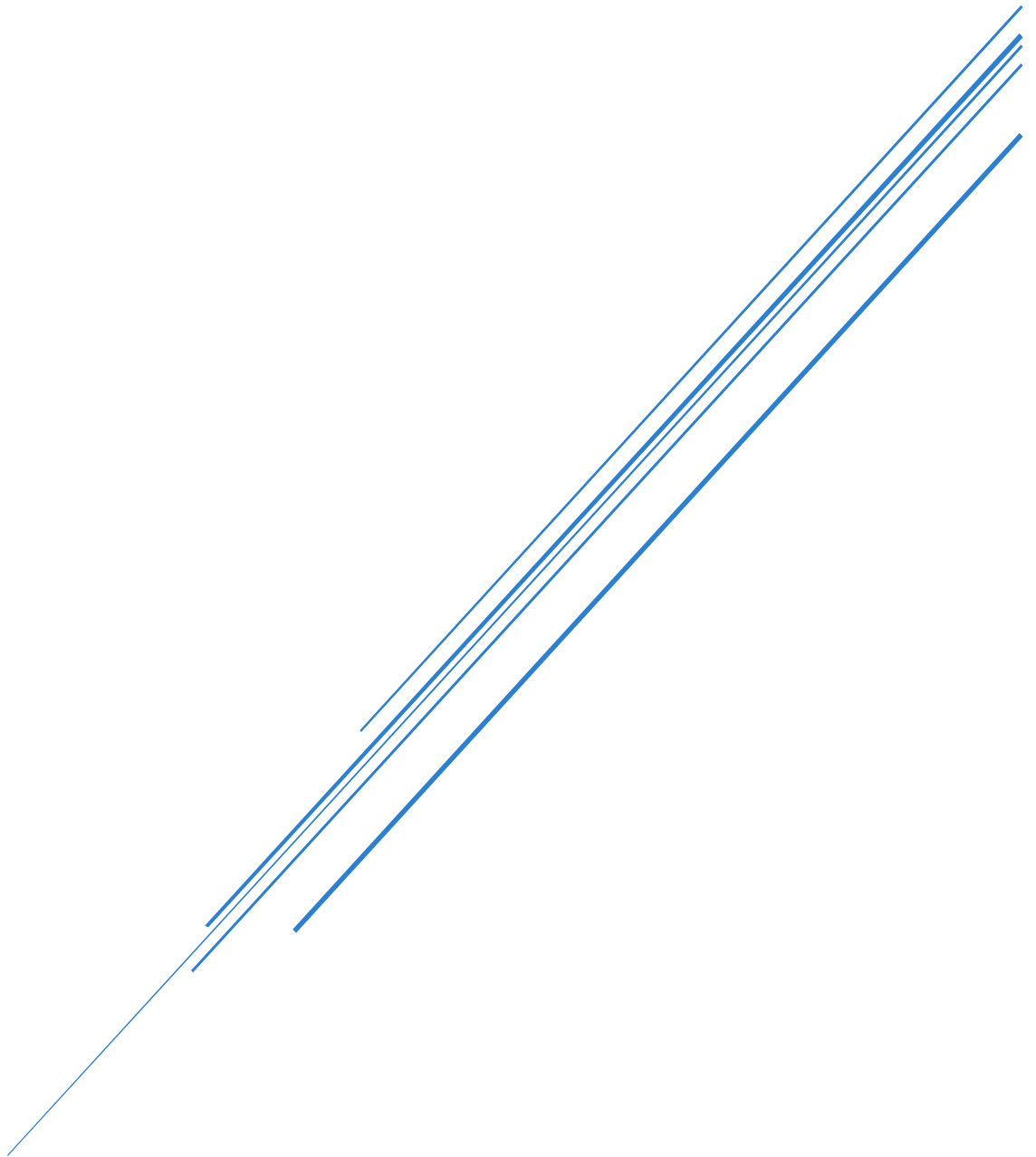


PROJECT “WACHTWOORD CHECKER EN -GENERATOR”



Glinshi Biantama Glinshi Setiowati
HICOH03Q

Inhoud

Inleiding	2
Plan	3
Code wachtwoord checker	4
Functioneel Ontwerp wachtwoord checker	6
Technisch Ontwerp wachtwoord checker.....	7
User Stories wachtwoord checker	8
Code wachtwoord generator	9
Functioneel ontwerp wachtwoord generator	10
Technisch ontwerp wachtwoord generator	11
User Stories wachtwoord generator.....	12
Evaluatie	13

Inleiding

Het verslag focust zich op de ontwikkeling van een gebruikersinterface voor een wachtwoordchecker en -generator. In dit verslag zal ik de functies van deze tool uitleggen, zoals het controleren van de sterkte van wachtwoorden en het genereren van nieuwe veilige wachtwoorden. De gebruikersinterface zal ik ontwerpen met een focus op gebruikersvriendelijkheid, om gebruikers te helpen bij het beheren van hun wachtwoorden.

