

InvestGreenBot

Neodata Mackathania



Memberi del Team



Campisano Salvatore
Stucente del Terzo Anno
Dipartimento di Matematica e Informatica
Corso di Laurea in Informatica Triennale



Campo Filippo
Stucente del Terzo Anno
Dipartimento di Matematica e Informatica
Corso di Laurea in Informatica Triennale

Ispirazione

La nostra ispirazione nasce dalla combinazione di idee innovative su come l'intelligenza artificiale e i chatbot possano essere resi utili, funzionali e intuitivi per l'uso quotidiano. Volevamo creare qualcosa che non solo fosse tecnologicamente avanzato, ma che avesse anche un impatto positivo sulla società e l'ambiente.



Cosa fa

InvestGreenBot è un chatbot specializzato nel settore economico e finanziario, progettato per aiutare gli utenti a prendere decisioni finanziarie informate e sostenibili. Il bot offre consigli su acquisti e gestione finanziaria di vari progetti, tenendo sempre in considerazione la sostenibilità ambientale.



Come l'abbiamo costruito

Abbiamo sviluppato InvestGreenBot utilizzando una combinazione di tecnologie di intelligenza artificiale e API disponibili e gratuite. Il processo ha coinvolto la creazione di algoritmi di machine learning per l'analisi dei dati finanziari e la costruzione di un'interfaccia utente intuitiva che consente un'interazione semplice e diretta con il chatbot.



Sfide in cui abbiamo affrontato

Una delle principali sfide che abbiamo affrontato è stata trovare API gratuite che potessero essere utilizzate senza problemi. Inoltre, abbiamo dovuto decidere se utilizzare un agente AI per migliorare l'interazione con gli utenti. Queste decisioni ci hanno portato a sperimentare diverse soluzioni fino a trovare quella più efficace e produttiva.



Risultati di cui siamo orgogliosi

Siamo orgogliosi di aver creato un chatbot innovativo e utile che non solo aiuta gli utenti nella gestione finanziaria, ma lo fa anche con un occhio di riguardo per l'ambiente. La nostra capacità di superare le sfide tecniche e collaborare efficacemente come team è stata una realizzazione significativa.



Cosa abbiamo imparato

Durante lo sviluppo di InvestGreenBot, abbiamo imparato molto sull'integrazione di API, la creazione di algoritmi di machine learning e l'importanza di un'interfaccia utente intuitiva. Abbiamo anche appreso l'importanza della sostenibilità e di come incorporarla in progetti tecnologici.



Cosa c'è di nuovo per InvestGreenBot

Il prossimo passo per InvestGreenBot è continuare a migliorare le sue funzionalità e ampliare la sua base di utenti. Vogliamo integrare ulteriori API per fornire consigli ancora più precisi e personalizzati e collaborare con organizzazioni che promuovono la sostenibilità ambientale per rendere il nostro chatbot ancora più utile e rilevante.



Cosa abbiamo utilizzato?

- GitHub (Repository)
- Python
- Streamlit
- GroqCloud
- GroqCloudAPI



ChatBot

InvestGreenBot.py > get_groq_response

You, 2 ore fa | 1 author (You)

```
1 import streamlit as st
2 from groq import Groq
3
4 # Funzione per ottenere una risposta dal modello Groq
5 def get_groq_response(messages):
6     client = Groq()
7     completion = client.chat.completions.create(
8         model="llama3-8b-8192",
9         messages=messages,
10         temperature=1,
11         max_tokens=1024,
12         top_p=1,
13         stream=True,
14         stop=None,
15     )
16     response = ""
17     for chunk in completion:
18         if chunk.choices[0].delta.content:
19             response += chunk.choices[0].delta.content
20     return response
21
22 # Funzione principale per eseguire la conversazione con l'assistente
23 def run_conversation():
24     # Inizializza la lista dei messaggi con il messaggio iniziale dell'assistente
25     if "messages" not in st.session_state:
26         st.session_state.messages = [
27             {"role": "assistant", "content": "🌟 Benvenuto! Come posso assisterti oggi nel mercato finanziario?"}
28         ]
29     st.session_state.last_input = ""
```

