Nana Ama, Keano, ciara, ConstaNtijn

Tech Genius

Heerlen

Persoonlijke AI

Inhoud

[Capture 1](#_Toc162343515)

[Contemplate 2](#_Toc162343516)

[Contract 3](#_Toc162343517)

[Compose 3](#_Toc162343518)

[Chart 4](#_Toc162343519)

[Prototype 5](#_Toc162343520)

[Testresultaten 5](#_Toc162343521)

[Uitgevoerde stappen/knelpunten 5](#_Toc162343522)

# Capture

Beschrijf de requirements en geef een eenduidige definitie van het probleem

**Probleem**: Wij als groepje Tech Genius hebben een nieuwe challenge gekregen: develop een stuk software waarin een koppeling gelegd wordt naar chatgpt en waar op basis van een dataset antwoorden gegeven kunnen worden. Wij moeten developen volgens de 7Cs methode.

**Requirements**: Een software ontwikkelen die een koppeling legt naar Chatgpt en antwoorden genereert op basis van een dataset.

|  |  |  |
| --- | --- | --- |
|  | Requirements | MoSCoW |
| 1. | Het systeem moet in staat zijn om via een API-verbinding te maken met ChatGPT voor het verwerken van natuurlijke taal en het genereren van antwoorden. | MH |
| 2. | De chatbot moet als antwoord zo weinig mogelijke woorden gebruiken om de kosten van de tokens laag te houden. | MH |
| 3. | De chatbot moet snel op de gebruikersinput reageren ongeveer binnen maximaal 5 seconden, waardoor de interactie soepel en sneller is | MH |
| 4. | Chatbot moet klant/gebruikersvriendelijk zijn | MH |
| 5. | Chatbot moet een aantal limiet van woorden hebben, Dit om ervoor te zorgen dat er geen misbruik van gemaakt kan worden om de kosten van de tokens laag te houden | MH |
| 6. | *De chatbot moet in staat zijn om specifieke taken uit te voeren, vragen te beantwoorden en ook op vervolgvragen te reageren.* | MH |
| 7. | De chatbot moet intuïtief zijn, dus gebruikers kunnen makkelijk begrijpen hoe ze met chatbot kunnen communiceren | SH |
| 8. | Implementatie van maatregelen om misbruik van de chatbot te voorkomen, zoals het vermijden van het genereren van schadelijke of ongepaste inhoud/informatie. | SH |
| 9. | *Gebruikers kunnen de mogelijkheid hebben om een persoonlijk profiel aan te maken binnen het systeem. Dit profiel kan basisinformatie bevatten zoals gebruikersnaam, voorkeurstaal, en andere relevante details.* | CH |

# Contemplate

Analyseren van het probleem

**Nodige Data**

* Chatgesprekken of vragen en antwoorden.
* Inputdata voor ChatGPT.

**Oplossingsstrategieën**

* Hoe wordt de koppeling met ChatGPT tot stand gebracht?
* Hoe wordt de dataset verwerkt?
* Vooraf getrainde model gebruiken: vooraf getrainde NLP-modellen zoals ChatGPT 4 om natuurlijke taal te begrijpen en antwoorden te genereren.

**Deelproblemen**

* ChatGPT-integratie.
* Dataset-verwerking.
* API-verbinding
* Modelselectie en -integratie: Het kiezen van het juiste NLP-model en het integreren ervan in de software vereist technische expertise.
* Gebruikersinteractie: Het ontwerpen van een intuïtieve en responsieve gebruikersinterface die het gemakkelijk maakt voor gebruikers om vragen te stellen en antwoorden te ontvangen, is essentieel voor een positieve gebruikerservaring.

# Contract

Bepaal de interface, beschrijf input- en outputdata

* **Input: Chatgesprekken of vragen:**
* De gebruiker zal input verstrekken in de vorm van chatgesprekken of specifieke vragen waarvoor een antwoord wordt gezocht.
* **Output: Antwoorden gegenereerd door ChatGPT.**
* Het systeem geeft als output de antwoorden die zijn gegenereerd door ChatGPT op basis van de ontvangen input.
* **Structuur van de dataset.**
* De dataset die wordt gebruikt voor het trainen van ChatGPT moet gestructureerd zijn, waaronder:
* Vragen: Een kolom met de inhoud van de lesboeken die nodig zijn voor de vragen te stellen dezen moeten per 2 tot 5 bladzijdes worden toegevoegd.
* Antwoorden: Een kolom met bijbehorende antwoorden op de gestelde vragen.
* Beschrijf de parameters voor de ChatGPT API.