In het verslag kunnen lezers een gedetailleerde beschrijving verwachten van hoe de wachtwoordchecker en -generator zal werken en hoe die in elkaar zitten. Ook zal ik in het verslag mijn plannen bespreken en zal ik mijn ervaringen delen die ik heb opgedaan tijdens het ontwikkelen van wachtwoordchecker en -generator.

Plan

Ik ben van plan om de wachtwoordchecker en -generator alleen te maken, zodat ik op mijn eigen tempo kan werken. Ik ben benieuwd of het op deze manier in mijn voordeel zal werken, aangezien ik meestal goed werk onder druk en die druk dan minder aanwezig zal zijn (de druk om samen op bepaalde afgesproken tijden aan dit project te werken).

In mijn vrije tijd zal ik werken aan dit project, uiteraard ook in de les. Elke zondag zal ik 1 à 2 uurtjes besteden aan dit vak en elke donderdag in de les een anderhalf uur. Dit zal ik doen vanaf 5 februari tot 14 april. De zondagen zal ik besteden aan het maken van de wachtwoordchecker en de donderdagen aan het maken van de generator. Zodra ik ze eerder af heb begin ik aan het verslag, zo niet begin ik met het schrijven van het verslag op 28 maart.

Ik zal veel YouTube filmpjes kijken, google gebruiken en de bestaande Python stof gebruiken die ik eerder heb gemaakt in de vorige periode. Ik geloof dat deze hulpmiddelen goed voor mij zullen werken om de wachtwoordchecker en -generator te maken. Chat GPT probeer ik zoveel mogelijk te vermijden, omdat ik het idee heb dat ik hiervan niet leer wat kan resulteren in dat ik het programmeren niet snap.

Om mijn werkzaamheden bij te houden en te organiseren zal ik een planning maken op elke zondag, zodat ik weet wat en wanneer ik iets af wil hebben.

Ik zal mijn gemaakte werk opslaan gewoon op mijn computer. Word en Visual Studio Code slaan gemaakt werk automatisch op, dus ik hoef niet bang te zijn dat gemaakt werk, verloren zal gaan.

Code wachtwoord checker

```
import re #Importeer de reguliere expressie module.

def wachtwoord_checker(vraag):
    #Controleer of de lengte van het wachtwoord minder is dan 8.
    lengte_error = len(vraag) < 8

    #Controleer of er geen cijfers in het wachtwoord zitten.
    cijfer_error = not re.search(r"\d", vraag)

    #Controleer of er geen hoofdletters in het wachtwoord zitten.
    hoofdletter_error = not re.search("[A-Z]", vraag)

    #Controleer of er geen kleine letters in het wachtwoord zitten.
    kleineletter_error = not re.search("[a-z]", vraag)

    #Controleer of er geen speciale tekens in het wachtwoord zitten.
    speciale_tekens_error = not re.search("[!@#$$%^&*()_+=]", vraag)

    #Maak een lijst van de verschillende fouten.
    error = [cijfer_error, lengte_error, hoofdletter_error,
kleineletter_error, speciale_tekens_error]

    #Tel het aantal fouten.
    all_error = sum(error)

    #Geef feedback op basis van het aantal fouten.
    if all_error == 0:
        return "Sterk wachtwoord"#Als er geen fouten zijn, is het een sterk
wachtwoord.
    elif all_error <= 2:
        return "Matig wachtwoord"#Als er minder dan of gelijk aan 2 fouten
zijn, is het een matig wachtwoord.
    else:
        tips = []
        #Voeg specifieke tips toe op basis van de aanwezige fouten.
        if cijfer_error:
            tips.append("voeg nummers toe")
        if lengte_error:
            tips.append("vergroot de wachtwoordlengte")
        if hoofdletter_error:
            tips.append("voeg hoofdletters toe")
        if kleineletter_error:
            tips.append("voeg kleine letters toe")
        if speciale_tekens_error:
            tips.append("voeg symbolen toe")
```

```
        return "Zwak wachtwoord. Tips: " + ", ".join(tips) #Geef een zwak  
wachtwoord en specifieke tips terug.  
  
#Vraag de gebruiker om een wachtwoord in te voeren.  
vraag = input("Vul een wachtwoord in: ")  
  
#Roep de wachtwoord_checker functie aan en print het resultaat.  
print(wachtwoord_checker(vraag))
```

Functioneel Ontwerp wachtwoord checker

De wachtwoord checker moet wachtwoorden controleren op sterkte. Het moet de gebruiker vragen om een wachtwoord in te voeren en vervolgens feedback geven over de sterkte ervan. Deze feedback moet aangeven of het wachtwoord sterk, matig of zwak is, en eventuele specifieke tips voor verbetering geven.

