



Programiranje v Pythonu

Generator naključnih gesel

Maturitetna seminarska naloga

Avtor: Luka Zupan 4.C

Mentor: Helena Starc Grlj

Šolsko leto: 2022/2023

Povzetek

Za maturitetno seminarsko nalogo sem naredil generator naključnih gesel v programskem jeziku python in ga vizualiziral s pomočjo tkinter GUI, Pythonovega grafičnega vmesnika.

Program je namenjen generiranju varnih in težko vdrljivih gesel, ki uporabniku omogočajo večjo varnost na spletu. Ko program zaženemo, izberemo število znakov in program generira geslo s podanim številom znakov. V programu nato kliknemo na gumb, da kopiramo geslo in ga lahko uporabimo za nek uporabniški račun.

Naredil sem tudi spletno stran, kjer sem predstavil svoj projekt in objavil vse dokumente v povezavi s projektom.

Abstract

I made a random password generator for my graduation project in informatics. I programmed it using Python programming language. My goal with this project is to make web a safer place for everybody. I want to make my program user friendly so everyone can use it, with as very little to no knowledge of computers.

Ključne besede

Python, naključno, varnost, tkinter GUI, generator, geslo, Visual Studi Code

KAZALO

1	UVOD.....	5
1.1	PREDSTAVITEV PROBLEMA.....	5
1.2	UPORABLJENA STROJNA OPREMA	5
1.3	UPORABLJENA PROGRAMSKA OPREMA.....	5
2	TEORETIČNI DEL	6
2.1	SKUPINE ZNAKOV	6
2.2	MODULI	6
2.3	NAKLJUČNA IZBIRA ZNAKOV.....	6
2.4	TKINTER OKNO.....	6
2.5	IDEJNI IZGLED PROGRAMA	7
2.6	DIAGRAM POTEKA	8
3	PRAKTIČNI DEL	9
3.1	OPIS REŠITVE IN KODA PROBLEMA	9
3.2	NAVODILA ZA UPORABO	13
3.3	PREDSTAVITEV NALOGE NA SPLETU.....	15
3.4	PORABA ČASA	16
4	ZAKLJUČEK.....	17
4.1	MOŽNE IZBOLJŠAVE.....	17
5	VIRI	18

KAZALO SLIK, GRAFOV IN TABEL

Slika 1: Idejna slika	7
Slika 2: Moduli	9
Slika 3: Skupine znakov	9
Slika 4: Tkinter okno	9
Slika 5: Label password length	9
Slika 6: Dolžina gesla	10
Slika 7: Polje password	10
Slika 8: Izbira znakov	10
Slika 9: If stavek-simboli in številke	11
Slika 10: Random choice	12
Slika 11: Generate button	12
Slika 12: Funkcija copy password	12
Slika 13: Modul pyperclip	13
Slika 14: Zaženi program	13
Slika 15: Program	13
Slika 16: Spreminjanje dolžine gesla	14
Slika 17: Izbiranje znakov	14
Slika 18: Generirano gesla	14
Slika 19: Spletna stran	15
Tabela 1: Strojna oprema	5
Tabela 2: Programska oprema	5
Graf 1: Časovna razporeditev	16

STVARNO KAZALO

funkcija, 6, 11, 13
 generate, 4, 6, 9, 10, 11, 12, 13, 15
 geslo, 2, 5, 6, 9, 10, 12, 15
 GUI, 2, 5, 6
 import, 9
 length, 4, 6, 8, 10, 11, 12
 modul, 6, 9, 14, 18
 python, 2, 5, 14
 random, 2, 6, 9, 12
 tkinter, 2, 5, 6, 9, 12, 17

1 UVOD

1.1 PREDSTAVITEV PROBLEMA

V vsakdanjem življenju se vse več srečujemo z uporabniškimi profili in gesli za vsako stvar, ki jo želimo početi na internetu. Večina ljudi za uporabniške profile uporablja enaka gesla. Zaradi tega postanejo lahke tarče za cyber napade. Sam sem to spoznal na težek način, saj so mi vdrli v 5 uporabniških računov naenkrat. Z slabimi izkušnjami na internetu sem se odločil narediti generator naključnih gesel v programskem jeziku python. Z generatorjem bom imel za vsak uporabniški račun naključno geslo, kar bo rešilo problem vdiranja v uporabniške račune.

