```python
from fastapi import FastAPI
from pydantic import BaseModel
import uvicorn
import pickle
import streamlit as st
import numpy as np
from sklearn.naive_bayes import GaussianNB
import pandas as pd

app = FastAPI()

class request_body(BaseModel): # takes parameter from user(website)
    fixed_acidity: float
    volatile_acidity: float
    citric_acid: float
    residual_sugar: float
    chlorides: float
    free_sulphur_dioxide: float
    total_sulphur_dioxide: float
    density: float
    ph: float
    sulphates: float
    alcohol: float


df=pd.read_csv("E:\\Machine Learning\\Wine.csv")

X = df.drop("quality", axis=1)
y = df["quality"]

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

clf = GaussianNB()

# you can divide dataset to train and test

clf.fit(X_train, y_train)  # fit - it learns the
```

```
parameter of machine learning

# save model in current directory(file) of the classifier

pickle.dump(clf,open('model.pkl','wb')) # write binary(non-readable)  # dump - save or put it

# load the model from current directory

loaded_model = pickle.load(open('model.pkl', 'rb')) # read binary

from PIL import Image # shows image on browser

def predict_input_page():      ## UI for user input uses streamlit

    img = Image.open("E:\\Machine Learning\\White_wine.jpg")

    st.image(img)

    st.title("Prediction of White Wine Quality ML Algorithm")

    fixed_acidity = st.text_input("Fixed acidity : ")
    volatile_acidity = st.text_input("Volatile acidity : ")
    citric_acid = st.text_input("Citric acid : ")
    residual_sugar = st.text_input("Residual sugar : ")
    chlorides = st.text_input("Chlorides : ")
    free_sulphur_dioxide = st.text_input("Free sulphur dioxide : ")
    total_sulphur_dioxide = st.text_input("Total sulphur dioxide : ")
    density = st.text_input("Density : ")
    ph = st.text_input("ph : ")
    sulphates = st.text_input("Sulphates : ")
    alcohol = st.text_input("Alcohol : ")
```

```python
   ok=st.button("Predict the quality")  # ok has True value when user clicks button

   #try:

   if ok==True:       # if user pressed ok button then True passed

           testdata=np.array([[fixed_acidity, volatile_acidity,citric_acid,residual_sugar,
                        chlorides, free_sulphur_dioxide,
                        total_sulphur_dioxide, density,ph,sulphates,alcohol]])

           classindx = loaded_model.predict(testdata)[0]

           st.header(classindx)

   #except:   # user way of writing error

        #st.info("enter some data")

# how the user will come to the website
@app.post('/predict') # web/gate point of website

def predict(data: request_body):

   # Making the data in a form suitable for prediction

   test_data = [[

       data.fixed_acidity,
       data.volatile_acidity,
       data.citric_acid,
       data.residual_sugar,
       data.chlorides,
       data.free_sulphur_dioxide,
       data.total_sulphur_dioxide,
            data.density,
            data.ph,
            data.sulphates,
            data.alcohol

    ]]

    # Predicting the Class

    class_idx = loaded_model.predict(test_data)[0]

    # Return the Result in form of dictionary

    return {'quality': class_idx}

# main method

if __name__ == "__main__":

    uvicorn.run(app,host="0.0.0.0", port=8000)


```