Sub-Title

Text



Image 11

## 2. *Mock up*

A Mock Up is a more complete and detailed visual representation of a user interface or web page created after the wireframing stage. Mock Ups contain design elements such as colours, fonts, graphics, and other UI elements that give a more accurate picture of the final look of the digital product. Here's a look at the Mock up project we are creating:

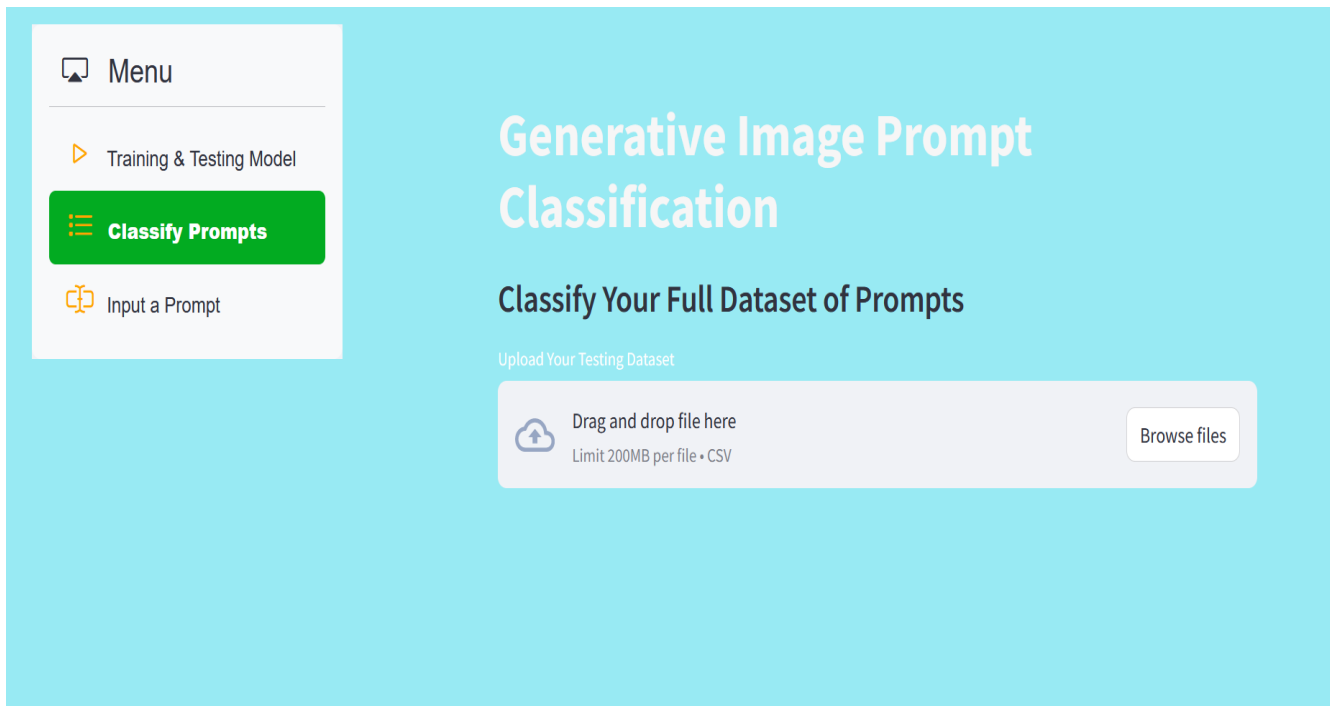


Image 12

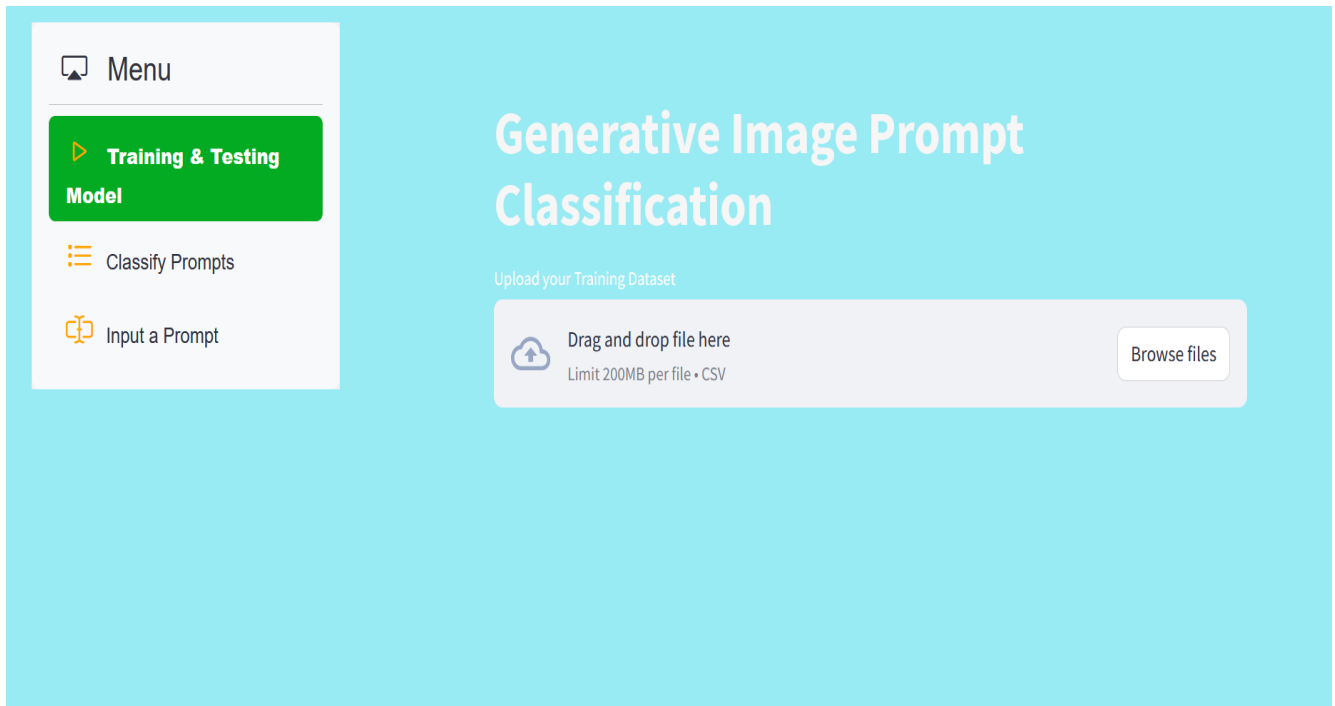


Image 13

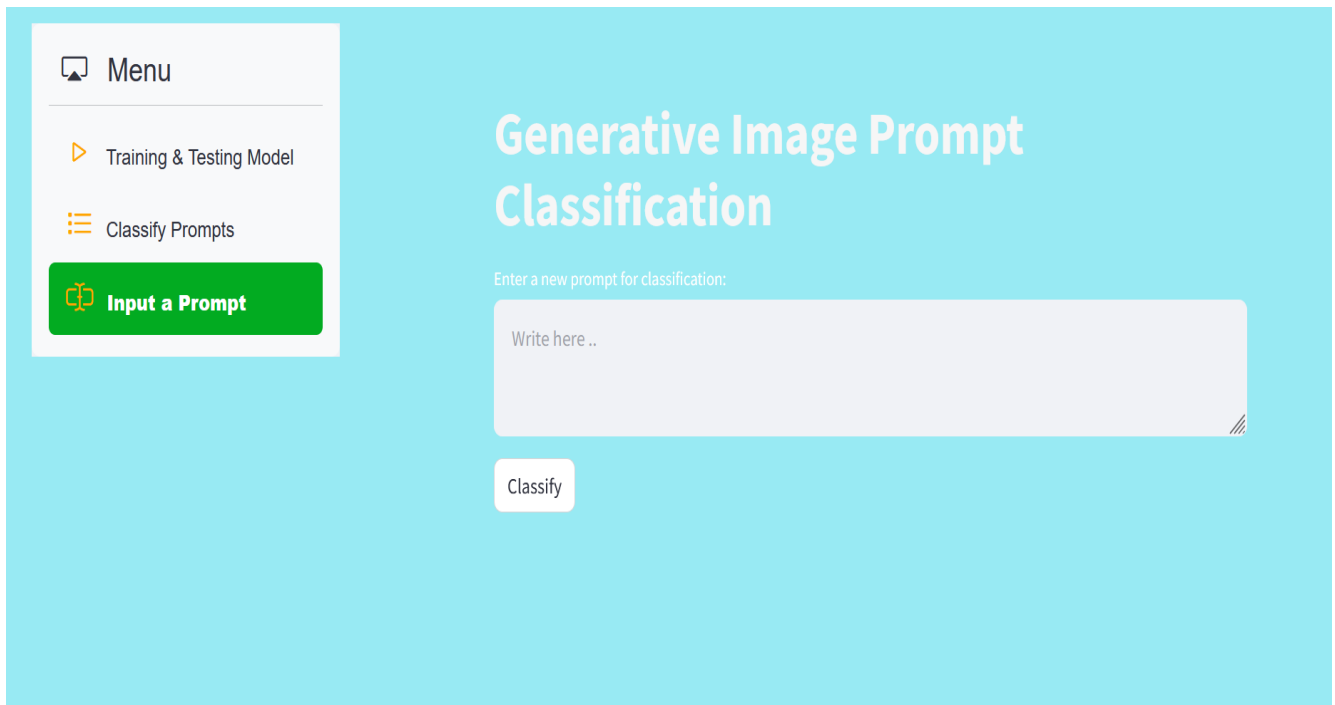


Image 14



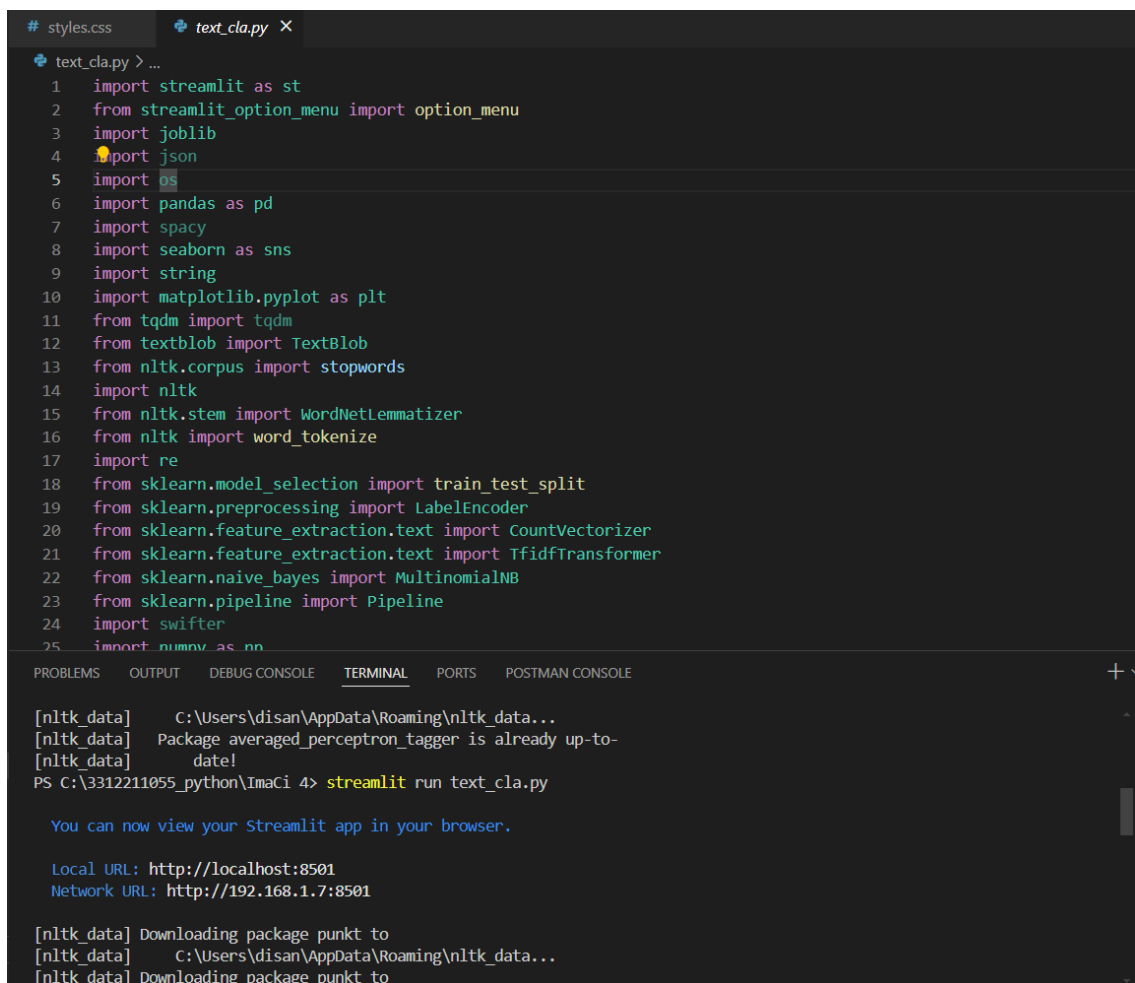
# Implementation Results

## A. Application Implementation

### 1. Application Architecture

This application is a web-based application built using Streamlit framework for UI and Scikit-learn based machine learning pipeline for data processing and classification. The application architecture consists of several main components:

- Frontend (UI): Using Streamlit to create an interactive user interface.



```
# styles.css  text_cla.py X
text_cla.py > ...
1  import streamlit as st
2  from streamlit_option_menu import option_menu
3  import joblib
4  import json
5  import os
6  import pandas as pd
7  import spacy
8  import seaborn as sns
9  import string
10 import matplotlib.pyplot as plt
11 from tqdm import tqdm
12 from textblob import TextBlob
13 from nltk.corpus import stopwords
14 import nltk
15 from nltk.stem import WordNetLemmatizer
16 from nltk import word_tokenize
17 import re
18 from sklearn.model_selection import train_test_split
19 from sklearn.preprocessing import LabelEncoder
20 from sklearn.feature_extraction.text import CountVectorizer
21 from sklearn.feature_extraction.text import TfidfTransformer
22 from sklearn.naive_bayes import MultinomialNB
23 from sklearn.pipeline import Pipeline
24 import swifter
25 import numpy as np
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS POSTMAN CONSOLE

```
[nltk_data] C:\Users\disan\AppData\Roaming\nltk_data...
[nltk_data] Package averaged_perceptron_tagger is already up-to-
[nltk_data] date!
PS C:\3312211055_python\ImaCi 4> streamlit run text_cla.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.7:8501

[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\disan\AppData\Roaming\nltk_data...
[nltk_data] Downloading package punkt to
```

Image 15

- Backend: Using Python with various libraries for text processing, such as NLTK, TextBlob, and Spacy.

