**PROJECT REPORT**

Skin Saathi

**A mini project submitted to UCER**

**For the degree of:**

Bachelor of Technology

Computer Science and Engineering

**Guided by:** Dr. Vijay Kumar Dwivedi

**Submitted by:**

* Kritika Chaurasia
* Mahima Tripathi

**Content**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Title** | **Page No.** |
| 1. | Abstract | 3 |
| 2. | Introduction | 4 |
| 3. | Objective of the Project | 5 |
| 4. | Benefits | 6 |
| 5. | Hardware/Software Requirements | 7 |
| 6. | Flowchart | 8 |
| 7. | Code | 9-12 |
| 8. | Conclusion | 13 |
| 9. | User Interface | 14-16 |

**1.Abstract**

Skin type classification plays a vital role in dermatology and cosmetic science, guiding individuals toward suitable skincare routines and treatments. This project explores the use of machine learning (ML) techniques to automate skin type analysis using Python.

The project highlights the potential of ML in enhancing dermatological diagnostics and sets the foundation for future developments involving image-based analysis and real-time.

**2.Introduction**

Skin is the largest organ of the human body and serves as a vital barrier between the internal systems and the external environment. Understanding skin type is essential for maintaining skin health, preventing dermatological issues, and selecting appropriate skincare products. Traditionally, skin type classification has been performed manually by dermatologists or cosmetologists through visual inspection and questionnaires. However, these methods are often subjective, time-consuming, and prone to inconsistencies due to individual interpretation and environmental factors.

With the advancement of artificial intelligence and data science, machine learning (ML) has emerged as a powerful tool for automating complex classification tasks. ML algorithms can learn patterns from data and make accurate predictions, offering a more objective and scalable approach to skin type analysis., oil secretion, sensitivity, pore size, and pigmentation levels.

Ultimately, this project demonstrates how machine learning can revolutionize dermatological diagnostics and skincare personalization. It lays the groundwork for future enhancements, including image-based analysis using deep learning and integration into mobile applications for real-time skin assessment.

**3. Objective of the Project**

The primary objective of this project is to build and evaluate various supervised learning models that can accurately predict skin type categories: oily, dry, combination, sensitive, and normal.

Activities involved:

• Automate skin type classification

• Analyze structured skin data

• Implement and compare multiple ML algorithms.

• Apply data preprocessing techniques.

**4. Benefits**

The primary goal of this project is to develop a machine learning-based system using Python that can accurately classify human skin types based on measurable physiological features. By leveraging structured data and supervised learning algorithms, the project aims to create a reliable, scalable, and objective tool for skin type analysis that can assist both dermatologists and consumers in making informed skincare decisions.

Advantages

• Objective Classification: Eliminates subjective judgment by using data-driven predictions.

• Faster Diagnosis: Reduces the time required for skin assessments.

• High Accuracy: Machine learning models can outperform traditional methods in precision and consistency.

• Better Product Recommendations: Helps users select skincare products suited to their specific skin type.

• Clinical Support: Assists dermatologists with a second opinion based on empirical data.

• Tech Integration: Can be adapted into mobile apps or online platforms for real-time skin analysis.

