```python
import streamlit as st
import pickle
import pandas as pd
import requests


def fetch_poster(movie_id):
    response = requests.get('https://api.themoviedb.org/3/movie/{}?api_key=8265bd1679663a7ea12ac168da84d2e8&language=en-US'.format(movie_id))
    data = response.json()

    return "https://image.tmdb.org/t/p/w500/" + data['poster_path']


def recommend(movie):
    movie_index = movies[movies['title'] == movie].index[0]
    distances = similarity[movie_index]
    movies_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x: x[1])[1:6]

    recommended_movies = []
    recommended_movies_posters = []
    for i in movies_list:
        movie_id = movies.iloc[i[0]].movie_id
        # fetch poster from API
        recommended_movies.append(movies.iloc[i[0]].title)
        recommended_movies_posters.append(fetch_poster(movie_id))
    return recommended_movies, recommended_movies_posters

similarity = pickle.load(open('similarity.pkl','rb'))
movies_dict = pickle.load(open('movie_dict.pkl','rb'))
movies = pd.DataFrame(movies_dict)

```

```
st.title('Movie Recommender System')

selected_movie_name = st.selectbox(
    "How would you like to be contacted?",
    movies['title'].values)
if st.button("Recommended"):
    names,posters = recommend(selected_movie_name)

    col1,col2,col3,col4,col5 = st.columns(5)
    with col1:
        st.text(names[0])
        st.image(posters[0])
    with col2:
        st.text(names[1])
        st.image(posters[1])
    with col3:
        st.text(names[2])
        st.image(posters[2])
    with col4:
        st.text(names[3])
        st.image(posters[3])
    with col5:
        st.text(names[4])
        st.image(posters[4])
```