```
# Functions to calculate sentiment and text length
def subj_txt(text):
    return TextBlob(text).sentiment[1]

def polarity_txt(text):
    return TextBlob(text).sentiment[0]

def len_text(text):
    if len(text.split()) > 0:
        return len(set(clean_txt(text).split())) / len(text.split())
    else:
        return 0

13 from nltk.corpus import stopwords
14 import nltk
15 from nltk.stem import WordNetLemmatizer
16 from nltk import word_tokenize
```

Image 16

- Machine Learning: Implementation of machine learning pipeline using Scikit-learn with Multinomial Naive Bayes model for text classification.

```
18 from sklearn.model_selection import train_test_split
19 from sklearn.preprocessing import LabelEncoder
20 from sklearn.feature_extraction.text import CountVectorizer
21 from sklearn.feature_extraction.text import TfidfTransformer
22 from sklearn.naive_bayes import MultinomialNB
23 from sklearn.pipeline import Pipeline
```

Image 17

- Data Storage: Pandas uses for temporary data manipulation, no permanent data storage is involved.

```
6 import pandas as pd
```

Image 18

## 2. Working Features Implemented

➤ Training & Testing Model:

- Upload datasets for model training.
- Text cleaning and preprocessing.
- Text classification model training.
- Model evaluation by displaying accuracy and prediction examples.

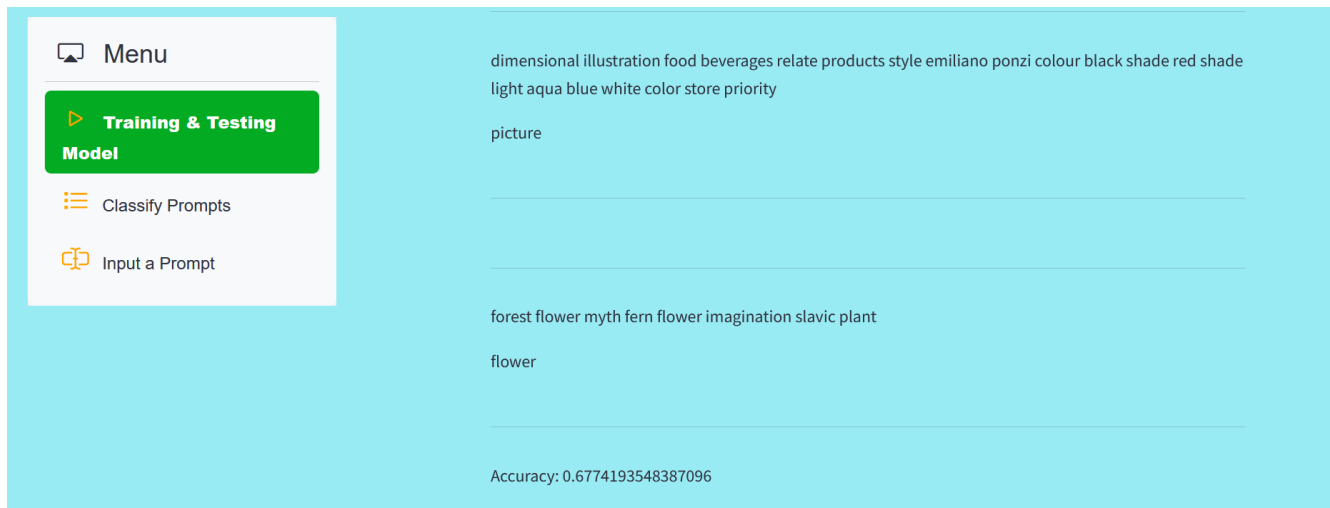
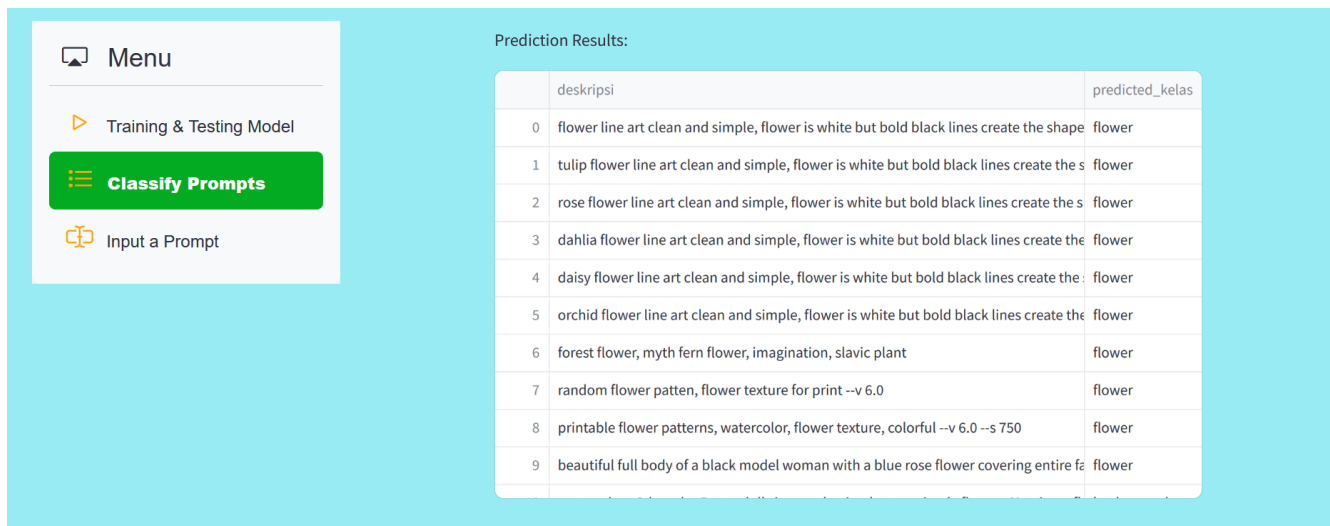


Image 19

### ➤ Classify Prompts:

- Upload new dataset for prediction.
- Cleaning of new text.
- Text class prediction using the trained model.
- Display the prediction result.



The screenshot shows a web application interface. On the left is a 'Menu' sidebar with three options: 'Training & Testing Model' (with a play icon), 'Classify Prompts' (highlighted with a green background and a list icon), and 'Input a Prompt' (with a speech bubble icon). The main area on the right is titled 'Prediction Results:' and contains a table with two columns: 'deskripsi' and 'predicted\_kelas'.

	deskripsi	predicted_kelas
0	flower line art clean and simple, flower is white but bold black lines create the shape	flower