Osnove pythona smo obdelali pri urah informatike v šoli, tisto kar bom potreboval pri tem projektu pa sem se naučil še dodatnega znanja pythona, ki mi bo pomagal pri študiju in nadaljevanju v smeri programiranja. Ker ne vem veliko o tkinter GUI ukazih, si bom pri težavah pomagal s spletno stranjo [stack-overflow](https://stackoverflow.com). Za programiranje bom uporabljal program Microsoft Visual Studi Code.

Cilj, ki sem si ga zadal je, da ustvarim čim bolj uporaben program, ki je prijazen uporabniku in je namenjen za splošno uporabo. Program bo v pomoč ljudem, da ostanejo varni na spletu in bo opozarjal ljudi na pomembnost varnega gesla.

1.2 UPORABLJENA STROJNA OPREMA

Za nalogo sem uporabil le svoj namizni računalnik.

Tabela 1: Strojna oprema

Procesor	Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz
RAM	8.00 GB
Grafična kartica	Radeon RX 570 Series 8GB
Zaslon	24.1 inch

1.3 UPORABLJENA PROGRAMSKA OPREMA

Za sam projekt sem uporabljal urejevalnik kode Visual Studio Code in celotno kodo napisal s tem programom. Tudi spletno stran sem naredil v tem programu. Dokumentacijo sem pisal v Microsoft Wordu 2019.

Tabela 2: Programska oprema

Operacijski sistem	Windows 11 Home
Programski jezik	Python 3.10
Urejevalnik	Visual Studio Code, Pycharm Community E.
Urejevalnik besedila	Microsoft Word 2018
Grafični vmesnik	Tkinter GUI
FTP prenos datotek na strežnik	Fillezilla
Spletni brskalnik	Google Chrome

2 TEORETIČNI DEL

2.1 SKUPINE ZNAKOV

Pri generatorju naključnih gesel je prvo potrebno navesti vse znake iz katerih bo kasneje računalnik izbral nekaj naključnih in jih premešal med sabo. Znake bom razdelil na več skupin:

- Uppercase_char
- Lowercase_char
- Symbols
- Numbers

Programu bom dodal funkcijo tega, da lahko uporabnik izbere, katere od skupin želi uporabiti. Če v geslu nočemo znakov, bo geslo izbrano le iz števil in črk. To bom naredil s pomočjo pythonovega modula tkinter GUI, s katerim bom ustvaril okno, v katerem se bo prikazala koda, gumb za kopiranje te kode in možnost izbiranja med znaki, iz katerih uporabnik hoče, da je generirano njegovo geslo.

2.2 MODULI

Za ta projekt bom moral v svoj pythonov program vnesti kar nekaj modulov: modul tkinter GUI, modul random ter modul pyperclip. S *tkinter GUI* bo program zagnan v oknu, da ga bo uporabnik lažje uporabljal. Uporabnik bo v oknu izbral določene možnosti, ki jih bo potem uporabila funkcija `generate_password` s pomočjo ukaza `entry.get`. Modul random bom potreboval v funkciji `generate_password`, da bo iz določene skupine znakov program naključno generiral niz z določeno dolžino. Modul pyperclip pa bo uporaben, ker hočem programu dodati možnost, da uporabnik kopira generirano kodo in jo takoj uporabi pri nekem uporabniškem računu.

2.3 NAKLJUČNA IZBIRA ZNAKOV

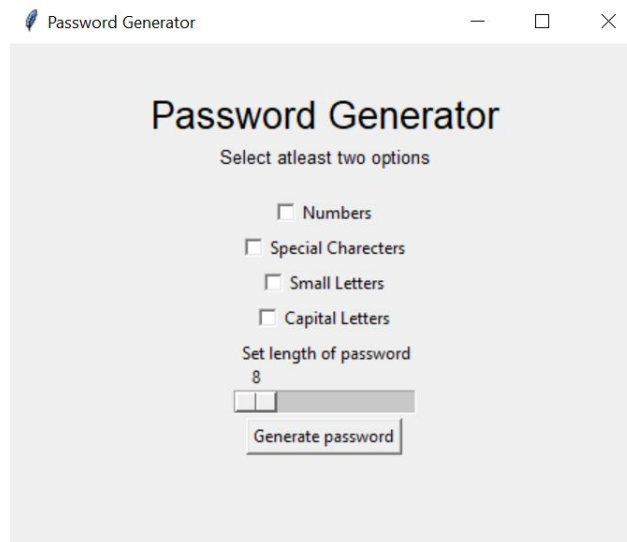
Iz skupine znakov chars, ki jo bo uporabnik lahko prilagodil po svoji želji, bo program naključno izbral naključne znake v niz z določeno dolžino, ki bo zapisana v spremenljivki `length`. Da bo program izbral naključne znake, bom uporabil ukaz `random.choice`, ukaz modula `random`. Največjo dolžino kode bom nastavil na 24, saj je ta največkrat uporabljena. Če bo uporabnik zapisal kodo večjo od 24, bo dolžina enaka 24.