# Compose

Deel het probleem eventueel op in deelproblemen:

* Deelprobleem 1: Integratie met ChatGPT.

1. Identificatie van data:

* Invoer: Chat Chatgesprekken of vragen die gebruikers stellen.
* Uitvoer: Antwoorden gegenereerd door ChatGPT

1. Oplossingsstrategie:

* **Koppeling met ChatGPT**: Maak gebruik van de OpenAI API voor ChatGPT.
* **Stappen**:

1. Ontvang chatbericht
2. Verzend het chatbericht naar de open AI API
3. Ontvang en verwerk het antwoord van de API.
4. Toon het antwoord aan de gebruiker.

* Deelprobleem 2: Dataset-verwerking.
  1. Identificatie van data:
     + Invoer: Dataset met inhoud van lesboeken
     + Uitvoer: Verwerkte dataset voor training.
  2. Oplossingsstrategie:
* Verwerkingsstappen:

1. Lees de dataset in.
2. Splits de dataset in vragen en antwoorden.
3. Bereid de dataset voor training voor.

# Chart

Gebruik besturingsstructuren om de volgorde van instructies te beschrijven:

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

# Prototype

Codeer de besturing in een programmeertaal:

* Implementeer de stappen in deelprobleem 1 en deelprobleem 2 volgens het stroomschema.
* Gebruik de benodigde bibliotheken om met ChatGPT te communiceren.

# Testresultaten

Test de code volgens de requirements:

* Voer tests uit met verschillende chatgesprekken en datasets.
* Zorg ervoor dat de antwoorden logisch en relevant zijn.
* Terug naar stap 1 indien nodig, bijvoorbeeld als de resultaten niet aan de requirements voldoen.

# Uitgevoerde stappen/knelpunten

Stappen die wij als een team uitgevoerd hebben om de chatbot te bouwen en ook uitleg van problemen en oplossingen.

**Uitgevoerd stappen – Problemen en oplossingen**

* Documentatie (PvA)

Wij begonnen eerst met onze documentatie, dus wij hebben eigenlijk onze planning vastgesteld, hoe wij dit project aanpak met behulp van de 7c stappen van programmeren. Houd in dat wij stap voor stap nodige hulpmiddel documenteert om dit document uit te voeren bijvoorbeeld analyseren tot requirement vaststellen. Hierdoor weten wij als groep wat te doen als alles wat wij nodig hebben voor dit project

* Mock ups van chatbot

Na documentatie zijn wij doorgegaan met het ontwerpen van de chatbot.

Probleem: Wij waren verschillende mock ups aan het maken omdat wij geen ontwerp konden kiezen vanwege dat iedereen in de groep een andere esthetiek heeft.

Oplossingen: Onze Mock ups/ ontwerp van de chatbot aan andere groepsgenoten laten zien voor zijn mening om te vergelijken met die van ons team zelf maar, uiteindelijk hadden wij een keuze gemaakt door alles die wij samen mooi vond aan de verschillende ontwerpen te combineren in een

* Coderen/ontwerpen van chatbot

Na het ontwerpen van de Chatbot met Canva zijn wij begonnen met het realiseren van het chatbot dus met HTML en CSS coderen voor webapplicatie, ook hebben wij javascript toegevoegd voor speciale functies die nodig is in de chatbot.

Problemen: De gemaakte Mock up ontwerp in Canva in de website te implementeren

Oplossingen: Hulp aan chatgpt gevraagd, dus het analyseren van onze codes om te kijken waar de fout was en de reden waarom het niet dezelfde uitzag als onze mockup

* API Key verbonden met chatbot.

Vervolgens was het tijd om de API KEY van Openai te verbinden met de chatbot toen het ontwerp klaar was, zodat je een conversatie met het chatbot kan uitvoeren of dat je reactie terugkrijgt als je een vraag stuurt.