1	tulip flower line art clean and simple, flower is white but bold black lines create the s	flower
2	rose flower line art clean and simple, flower is white but bold black lines create the s	flower
3	dahlia flower line art clean and simple, flower is white but bold black lines create the	flower
4	daisy flower line art clean and simple, flower is white but bold black lines create the	flower
5	orchid flower line art clean and simple, flower is white but bold black lines create the	flower
6	forest flower, myth fern flower, imagination, slavic plant	flower
7	random flower patten, flower texture for print --v 6.0	flower
8	printable flower patterns, watercolor, flower texture, colorful --v 6.0 --s 750	flower
9	beautiful full body of a black model woman with a blue rose flower covering entire fa	flower

Image 20

### ➤ *Input a Prompt:*

- Manual text input by the user.
- Input text cleaning.
- Prediction of input text class using the trained model.
- Display the prediction result.

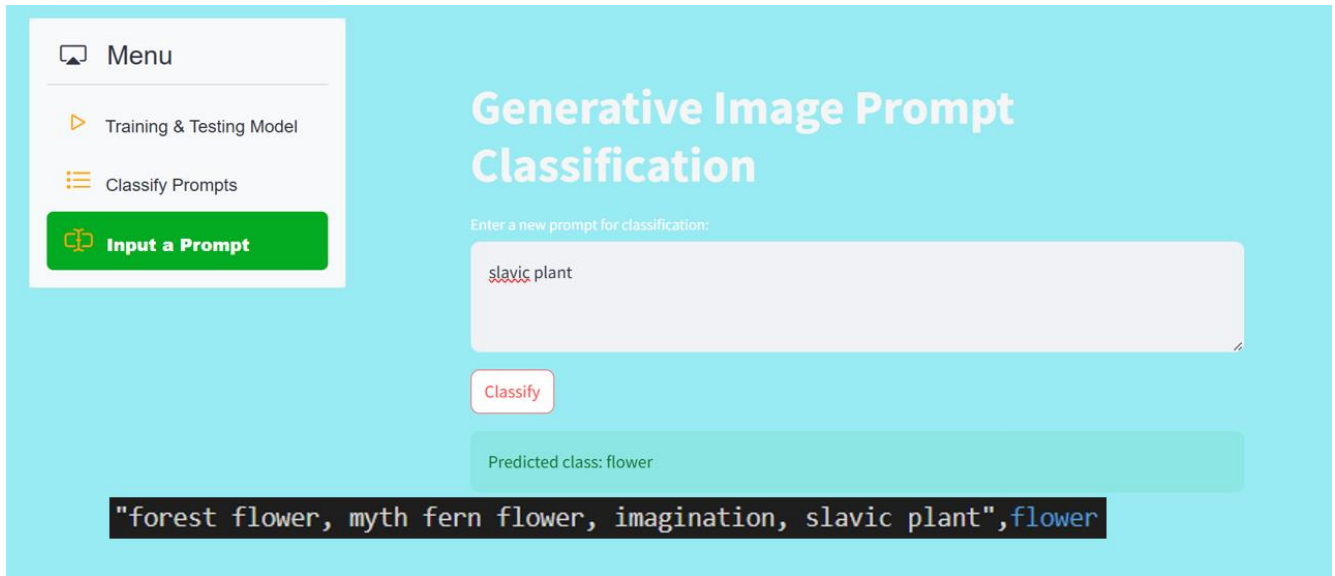


Image 21



Image 22

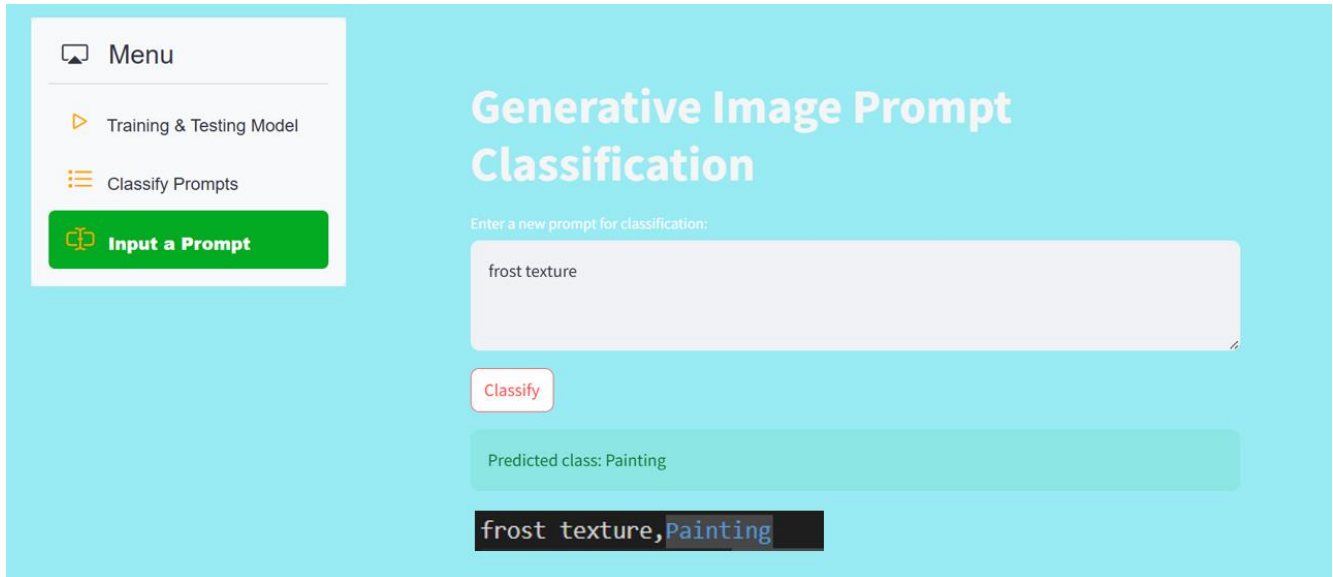


Image 23



Image 24

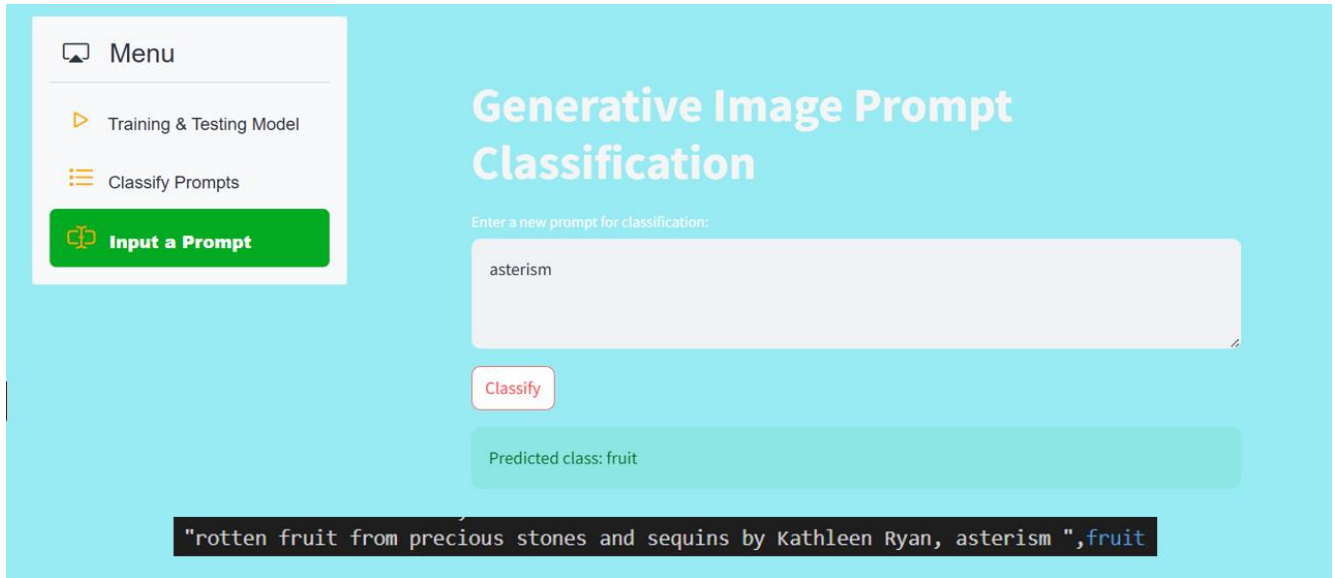


Image 25

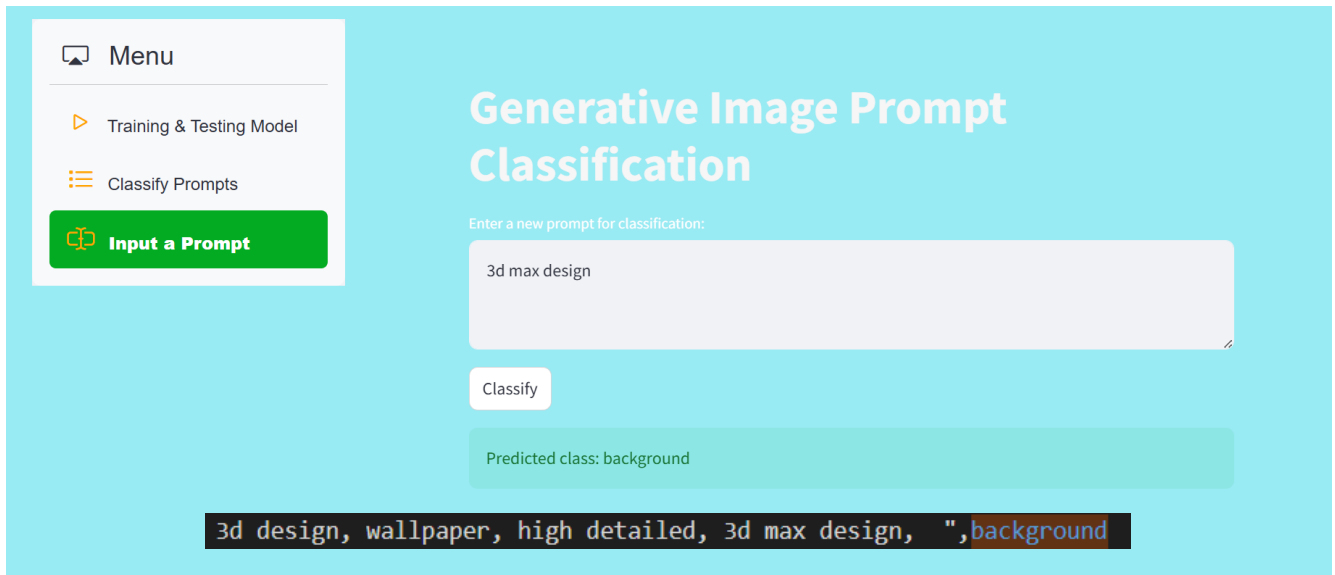


Image 26

---

➤ Features That Don't Work:

- Implementation of IoT features and permanent data storage: This application does not involve IoT components or data storage in a permanent database.
- Error handling for various cases: More comprehensive error handling may be required to handle various scenarios such as corrupted files, inappropriate file formats, etc.

Overall, the application successfully implemented the main planned features, providing the ability for model training, text prediction, and manual text input with results displayed interactively through a Desktop Application-based user interface.

### 3. System Implementation Method

➤ Text Processing:

- Text cleaning is done by removing punctuation, numbers, and common stop words.
- Lematisation uses NLTK to convert words into base forms.

➤ Machine Learning Models:

- The pipeline consists of CountVectorizer, TfidfTransformer, and MultinomialNB.