2.4 TKINTER OKNO

Ko bo uporabnik zagnal program, se bo odprlo okno, kjer bo uporabnik lahko uporabljal vse funkcije programa. Program bom naredil s tkinterjem, ki ima drugačen nabor ukazov kot Python brez dodanih modulov. Programirati v Tkinterju se bom naučil s pomočjo YouTube in drugih spletnih strani.

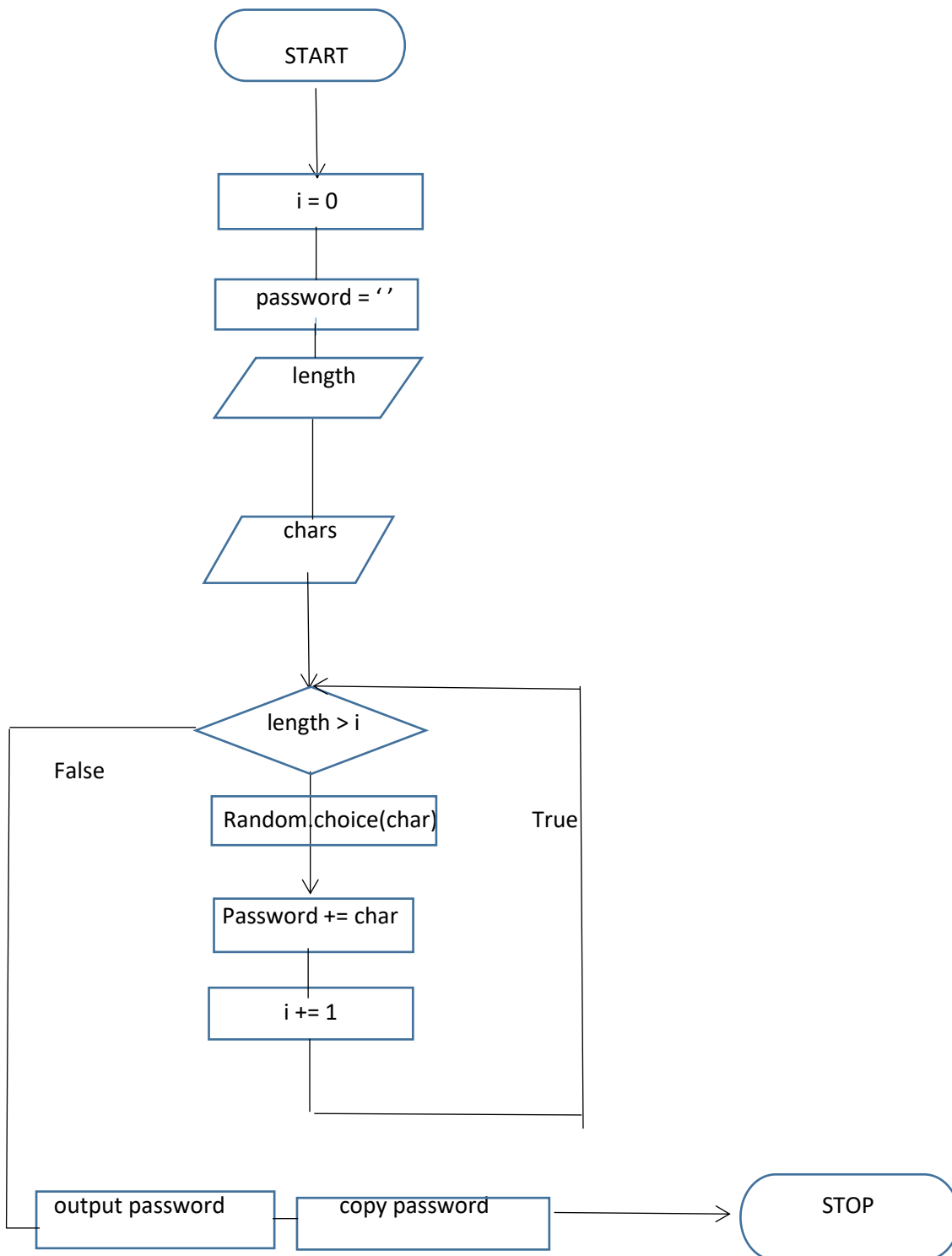
2.5 IDEJNI IZGLED PROGRAMAS

Mojo idejo za program sem vzel iz spodnje slike. Program bom naredil na podoben način kot je prikazan na sliki spodaj:



Slika 1: Idejna slika

2.6 DIAGRAM POTEKA



Zgoraj sem narisal idejni diagram poteka mojega programa.

3 PRAKTIČNI DEL

3.1 OPIS REŠITVE IN KODA PROBLEMA

Svoji kodi sem najprej dodal knjižnico tkinter. Da sem si delo olajšal, sem to naredil z ukazom **from tkinter import ***. S tem sem si poenostavil nadaljno programiranje, saj mi pri vsakem ukazu iz modula tkinter ni bilo treba pisati tkinter.

```
from tkinter import *
import random
import pyperclip
```

Slika 2: Moduli

Dodal sem tudi modul random z ukazom **import random**, da sem kasneje lahko uporabil funkcijo random za izbiro naključnih znakov. Modul pyperclip sem dodal z ukazom **import pyperclip**. Ta modul sem kasneje potreboval, saj ima določene ukaze, ki sem jih potreboval, da uporabnik mojega programa lahko kopira geslo.

Nato sem znake razdelil v skupine in jih napisal v posamezni niz. Vsak niz sem poimenoval po določenem vzorcu: **uppercase_chars**, **lowercase_chars**, **number_chars** in **symbol_chars**. Nize sem zapisal v taki obliki:

```
uppercase_chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
lowercase_chars = "abcdefghijklmnopqrstuvwxyz"