**5. Hardware and Software Requirements**

**Hardware Requirements**

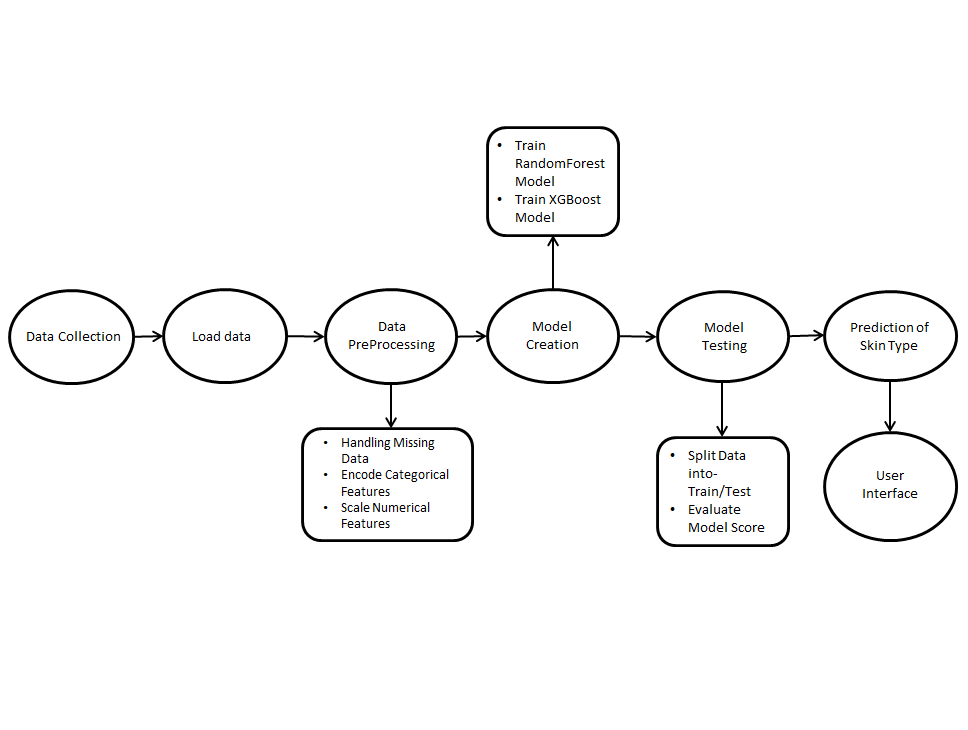
Computer/Laptop with minimum specifications:

* Processor: Intel Core i5 or equivalent (i7 recommended for faster processing)
* RAM: 8 GB (16 GB recommended for handling larger datasets)
* Storage: 256 GB SSD or higher.
* Stable Internet Connection (for downloading libraries, datasets, and documentation)

**Software Requirements**

* Python (version 3.7 or above) – Core programming language for ML implementation
* Python Libraries: (Pandas ,NumPy, scikit-learn)
* Dataset:
* Structured dataset containing skin features and labeled skin types (CSV or Excel format)
* Anaconda – For managing Python environments and packages.

**6. Flow Chart**

****

**7. Code**

train\_model.py

import os

os.chdir("K:\\Kritika\\Internship Project\\AI Skin type Predictor\\skin")

import pandas as pd

from sklearn.ensemble import RandomForestClassifier

from xgboost import XGBClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

import joblib

# Load and preprocess data

df = pd.read\_csv("skin\_data\_100k.csv")

# Identify categorical columns

categorical\_cols = ["gender", "weather", "oiliness", "acne",

"tightness\_after\_wash", "makeup\_usage",

"flaking", "redness\_itchiness"]

# Create encoders for each categorical column

encoders = {}

for col in categorical\_cols:

enc = LabelEncoder()

df[col] = enc.fit\_transform(df[col])

encoders[col] = enc

# Encode target

target\_encoder = LabelEncoder()

y\_encoded = target\_encoder.fit\_transform(df["skin\_type"])

# Separate features and target

X = df.drop("skin\_type", axis=1)

# Scale features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Train/test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X\_scaled, y\_encoded, test\_size=0.2, random\_state=42

)

# Train RandomForest

model = RandomForestClassifier()

model.fit(X\_train, y\_train)

# Train XGBoost

xgb = XGBClassifier()

xgb.fit(X\_train, y\_train)

# Save model, scaler, and encoders

joblib.dump(model, "skin\_model.pkl")

joblib.dump(scaler, "scaler.pkl")

for name, encoder in encoders.items():

joblib.dump(encoder, f"{name}\_encoder.pkl")

joblib.dump(target\_encoder, "target\_encoder.pkl")

# Print scores

print("RandomForest score:", model.score(X\_test,y\_test))

print("XGBoost score:", xgb.score(X\_test,y\_test))

print(pd.Series(model.predict(X\_test)).value\_counts())