You, 23 ore fa • Upadte Project



ChatBot

```
30
31 # Visualizza i messaggi della conversazione
32 for message in st.session_state.messages:
33     if message["role"] == "user":
34         st.markdown(f'<div style="padding: 10px; border-radius: 10px; background-color: #d3f5ec; margin: 10px 0;">{message["content"]}</div>')
35     elif message["role"] == "assistant":
36         st.markdown(f'<div style="padding: 10px; border-radius: 10px; background-color: #d6e1f7; margin: 10px 0;">{message["content"]}</div>')
37
38 # Input dell'utente
39 input_message = st.text_input("Scrivi qui il tuo messaggio:")
40 if "input_message" not in st.session_state:
41     st.session_state.input_message = ""
42 if input_message != st.session_state.input_message:
43     st.session_state.input_message = input_message
44
```

```
30
31 # Visualizza i messaggi della conversazione
32 for message in st.session_state.messages:
33     if message["role"] == "user":
34         st.markdown(f'<div style="padding: 10px; border-radius: 10px; background-color: #d3f5ec; margin: 10px 0;">{message["content"]}</div>')
35     elif message["role"] == "assistant":
36         st.markdown(f'<div style="padding: 10px; border-radius: 10px; background-color: #d6e1f7; margin: 10px 0;">{message["content"]}</div>')
37
38 # Input dell'utente
39 input_message = st.text_input("Scrivi qui il tuo messaggio:")
40 if "input_message" not in st.session_state:
41     st.session_state.input_message = ""
42 if input_message != st.session_state.input_message:
43     st.session_state.input_message = input_message
44
```


ChatBot

```
37
38 # Input dell'utente
39 input_message = st.text_input("Scrivi qui il tuo messaggio:")
40 if "input_message" not in st.session_state:
41     st.session_state.input_message = ""
42 if input_message != st.session_state.input_message:
43     st.session_state.input_message = input_message
44
45 # Invia automaticamente la richiesta quando l'utente preme "Enter" sulla tastiera
46 if st.session_state.input_message != st.session_state.last_input:
47     if st.session_state.input_message:
48         st.session_state.messages.append({"role": "user", "content": st.session_state.input_message})
49
50         # Ottenere la risposta dell'assistente dal modello Groq
51         response = get_groq_response(st.session_state.messages)
52         st.session_state.messages.append({"role": "assistant", "content": response})
53
54         # Salva l'ultimo input per evitare cicli infiniti
55         st.session_state.last_input = st.session_state.input_message
56
57         # Pulisce l'input dell'utente dopo l'invio
58         st.session_state.input_message = ""
59
60         # Ripetere l'input dopo l'invio
61         st.experimental_rerun()
62
```

```
def get_groq_response(messages):
    """
    Invia la richiesta al modello Groq e restituisce la risposta.
    """
    # Creazione del client Groq
    client = Groq(api_key=groq_api_key)

    # Creazione della richiesta
    chat_completion = client.chat.completions.create(
        messages=messages,
        model="llama3-70b-8192",
        temperature=0.5,
        max_tokens=1024,
    )

    # Estrazione della risposta
    response = chat_completion.choices[0].message.content

    return response
```






ChatBot

Deploy ⋮

Chat di Assistenza Finanziaria

Benvenuto nella chat di assistenza finanziaria!