symbol_chars = "!@#$%&='?';.\|/|~,>*( )<"
number_chars = "0123456789"
```

Slika 3: Skupine znakov

Preden sem zapisal funkcijo generate_password, sem najprej uporabil ukaza **root** in **root.geometry**. Z ukazom **root** se odpre okno, ko uporabnik zažene program, zato sem ukaza zapisal pred celotno kodo. Z ukazom **root.geometry** sem definiriral velikost okna, ki se odpre, ko zaženemo program.

```
root = Tk()
root.geometry("400x300+100+100")
```

Slika 4: Tkinter okno

Oknu sem nato dodal še ostale elemente. Ukazi v tkinterju so bili nekaj novega zame, zato sem se jih moral naučiti. So na nek način zelo podobni, ampak se uporabljajo drugače. Najprej sem z ukazom **label** dodal posamezni tekst in nato uporabil še ukaz **label.pack()**, s katerim določimo kje se element pokaže in kolikšno velikost zavzame.

```
length_pass = Label(root, text = "Password length:" )
length_pass.pack()
```

Slika 5: Label password length

Vse ostale napise sem dodal na enak način, kot je prikazano na sliki zgoraj. Tekstu je sledilo polje, v katerega uporabnik napiše željeno dolžino gesla. Uporabil sem class **entry**, definiral širino polja in kolikšna je največja dolžina gesla, ki jo lahko uporabnik vpiše v polje. To sem naredil z ukazom **length_entry.insert()**. Dolžino sem nastavil med številoma 1 in 24, saj se mi zdi, da za pomembne uporabniške račune, kot so paypal, spletne banke in e-pošte geslo ne more biti predolgo. Na koncu sem polje ponovno omejil z ukazom **length_entry.pack()**.

```
length_entry = Entry(root,width = 5)
length_entry.insert(1,24)
length_entry.pack()
```