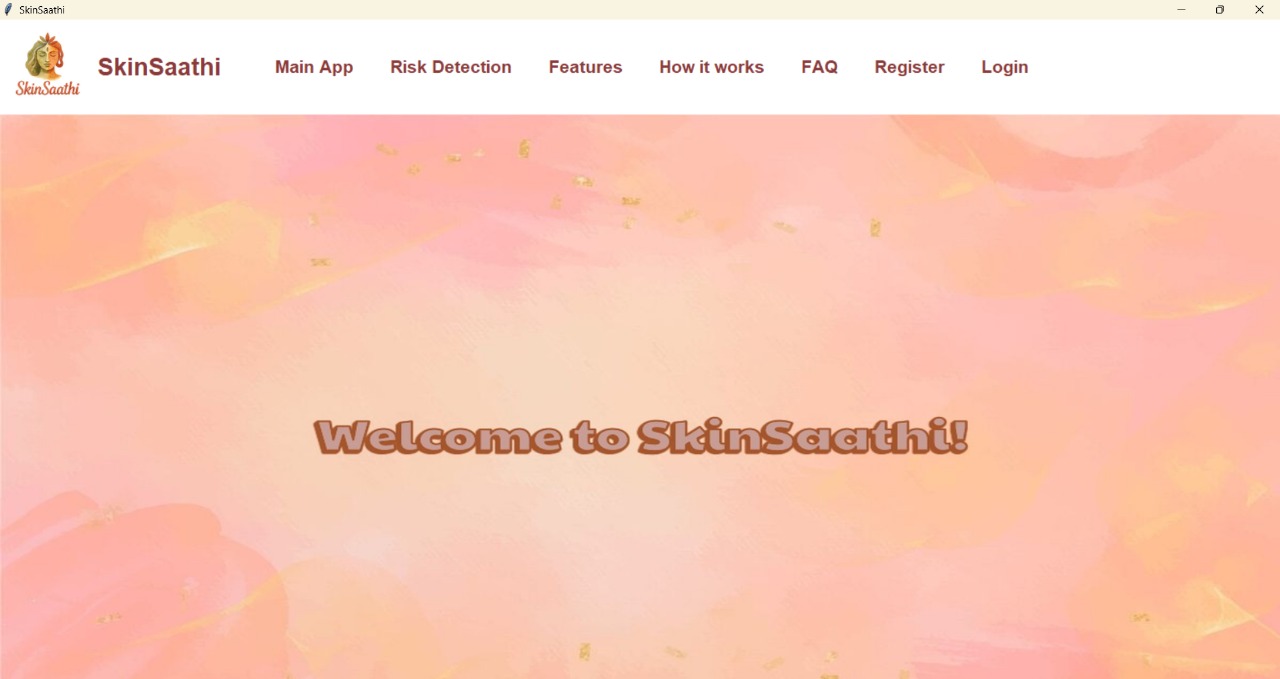
**8. Conclusion**

This project successfully applies machine learning to classify skin types and deliver personalized skincare suggestions. The Python-based system is accurate, scalable, and user-friendly, making skin analysis more accessible.