- Model training is performed using the dataset uploaded by the user.
- The model is saved using Joblib for use in the next session.



---

➤ User Interface:

- Created using Streamlit to provide an easy-to-use interface for file upload, text input, and displaying prediction results.
- Streamlit option menu is used for navigation between features.

#### 4. Programming Language

- Python: Used for the entire application, both backend (text processing, model training and prediction) and frontend (Streamlit).

#### 5. Framework

- Streamlit: For creating user interfaces.
- Scikit-learn: For building and training machine learning models.
- Pandas: For data manipulation and analysis.
- Seaborn and Matplotlib: For data visualisation.
- NLTK, TextBlob, Spacy: For text processing.

#### 6. Database

- There is no permanent database: Data is stored and processed in memory using Pandas.

## B. Database Implementation

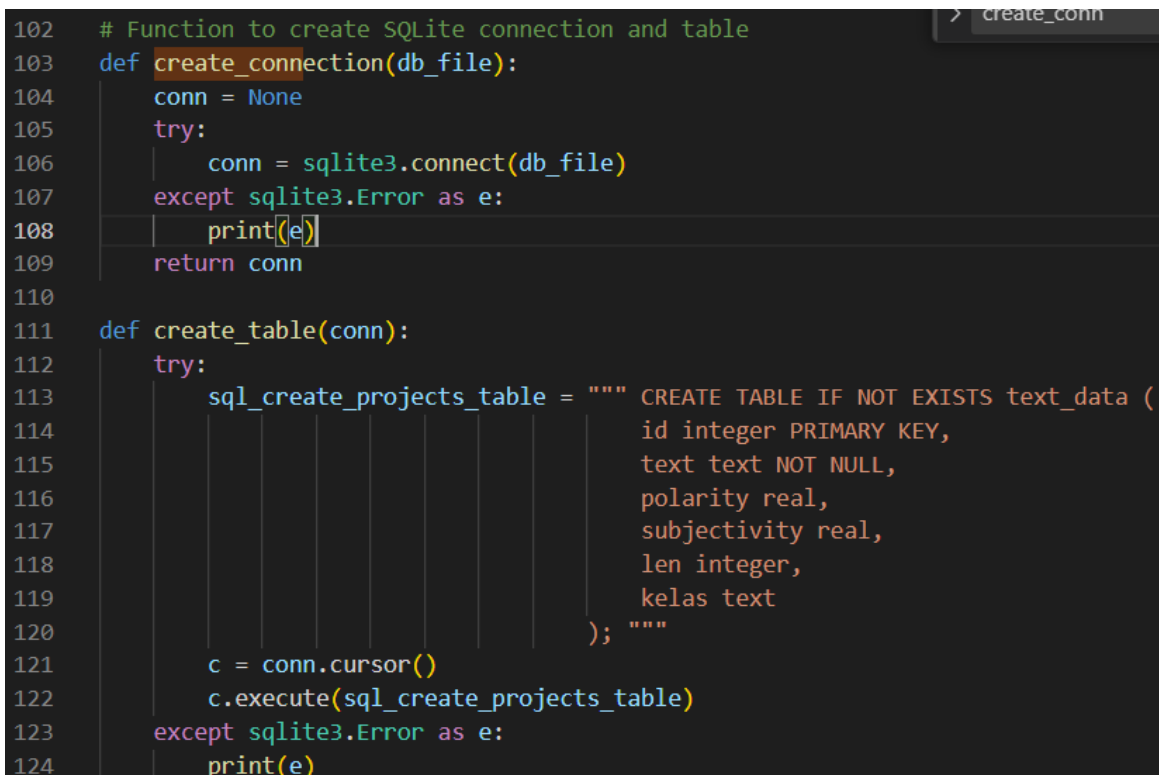
For the database implementation in this project, we use SQLite as the Database Management System (DBMS) due to its ease of setup and lightweight usage and does not require a separate server. SQLite is also easy to integrate with Python, which is the main programming language in this project.

Implementation Steps:

➤ **Creating Connections to Databases and Tables:**

- The 'create\_connection' function is used to create a connection to the SQLite database file.
- The 'create\_table' function is used to create the 'text\_data' table if it does not already exist.

Here is the code to create connections and tables:

A screenshot of a code editor with a dark background and light-colored text. The code is written in Python and defines two functions: 'create\_connection' and 'create\_table'. The 'create\_connection' function takes 'db\_file' as an argument and attempts to connect to a SQLite database. The 'create\_table' function takes a connection object 'conn' and creates a table named 'text\_data' with various columns. The code is line-numbered from 102 to 124. A tab labeled 'create\_conn' is visible in the top right corner of the editor.