Belangrijke Stappen

1. Instaleer python of zorg ervoor dat je het hebt
2. Installeer the python OpenAI library in je terminal
3. Maak een python file in je gekozen IDE-omgeving

Stap 1: Verkrijg een OpenAI API-sleutel

* Ga naar de OpenAI-website en maak een account aan of log in als je er al een hebt.
* Navigeer naar de API-sectie van je account.
* Genereer een nieuwe API-sleutel als je er nog geen hebt.
* Kopieer de API-sleutel naar het klembord.

Stap 2: Voeg de API-sleutel toe aan het bestand

* Open het python file waarin je de OpenAI API-sleutel wilt gebruiken.
* Zoek naar de regel waar openai.api\_key wordt ingesteld en plak daar de API-sleutel die je hebt gekopieerd.
* Sla je file op. (CTRL+S)

Stap 3: Voer de Flask-applicatie uit

* Zorg ervoor dat je alle benodigde Python-bibliotheken hebt geïnstalleerd. Je kunt ze installeren met pip install openai.
* Navigeer naar de map waarin je python file zich bevindt.
* Open een terminal of opdrachtprompt in die map.
* Voer het commando python naam\_van\_het\_script.py uit, waarbij naam\_van\_het\_script.py de naam is van je bestand/file [Bijv: test.py].
* Als alles correct is geconfigureerd, zou de file moeten worden gestart en kun je deze openen in je webbrowser.
* Pdf-link in de chatbot toevoegen

1. Verzamel de PDF-links: Zorg ervoor dat je de URLs hebt van de PDF-bestanden die je wilt delen via de chatbot. Deze links kunnen naar PDF-bestanden op je eigen server of naar externe bronnen verwijzen.
2. Identificeer het berichttype: Bepaal welk type bericht je wilt gebruiken om de PDF-link weer te geven. Dit kan een bericht zijn van de chatbot zelf of van de gebruiker.
3. Formaat van het bericht: Zorg ervoor dat het bericht duidelijk aangeeft dat het om een PDF-link gaat. Dit kan worden gedaan door de tekst van het bericht expliciet te maken of door een speciaal symbool of pictogram toe te voegen.
4. Plaatsing van de link: Voeg de PDF-link toe aan het bericht op een manier die gebruikers kunnen herkennen en erop kunnen klikken. Dit kan door simpelweg de URL weer te geven of door een klikbare hyperlink te maken.

* Testen

We zijn vervolgens aan de slag gegaan met het uittesten van de chatbot om te kijken of alles ook daadwerkelijk functioneert. We hebben de chatbot verschillende vragen gesteld en gekeken of die ook op verschillende manieren kan reageren.

* Hij kan in verschillende talen antwoorden
* Hij kan bestanden uploaden
* Hij is mobile responsive

Problemen: Hij kan het bestand wel lezen maar er niet op antwoorden

* Data van chatbot bewaren.

Om gegevens in een chatbot te bewaren, kun je verschillende benaderingen gebruiken, afhankelijk van de behoeften en de complexiteit van de chatbot. Hier zijn voorbeelden van een paar methodes:

* Gebruik van een database: Een van de meest gebruikelijke methoden is het opslaan van gegevens in een database. Dit kan een relationele database zoals MySQL, PostgreSQL, of een NoSQL-database zoals MongoDB zijn. Je kunt de database gebruiken om gebruikersprofielen, gespreksgeschiedenis, voorkeuren en andere relevante informatie op te slaan.
* Gebruik van bestanden: Voor eenvoudigere toepassingen kun je gegevens opslaan in platte tekstbestanden of JSON-bestanden. Dit kan handig zijn voor het opslaan van configuratiegegevens, eenvoudige logs of tijdelijke gegevens.
* Gebruik van sessieopslag: Als je alleen tijdelijke gegevens tijdens een sessie moet bewaren, kun je gebruikmaken van sessieopslag in het geheugen. Dit kan worden bereikt met behulp van sessievariabelen in webapplicaties of met behulp van ingebouwde functies van chatbot-frameworks.
* Cloudopslag: Maak gebruik van cloudopslagdiensten zoals AWS S3, Google Cloud Storage of Azure Blob Storage om bestanden op te slaan die worden gedeeld of gebruikt door de chatbot.
* Gebruik van externe API's: Als je gegevens moet opslaan die elders worden beheerd, zoals gebruikersprofielen op sociale-mediaplatforms, kun je gebruikmaken van de API's van die platforms om toegang te krijgen tot en gegevens op te halen of bij te werken.