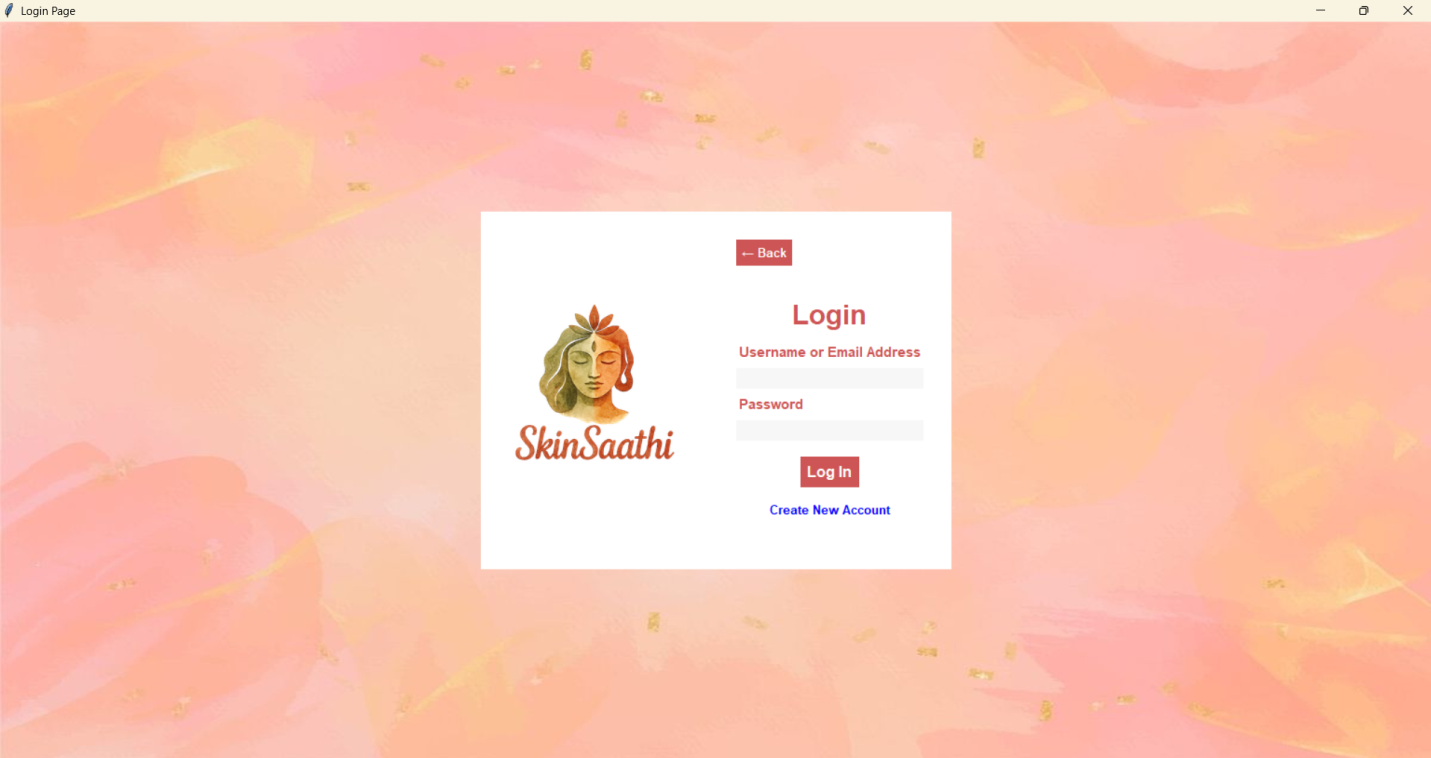
Sincere thanks to our guide Dr.Vijay Kumar Dwivedi Your guidance and encouragement made this project possible.