```
102 # Function to create SQLite connection and table
103 def create_connection(db_file):
104     conn = None
105     try:
106         conn = sqlite3.connect(db_file)
107     except sqlite3.Error as e:
108         print(e)
109     return conn
110
111 def create_table(conn):
112     try:
113         sql_create_projects_table = """ CREATE TABLE IF NOT EXISTS text_data (
114             id integer PRIMARY KEY,
115             text text NOT NULL,
116             polarity real,
117             subjectivity real,
118             len integer,
119             kelas text
120         ); """
121         c = conn.cursor()
122         c.execute(sql_create_projects_table)
123     except sqlite3.Error as e:
124         print(e)
```

Image 27

## ➤ Saving Data to a Database:

- After cleaning and processing the text data, the data is saved to the 'text\_data' table in the SQLite database using the 'to\_sql' method of Pandas.

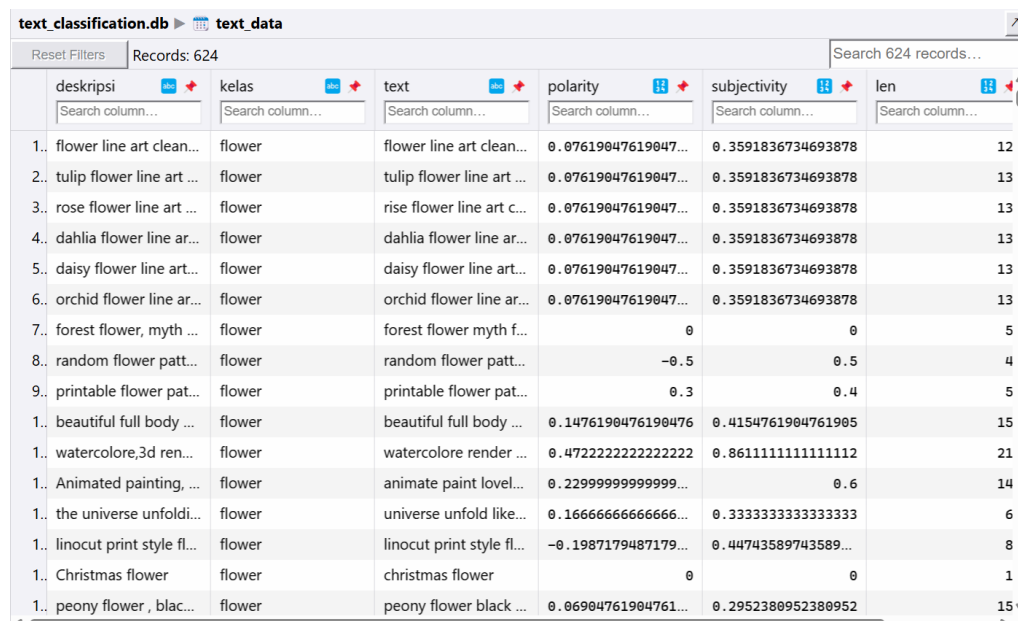
Here is the code to save the data:

```
219 conn = create_connection('text_classification.db')
220 if conn is not None:
221     create_table(conn)
222     df.to_sql('text_data', conn, if_exists='replace', index=False)
223     st.write('Data saved to SQLite database')
224 else:
225     st.write("Error! Cannot create the database connection.")
```

Image 28

## ➤ Sample Query:

- Here is all the data from the 'text\_data' table, in SQLite3:



	deskripsi	kelas	text	polarity	subjectivity	len
1..	flower line art clean...	flower	flower line art clean...	0.07619047619047...	0.3591836734693878	12
2..	tulip flower line art ...	flower	tulip flower line art ...	0.07619047619047...	0.3591836734693878	13
3..	rose flower line art ...	flower	rise flower line art c...	0.07619047619047...	0.3591836734693878	13
4..	dahlia flower line ar...	flower	dahlia flower line ar...	0.07619047619047...	0.3591836734693878	13
5..	daisy flower line art...	flower	daisy flower line art...	0.07619047619047...	0.3591836734693878	13
6..	orchid flower line ar...	flower	orchid flower line ar...	0.07619047619047...	0.3591836734693878	13
7..	forest flower, myth ...	flower	forest flower myth f...	0	0	5
8..	random flower patt...	flower	random flower patt...	-0.5	0.5	4
9..	printable flower pat...	flower	printable flower pat...	0.3	0.4	5
1..	beautiful full body ...	flower	beautiful full body ...	0.1476190476190476	0.4154761904761905	15
1..	watercolore,3d ren...	flower	watercolore render ...	0.4722222222222222	0.8611111111111112	21
1..	Animated painting, ...	flower	animate paint lovel...	0.2299999999999999	0.6	14
1..	the universe unfoldi...	flower	universe unfold like...	0.1666666666666666	0.3333333333333333	6
1..	linocut print style fl...	flower	linocut print style fl...	-0.1987179487179...	0.44743589743589...	8
1..	Christmas flower	flower	christmas flower	0	0	1
1..	peony flower , blac...	flower	peony flower black ...	0.06904761904761...	0.2952380952380952	15

Image 29

## C. Application Testing and Deployment

Application testing is carried out to ensure that the application functions as expected. The following are the steps of testing the application made, complete with test cases or test scenarios, test dates, parties who conduct tests and endorsements and documentation.

### ➤ TEST PLAN & EXECUTION USER ACCEPTANCE TEST

TEST PLAN & EXECUTION USER ACCEPTANCE TEST	
Project Name	: PBL IF 23-2-16
Title of PBL	: Generative Prompt Image Classification on Promptails Website
Project Manager	: Agung Riyadi, S.Si. M.Kom
Date of Document	: 9 June 2024

Table 9

RESULTS OF USER ACCEPTANCE TESTING (UAT)				
UAT ID	Testing Scenarios	Examiner's Name	Status	Additional Comments
UAT01	<p><b>Functional Reference ID:</b> F001</p> <p><b>Test Description:</b> This test is performed to ensure that the admin can enter the dataset into the model correctly. The purpose of the test is to verify that the system can accept and process the dataset provided by the admin without error.</p> <p><b>Test Scenario:</b></p> <ol style="list-style-type: none"><li>1. Open the browser</li><li>2. Navigate to the Input a Prompt page</li><li>3. Upload the dataset file</li><li>4. Click the submit button</li></ol>	Irgi Haslan Putra	Successful	-

	<b>Expected result:</b> The user is redirected to a confirmation page with the message "Dataset uploaded successfully."			
UAT02	<p><b>Functional Reference ID:</b> F002</p> <p><b>Test Description:</b> This test is performed to ensure that the admin can correctly train the dataset on the model. The purpose of testing is to verify that the training process runs according to specifications without any errors or interference.</p> <p><b>Test Scenario:</b></p> <ol style="list-style-type: none"> <li>1. Open the browser</li> <li>2. Navigate to the Training &amp; Testing Model page</li> <li>3. Select the dataset that you want to train</li> <li>4. Click the "Start Training" button</li> <li>5. Wait for the training process to finish</li> </ol> <p><b>Expected result:</b> The system displays the message "Training successful" and the results of the trained model.</p>	Meysy Sinaga	Successful	-
UAT03	<p><b>Functional Reference ID:</b> F003</p> <p><b>Test Description:</b> This test is performed to ensure that the admin can test the dataset in the model correctly. The test objective is to verify that the testing process runs according</p>	David Patrick Ken	Successful	-

	<p>to specifications and the results are accurate.</p> <p><b>Test Scenario:</b></p> <ol style="list-style-type: none"> <li>1. Open the browser</li> <li>2. Navigate to the Training &amp; Testing Model page</li> <li>3. Select the dataset that you want to test</li> <li>4. Click the "Start Testing" button</li> <li>5. Wait until the testing process is complete</li> </ol> <p><b>Expected result:</b> The system displays testing results with model performance statistics.</p>			
UAT04	<p><b>Functional Reference ID:</b> F004</p> <p><b>Test Description:</b> This test is conducted to ensure that the admin can classify the prompt from the AI-based Image Generative Prompt correctly. The test objective is to verify that the classification runs well and the results are as expected.</p> <p><b>Test Scenario:</b></p> <ol style="list-style-type: none"> <li>1. Open the browser</li> <li>2. Navigate to the Classify Prompts page</li> <li>3. Select the prompt you want to classify</li> <li>4. Click the "Start Classification" button</li> <li>5. Wait for the classification process to finish</li> </ol>	Galih Ivan	Successful	-

	<b>Expected result:</b> The system displays the classification results with the appropriate category.			
UAT05	<b>Functional Reference ID:</b> F005	Irgi Haslan Putra	Successful	-
	<p><b>Test Description:</b> This test is performed to ensure that the admin can add a new prompt for classification correctly. The test objective is to verify that the system accepts and processes new prompts without error and performs classification correctly.</p> <p><b>Test Scenario:</b></p> <ol style="list-style-type: none"> <li>1.Open the browser</li> <li>2.Navigate to the Input a Prompt page</li> <li>3.Enter a new prompt text in the input box</li> <li>4.Click the "Classify Prompt" button</li> <li>5.Wait for the classification process to finish</li> </ol> <p><b>Expected result:</b> The system displays the classification result of the new prompt with the appropriate category.</p>			
UAT06	<b>Functional Reference ID:</b> F008	David Patrick Ken	Successful	-
	<b>Test Description:</b> This test is performed to ensure that the admin can change the configuration and model settings correctly. The test objective is to verify that			

---

	<p>configuration changes are saved and applied without error.</p> <p><b>Test Scenario:</b></p> <ol style="list-style-type: none"> <li>1.Open the browser</li> <li>2.Navigate to the Settings page</li> <li>3.Change one of the model settings (e.g., learning rate parameter)</li> <li>4.Click the "Save Changes" button</li> <li>5.Refresh the page and verify that the changes are saved</li> </ol> <p><b>Expected result:</b> The system saves and applies the configuration changes successfully.</p>			
--	---	--	--	--



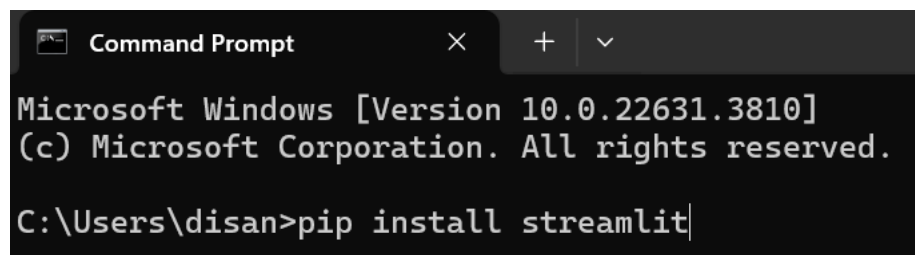
## ➤ Software Configuration and Installation Flow

To ensure the application can run properly, here are the configuration and software installation steps that need to be done:

### Software Configuration and Installation

#### Step 1: Python and Streamlit Installation

- Make sure Python is already installed on your system. If not, download and install Python from the official Python website.
- Install Streamlit using pip:

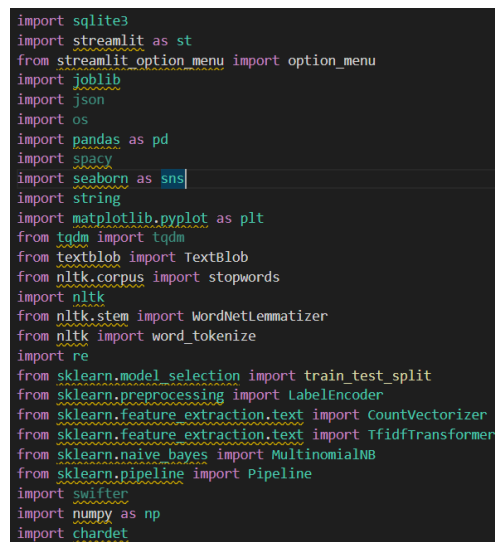