De wachtwoord checker werkt door het ontvangen van een wachtwoord als input. Vervolgens controleert hij of het wachtwoord niet voldoet aan verschillende punten, zoals lengte, aanwezigheid van cijfers, hoofdletters, kleine letters en speciale tekens. Op basis van deze controle geeft het programma feedback over de sterkte van het wachtwoord en eventuele tips voor verbetering.

Wanneer de gebruiker het programma gebruikt, zal die eerst worden gevraagd om een wachtwoord in te voeren. Na het invoeren van het wachtwoord zal de wachtwoordchecker meteen feedback geven over de sterkte van het wachtwoord en eventuele tips voor verbetering weergeven. De gebruiker zal deze feedback zien op het scherm en kan vervolgens beslissen om het ingevoerde wachtwoord hetzelfde te houden of te veranderen op basis van de ontvangen tips.

Technisch Ontwerp wachtwoord checker

Programmeertaal

Voor dit project wordt Python gebruikt als programmeertaal. Python is gekozen vanwege school, momenteel leren we nu om te programmeren in Python en is de mogelijkheid om in andere programmeertalen te programmeren aanwezig maar niet handig.

Versiebeheer

Voor versiebeheer wordt Git gebruikt in combinatie met een online platform voor code-hosting, zoals GitHub. Het project zal worden aangemaakt als een Git-repository. Alle wijzigingen en updates aan de code zullen worden gevolgd en vastgelegd met Git-commentberichten. Er zullen regelmatig comments worden gedaan om de voortgang van het project vast te leggen en om een volledige geschiedenis van wijzigingen bij te houden. Door gebruik te maken van Git en een online platform voor code-hosting, kunnen teamleden samenwerken, wijzigingen beoordelen en feedback geven en kan de code veilig worden bewaard en gedeeld met anderen.

User Stories wachtwoord checker

Gebruiker: Wilt zijn/haar wachtwoord controleren om te zien of het veilig genoeg is met behulp van de wachtwoordchecker.

Stappen:

- *Het programma vraagt de gebruiker om een wachtwoord in te voeren.
- *De gebruiker voert een wachtwoord in.
- *De gebruiker klikt op de knop(enter)om het wachtwoord te controleren.
- *De uitslag van de sterkte van het wachtwoord wordt aan de gebruiker getoond.

Acties wachtwoord checker:

Als de gebruiker het programma start, wordt een prompt weergegeven waarin wordt gevraagd om een wachtwoord in te voeren.

Als het wachtwoord:

- *Minstens 8 tekens lang is,
- *Ten minste één cijfer bevat.
- *Ten minste één hoofdletter bevat,
- *Ten minste één kleine letter bevat,
- *Ten minste één speciaal teken bevat, dan wordt het wachtwoord als "Sterk" gezien en krijgt de gebruiker een bevestiging van een sterk wachtwoord.

Als het wachtwoord niet aan een van deze punten voldoet:

- *Wordt het wachtwoord als "Zwak" gezien.
- *Worden de ontbrekende criteria aangegeven, zoals "Wachtwoord moet minstens 8 tekens lang zijn" of "Voeg minstens één cijfer toe".

Als het wachtwoord niet aan deze punten voldoet kan die het wachtwoord aanpassen en opnieuw controleren.

Code wachtwoord generator

```
import re #Importeer de regular expressions module.
from random import randrange #Importeer de randrange-functie uit de random-
module.

def wachtwoord_checker():
    vraag_cijfers = int(input("Geef een wachtwoord lengte op: ")) #Vraag de
gebruiker om de lengte van het wachtwoord.

    #Genereer een wachtwoord van de opgegeven lengte met willekeurige
karakters.
    password = ''.join([chr(randrange(33,127)) for x in range(vraag_cijfers)])

    #Print het gegenereerde wachtwoord.
    print("Wachtwoord is:", password)

#Roep de functie aan om een wachtwoord te genereren en te printen.
wachtwoord_checker()
```

Functioneel ontwerp wachtwoord generator

Het programma moet een wachtwoord genereren van een door de gebruiker opgegeven lengte en dit wachtwoord dan tonen.