**9. User Interface**

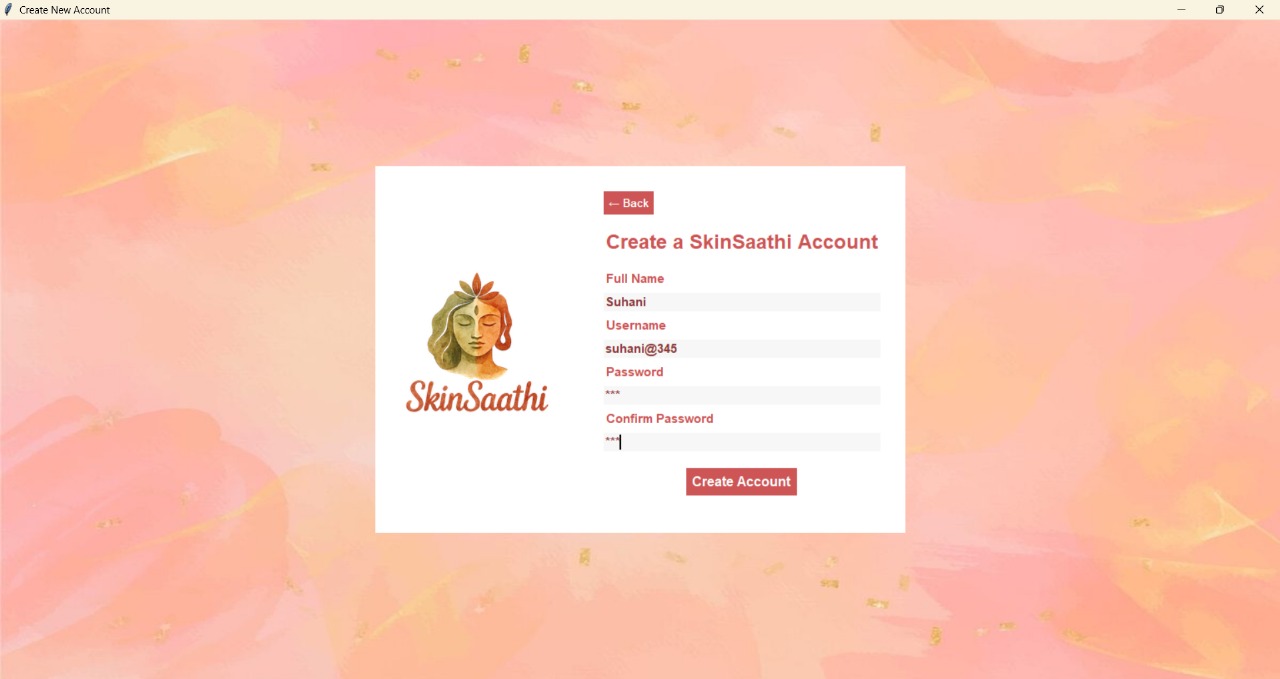
Landing page

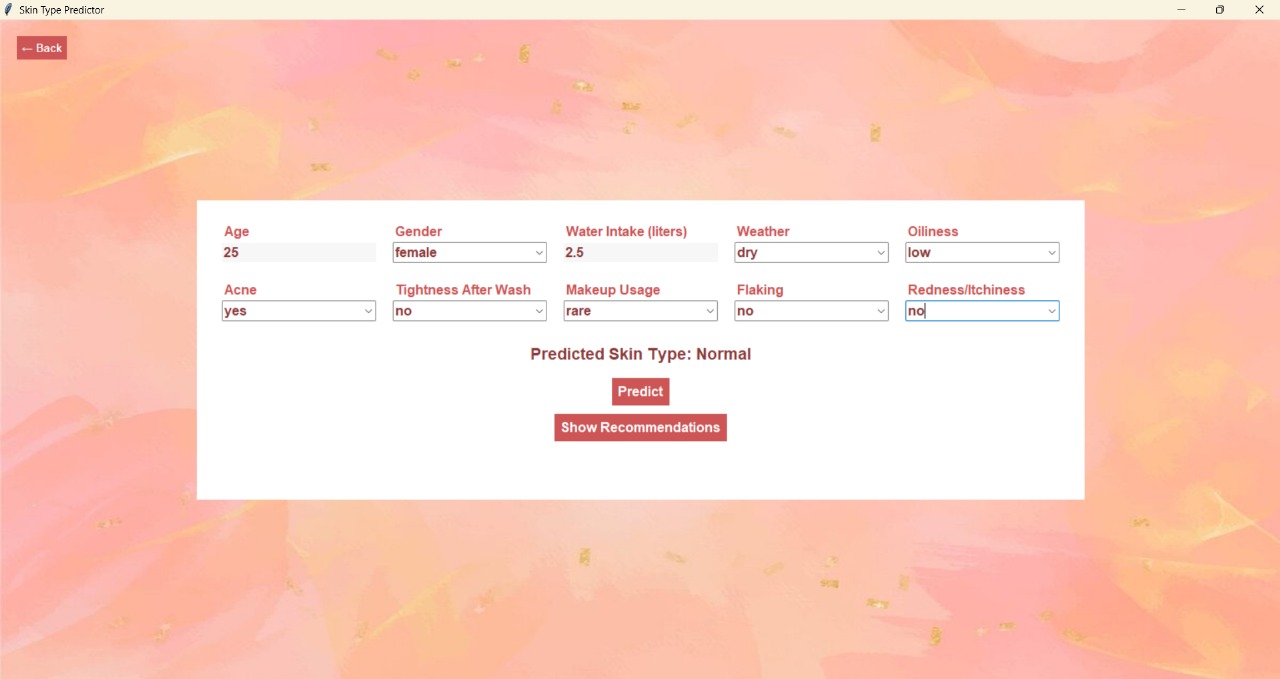
****

Login page

