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

def fetch_poster(movie_id):
    url = 'http://api.themoviedb.org/3/movie/{}?api_key=66384356adaa881c33368412c3973198&language=en-US'.format(movie_id)
    response = requests.get(url)
    data = response.json()
    return "https://image.tmdb.org/t/p/w500" + data['poster_path']

def recommend(movie):
    try:
        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
            recommended_movies.append(movies.iloc[i[0]].title)
            recommended_movies_posters.append(fetch_poster(movie_id))
        return recommended_movies, recommended_movies_posters

    except IndexError:
        return ["Movie not found in database."], []  # Handle movie not found
    except Exception as e:
        return [f"Error: {str(e)}"], []  # Handle unexpected errors

# Load data
```

```
32 movies_dict = pickle.load(open('movies_dict.pkl', '
   rb'))
33 movies = pd.DataFrame(movies_dict)
34
35 similarity = pickle.load(open('similarity.pkl', 'rb
   '))
36
37 # Streamlit app layout
38 st.title('Movie Recommendation System')
39
40 # Select box for movie titles
41 selected_movie_name = st.selectbox(
42     "Select a movie to get recommendations:",
43     movies['title'].values
44 )
45
46 # Button to get recommendations
47 if st.button('Recommend'):
48     names, posters = recommend(selected_movie_name)
49
50     col1, col2, col3, col4, col5 = st.columns(5)
51     with col1:
52         st.write(f"**{names[0]}**")
53         st.image(posters[0])
54     with col2:
55         st.write(f"**{names[1]}**")
56         st.image(posters[1])
57     with col3:
58         st.write(f"**{names[2]}**")
59         st.image(posters[2])
60     with col4:
61         st.write(f"**{names[3]}**")
62         st.image(posters[3])
63     with col5:
64         st.write(f"**{names[4]}**")
65         st.image(posters[4])
66
67
68
69
70
```

```
71
72
73
74
75
76
77
78
```