De wachtwoord generator werkt door het ontvangen van een wachtwoordlengte als input. Het programma genereert vervolgens een wachtwoord van de opgegeven lengte met willekeurige karakters. Het gegenereerde wachtwoord wordt geprint en aan de gebruiker getoond.

Wanneer de gebruiker het programma gebruikt, zal die eerst worden gevraagd om een wachtwoord lengte in te voeren. Na het invoeren van de lengte, zal het programma een wachtwoord genereren die voldoet aan bepaalde criteria en dit aan de gebruiker tonen. De gebruiker heeft de mogelijkheid om het gegenereerde wachtwoord te kopiëren of te gebruiken zoals gewenst.

Technisch ontwerp wachtwoord generator

Programmeertaal

Voor dit project wordt Python gebruikt als programmeertaal. Python is gekozen vanwege school, momenteel leren we nu om te programmeren in Python en is de mogelijkheid om in andere programmeertalen te programmeren aanwezig maar niet handig.

Versiebeheer

Voor versiebeheer wordt Git gebruikt in combinatie met een online platform voor code-hosting, zoals GitHub. Het project zal worden aangemaakt als een Git-repository. Alle wijzigingen en updates aan de code zullen worden gevolgd en vastgelegd met Git-commentberichten. Er zullen regelmatig comments worden gedaan om de voortgang van het project vast te leggen en om een volledige geschiedenis van wijzigingen bij te houden. Door gebruik te maken van Git en een online platform voor code-hosting, kunnen teamleden samenwerken, wijzigingen beoordelen en feedback geven, en kan de code veilig worden bewaard en gedeeld met anderen.

User Stories wachtwoord generator

Gebruiker: Wilt een wachtwoord kunnen genereren met een opgegeven lengte, zodat die een sterk en willekeurig wachtwoord kan gebruiken voor diverse gelegenheden en niet snel gehackt zal worden.

Stappen:

- *Het programma vraagt de gebruiker om een gewenste lengte voor het wachtwoord in te voeren.
- *De gebruiker voert een getal in voor de gewenste lengte van het wachtwoord.
- *Het programma genereert een wachtwoord met de opgegeven lengte en willekeurige tekens.
- *Het gegenereerde wachtwoord wordt aan de gebruiker getoond

Acties wachtwoord generator:

- *Als de gebruiker het programma start, wordt een prompt weergegeven waarin wordt gevraagd om de gewenste lengte voor het wachtwoord in te voeren.
- *Als de gebruiker iets anders dan een getal invoert, geeft het programma een foutmelding en vraagt die opnieuw om een getal in te voeren.
- *Het programma genereert een wachtwoord dat bestaat uit een reeks willekeurige karakters van de opgegeven lengte. De karakters kunnen alfanumeriek zijn en kunnen symbolen bevatten.
- *Na het genereren van het wachtwoord, toont het programma het aan de gebruiker.

Als het wachtwoord is gegenereerd kan de gebruiker die kopiëren en gebruiken.

Evaluatie

Ik vond het verassend leuk om te werken aan dit project. Eerst keek ik er minder naar uit om onderzoek te doen naar hoe ik de wachtwoordchecker en -generator überhaupt kon maken, alleen toen ik eenmaal bezig was en het begon te begrijpen begon ik het leuk te vinden. Eerst schreef ik uit wat ik wilde in de codes, waarna ik stapsgewijs opschreef wat mij logisch leek wat er moest gebeuren om de codes succesvol te laten werken. Na dit gedaan te hebben begon ik met mijn onderzoek: eerst kijken naar de door school gegeven stof en daarna het bekijken van Youtube filmpjes. Als ik hier niet uitkwam, ging ik naar de docent toe voor advies. Op een gegeven moment raakte ik in de knoop met mijn code, op aanraden van de docent ben ik uiteindelijk Chat GBT gaan gebruiken. Ik ben erachter gekomen dat dit heel handig is. Eerst vond ik Chat GBT een tool, die mensen luier maakte. Nu zie ik dat Chat GBT erg goede uitleg kan geven en voor vrijwel alles ook goede tips. Uiteindelijk kon ik bij Chat GBT terecht voor vragen en advies, waar de docent ook zelf niet de antwoorden van wist. Voor de rest vond ik dit een leerzame opdracht en ben ik wijzer geworden in het maken van codes.