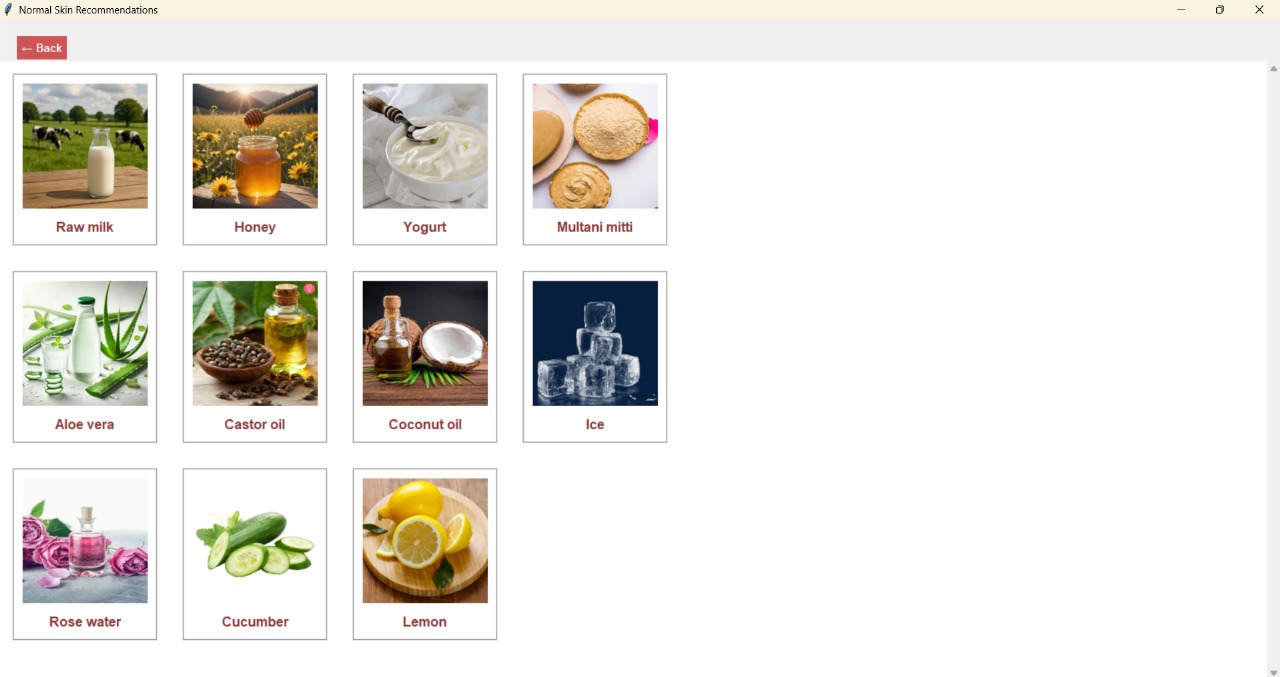
****

Create New Account Page

****

Main App****

Show Recommendation page

****