```
Microsoft Windows [Version 10.0.22631.3810]
(c) Microsoft Corporation. All rights reserved.

C:\Users\disan>pip install streamlit
```

Image 30

#### Step 2: Installing Project Dependencies



```
import sqlite3
import streamlit as st
from streamlit_option_menu import option_menu
import joblib
import json
import os
import pandas as pd
import spacy
import seaborn as sns
import string
import matplotlib.pyplot as plt
from tqdm import tqdm
from textblob import TextBlob
from nltk.corpus import stopwords
import nltk
from nltk.stem import WordNetLemmatizer
from nltk import word_tokenize
import re
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
import swifter
import numpy as np
import chardet
```

Image 31

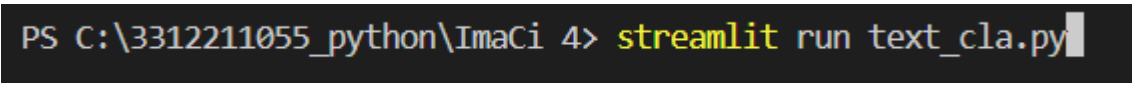
---

### Step 3: SQLite Database Configuration

- The SQLite database will be automatically created when the application starts. No additional steps are required for database configuration.

### Step 4: Running the Application

- To run the Streamlit application, open a terminal or command prompt, navigate to the project directory.