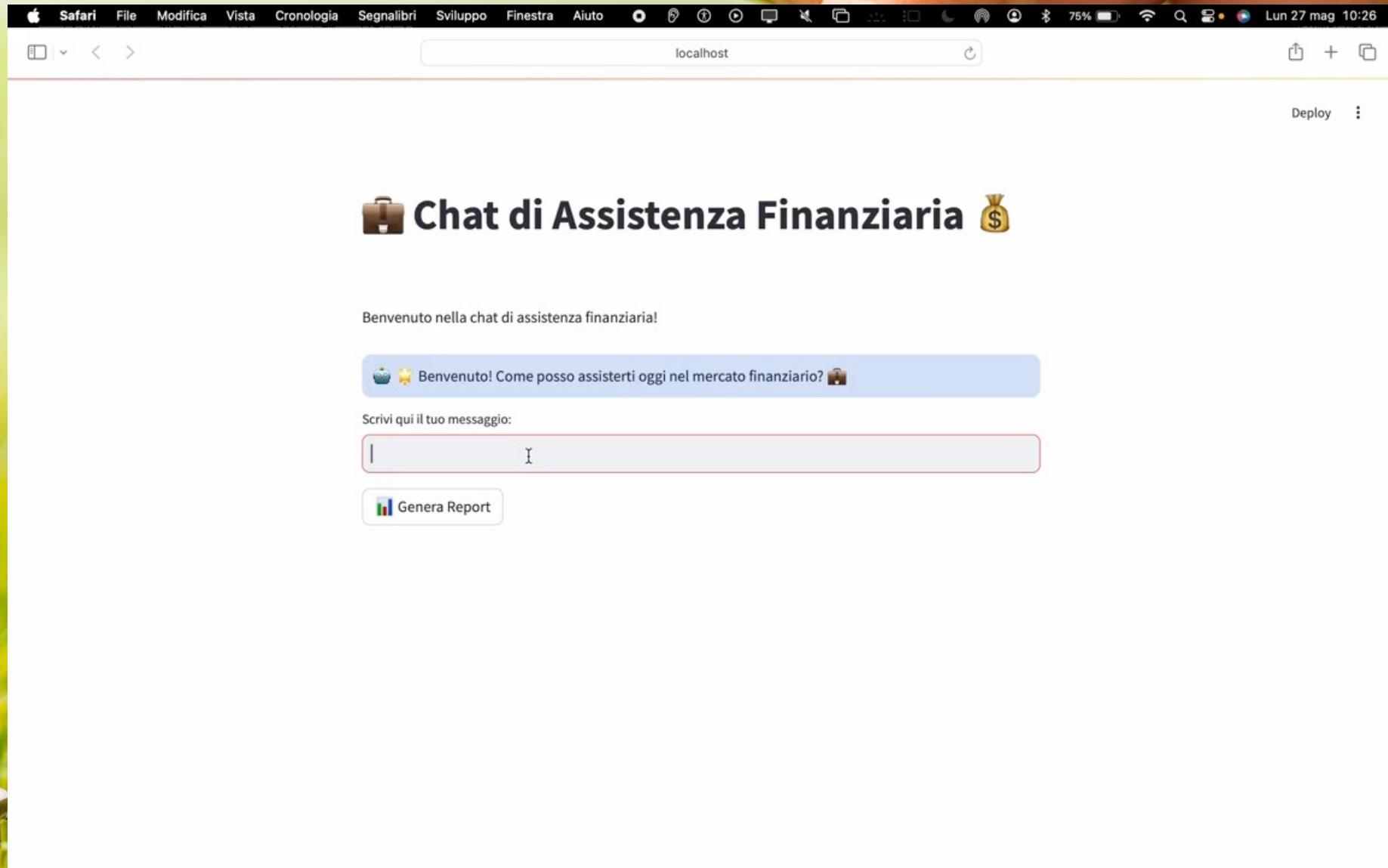
  Benvenuto! Come posso assisterti oggi nel mercato finanziario? 

Scrivi qui il tuo messaggio:

| Press Enter to apply

 Genera Report

Demo



A hand is shown dropping a stack of coins onto a grassy field. Several other stacks of coins are already on the ground, and a small green plant is growing from the grass in the center. The background is a bright, sunny outdoor scene with a clear blue sky and green grass.

“Impara, investi e fai la differenza: InvestGreen è il tuo chatbot per un futuro sostenibile.”