```
import streamlit as st
import pickle
import time
from datetime import datetime

# Load the sentiment analysis model
model = pickle.load(open('Twitter_sentiment.pkl', 'rb'))

# Initialize session state variables
if 'logged_in' not in st.session_state:
    st.session_state.logged_in = False
if 'positive_tweet_count' not in st.session_state:
    st.session_state.positive_tweet_count = 0
if 'age' not in st.session_state:
    st.session_state.age = None
if 'tweets' not in st.session_state:
    st.session_state.tweets = []
if 'registered_users' not in st.session_state:
    st.session_state.registered_users = []

# Define the registration page
def register_page():
    st.title("Register")
    new_email = st.text_input("Email")
    new_password = st.text_input("Password", type="password")
    confirm_password = st.text_input("Confirm Password",
type="password")
    age = st.number_input("Age", min_value=1, max_value=100, step=1)
    register_button = st.button("Register")

    if register_button:
        if new_password == confirm_password:
            st.session_state.registered_users.append({'email':
new_email, 'password': new_password, 'age': age})
            st.success("Registration successful! Please log in.")
        else:
            st.error("Passwords do not match")

# Define the login page
def login_page():
    st.title("Login")
    email = st.text_input("Email")
    password = st.text_input("Password", type="password")
    login_button = st.button("Login")

    if login_button:
        for user in st.session_state.registered_users:
            if user['email'] == email and user['password'] == password:
                st.success("Login successful!")
                st.session_state.logged_in = True
                st.session_state.age = user['age']
                st.session_state.email = email
                break
        else:
            st.error("Invalid email or password")

# Define the logout functionality
def logout():
    st.session_state.logged_in = False
    st.session_state.age = None
    st.session_state.email = None
    st.session_state.positive_tweet_count = 0
    st.session_state.tweets = []
    st.success("Logged out successfully!")

# Define the sentiment analysis functionality
def sentiment_analysis():
    st.title('Twitter Sentiment Analysis')

    # Display profile icon with positive badge if applicable
    st.sidebar.header("Profile")
    st.sidebar.write("Username:", st.session_state.email)
    if st.session_state.positive_tweet_count > 5:
        st.sidebar.success("🌟 Positive Badge 🌟")

    tweet = st.text_input('Enter your tweet')
    submit = st.button('Predict')

    if submit:
        start = time.time()
        prediction = model.predict([tweet])
        end = time.time()

        st.write('Prediction time taken:', round(end - start, 2),
'seconds')

        if st.session_state.age is not None and st.session_state.age <
18 and prediction[0] == "Negative":
            st.warning("As you are under 18, you can only post positive
tweets.")
        else:
            timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
            st.session_state.tweets.append((tweet, timestamp))
            st.write("Tweet can be posted.")
            if prediction[0] == "Positive":
                st.session_state.positive_tweet_count += 1
            display_tweets()

# Define the function to display tweets with time and date
def display_tweets():
    st.subheader("Posted Tweets")
    for tweet, timestamp in st.session_state.tweets:
        st.write(f"{timestamp} - {tweet}")

# Render appropriate page based on authentication status
if "logged_in" not in st.session_state or not
st.session_state.logged_in:
    if st.session_state.registered_users == []:
        register_page()
    else:
        login_page()
else:
    st.button("Logout", on_click=logout)
    sentiment_analysis()
```