```
PS C:\3312211055_python\ImaCi 4> streamlit run text_cla.py
```

A screenshot of a Windows command prompt window. The prompt shows the current directory as C:\3312211055\_python\ImaCi and the command 'streamlit run text\_cla.py' has been entered. The 'streamlit' command is highlighted in yellow. A white cursor is at the end of the command line.

Image 32

LOAD TESTING RESULTS WITH LOCUST

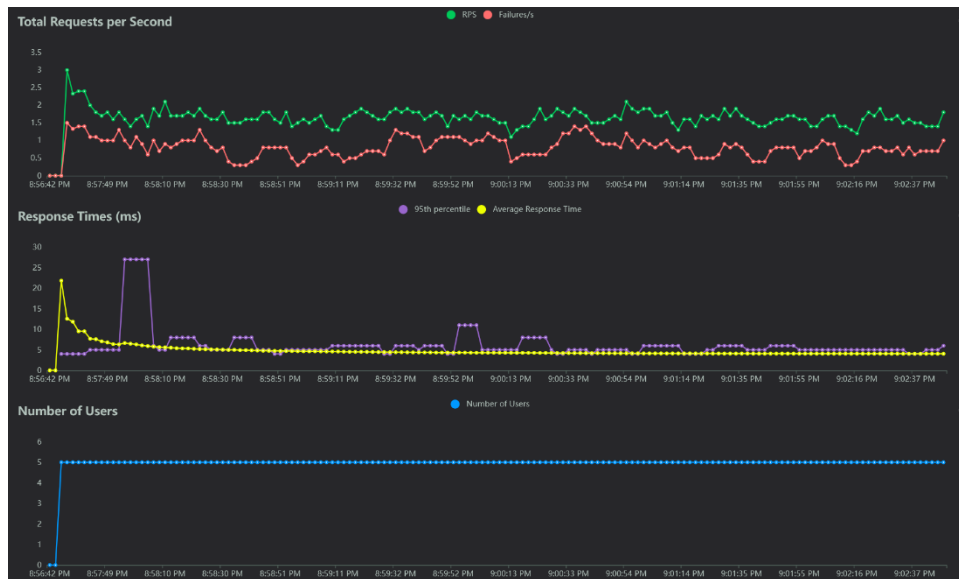
Table 10

ID	: LOT001	
Testing Description	: This test was conducted to evaluate the performance of the system when accessed by 50 users simultaneously. The goal is to ensure that the system remains responsive and functions properly under moderate workloads.	
Method Type	: GET	
Test Scenario	Total Users	: 50
	Ramp Up	: 50 per second
	Time	: 5 Min
Response Time (ms)	Min	: 1 Min(ms)
	Average	: 11.85
	Max	: 3130
Requests	Total Requests	: 20 per second
	Failed Requests	: 8
Error rate (%)	:	
Testing Chart	Screenshot of Locust	

The screenshot displays three line charts from a Locust load test. The top chart, 'Total Requests per Second', shows a green line for RPS peaking at approximately 25 and a red line for failures per second remaining near zero. The middle chart, 'Response Times (ms)', shows a purple line for the 95th percentile and a yellow line for the average response time, both peaking around 100ms and then stabilizing. The bottom chart, 'Number of Users', shows a blue line that ramps up to 50 users and remains constant. All charts share a common x-axis representing time from 7:58:20 PM to 8:03:28 PM.

Table 11

<b>ID</b>	: LOT002	
<b>Testing Description</b>	: This test was conducted to evaluate the performance of the system when accessed by 5 users simultaneously. The goal is to ensure that the system remains responsive and functions properly under low workloads.	
<b>Method Type</b>	: POST	
<b>Test Scenario</b>	Total Users	: 5
	Ramp Up	: 5
	Time	: 6 Min
<b>Response Time (ms)</b>	Min	: 2 Min(ms)
	Average	: 3.49
	Max	: 42
<b>Requests</b>	Total Requests	: 730
	Failed Requests	: 730
<b>Error rate (%)</b>	:	
<b>Testing Chart</b>	Screenshot of Locust	



---

# Conclusion

## Summary

The Generative Prompt Image Classification project on the Promptails Website has successfully achieved the main goal of developing a machine learning-based text classification system that can be used for various applications, such as sentiment analysis, topic detection, and so on. In the process of working on this project, we have gone through several important stages, from planning, analysis, design, implementation, to testing.

Overall, this project provided valuable experience in developing machine learning-based applications and demonstrated the importance of each stage in the software development cycle. The end result of a ready-to-use application shows that collaboration and good planning can produce a quality product. We hope that the system that has been developed can continue to be improved and adapted for wider needs in the future. Thank you to all those who have supported and contributed to this project.

---

## Lesson Learned

- **Data Quality and Diversity Matter:** Quality and diverse data lead to better models. Make sure the data is complete, diverse, and error-free.
- **Data Preprocessing Matters:** Proper data preprocessing is important for training models. This includes cleaning, normalising, and converting the data to an appropriate format.
- **Feature Selection Improves Performance:** Selection of relevant features improves model efficiency and accuracy. Use techniques like PCA to reduce dimensionality and focus on informative features.
- **Powerful Deep Learning Algorithms:** Deep learning algorithms are effective for image classification. Use these algorithms to extract and classify features from images.
- **Iterative Refinement Yields Better Results:** Use an iterative refinement process to continuously improve the accuracy and efficiency of the model. Make adjustments to the model architecture, preprocessing steps, or hyperparameters based on the evaluation results.

---

# Bibliography

[1]J. Brownlee, "Machine Learning Mastery With Python: Understand Your Data, Create Accurate Models, and Work Projects End-To-End," 1st ed. Machine Learning Mastery, 2016.

[2]S. Raschka and V. Mirjalili, "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2," 3rd ed. Birmingham, UK: Packt Publishing, 2019.

[3]C. Bishop, "Pattern Recognition and Machine Learning," 1st ed. New York, NY, USA: Springer, 2006.

[4]T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," 2nd ed. New York, NY, USA: Springer, 2009.

[5]A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems," 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.

[6]H. Zhang, "The Optimality of Naive Bayes," in Proc. of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004), Miami Beach, FL, USA, May 2004, pp. 562-567.

[7]"Scikit-learn: Machine Learning in Python," [Online]. Available: <https://scikit-learn.org>. [Accessed: 15-Jun-2024].

[8]"Keras Documentation," [Online]. Available: <https://keras.io>. [Accessed: 20-Jun-2024].

[9]"TensorFlow Documentation," [Online]. Available: <https://www.tensorflow.org>. [Accessed: 25-Jun-2024].

---

[10]"Locust: A Modern Load Testing Framework," [Online]. Available: <https://locust.io>. [Accessed: 10-May-2024].

[11]"Naive Bayes Classifier," [Online]. Available: [https://scikit-learn.org/stable/modules/naive\\_bayes.html](https://scikit-learn.org/stable/modules/naive_bayes.html). [Accessed: 10-Jun-2024].

[12]"Streamlit Documentation," [Online]. Available: <https://docs.streamlit.io>. [Accessed: 10-Jun-2024].

[13]"Introduction to Computer Vision and Image Processing," [Online]. Available: <https://www.coursera.org/learn/introduction-computer-vision>. [Accessed: 25-Jun-2024].

[14]A. S. Riyadi, D. Sweetania, dan P. Sularsih, "Image Processing/Augmentation with Scikit-Image Python Library," DINAMIKA DOTCOM, 31 Januari 2022. [Daring]. Tersedia:



---

# APPENDIX

1. Test Document (In separate file) :

<https://drive.google.com/file/d/13w9X6TnoyZX96s1FxE7S-kvSo2OcJPHP/view?usp=sharing>

2. Presentation Document (In separate file):

[https://drive.google.com/file/d/1yUdYMB9XbZVpQf5z0DbvgRwGboGuU\\_QZ/view?usp=sharing](https://drive.google.com/file/d/1yUdYMB9XbZVpQf5z0DbvgRwGboGuU_QZ/view?usp=sharing)

3. Manual Handbook (In separate file):

[https://drive.google.com/file/d/1oSP2oJi7uw8DgaA\\_OMft5W8JHI7PJ3CT/view?usp=sharing](https://drive.google.com/file/d/1oSP2oJi7uw8DgaA_OMft5W8JHI7PJ3CT/view?usp=sharing)

4. Application Product Source Code (Uploaded on the git hub website):

<https://github.com/Rafsy128/Image-Generative-Prompt-Classification>

5. Product Demo Video (Uploaded on Youtube website):

<https://www.youtube.com/watch?v=IDGvV5Qt7JY>

6. Minutes of Handover (In a separate file):

<https://drive.google.com/drive/folders/1NwQjvS3wrEpGAL5e7bxz-RCHPI8I-xD0?usp=sharing>

7. Poster (In separate file):

<https://drive.google.com/drive/folders/1qI4KSiVjFHmVafoShVNTGSsu6q1xZqn6?usp=sharing>