Slika 6: Dolžina gesla

Naslednje, kar me je čakalo je bilo polje, v katerem se izpiše naključno generirano geslo, ko mu določimo dolžino in znake. Tega sem se ponovno lotil s classom **entry**, ki je namenjen vsem poljem, v katere nekaj vpišemo ali pa se v polju nekaj izpiše. Določil sem širino polja in ga z ukazom **justify** premaknil na sredino okna. Polje sem omejil z ukazom **password_field.pack()**.

```
password_field = Entry(root, width=24, justify='center')
password_field.pack()
```

Slika 7: Polje password

Zdaj sem potreboval še način, kako bo uporabnik izbral znake, ki jih hoče uporabiti v svojem geslu. Na koncu sem se odločil, da je najlažje, če uporabnik obkljuka vse znake, ki jih hoče uporabiti. Naredil sem dva checkboxa, enega za številke in enega za simbole. Tukaj sem imel malo več težav, saj mi checkbox sprva ni hotel delovati. Pomagal sem si s spletno stranjo [stackoverflow](https://stackoverflow.com), da sem dodal class boolean obema checkboxoma (če checkbox obkljukaš je vrednost true, drugače pa false). Z ukazom **BooleanVar()** sem rešil problem. Za checkbox sem uporabil ukaz **Checkbutton**, ter mu dodal spremenljivko.

```
#checkbox za uporabo števk
include_numbers = BooleanVar()
numbers_checkbox = Checkbutton(root, text="Include Numbers", variable=include_numbers)
numbers_checkbox.pack()

#checkbox za uporabo simbolov
include_symbols = BooleanVar()
symbols_checkbox = Checkbutton(root, text="Include Symbols", variable=include_symbols)
symbols_checkbox.pack()
```

Slika 8: Izbira znakov

Na zgornji sliki je prikazana koda, ki sem jo uporabil, da sem ustvaril delujoči checkbox. Tako sem končal z vizualnim delom projekta. Zdaj sem potreboval še dve funkciji, **generate_password** in **copy_password**, da se geslo v oknu prikaže in ga uporabnik lahko kopira.

V funkciji `generate_password` sem naredil novo spremenljivko `chars`, kateri sem dodal male in velike črke. Nato sem ustvaril novo spremenljivko `length`, ki sem jo povezal z `length_entry`. Pri tem sem imel več težav, saj podatek o dolžini, ki ga uporabnik vpiše v polje, dobimo z ukazom **`entry.get`** in ne moremo spremenljivke le enačiti z `length_entry`. Tako izgleda celotna funkcija:

Spremenljivki `length` sem vrsto spremenil v integer, saj drugače številko, ki jo uporabnik napiše v polje vzame kot string, kar povzroči napako v funkciji.

```
def generate_password():
    chars = uppercase_chars + lowercase_chars
    length = int((length_entry.get()))
    if length > n_length:
        length = n_length

    if include_numbers.get():
        chars += number_chars
    if include_symbols.get():
        chars += symbol_chars

    password = ''.join(random.choice(chars)for _ in range(length))
    password_field.delete(0,END) # pobriše polje od zacetka do konca
    password_field.insert(0, password) # prikaze kodo v polju

generate_button = Button(root,text="Generate Password",command=generate_password)
generate_button.pack()
```

Slika 8: Funkcija `generate_password`

V nadaljevanju sem naredil if stavek, kjer če uporabnik vpiše vrednost večjo od maksimalne dolžine(nastavljena je na 24), dolžino gesla program nastavi na 24.

Nato sem naredil še dva if stavka za uporabo simbolov in števil. Ponovno sem uporabil ukaz **`.get`**, a tokrat v if stavku. Če iz checkboxa dobimo odziv, sepravi če uporabnik checkbox obkljuka, potem v spremenljivko `chars` funkcija doda to skupino znakov.

```
if include_numbers.get():
    chars += number_chars
if include_symbols.get():
    chars += symbol_chars
```

Slika 9: If stavek-simboli in številke

Nato sem moral kodi dodati še to, da naključno izbere znak iz skupine chars in vse znake doda skupaj. Najprej sem ustvaril prazen niz in ga poimenoval **password**. Nato sem uporabil ukaz **.join**, ki združi vse naključno izbrane znake v en niz. Znake sem naključno izbral z ukazom **random.choice(chars)**. Uporabil sem zanko for, ampak na drugačen način, kot sem ga bil vajen, saj for zanka v mojem primeru nima števca. Uporabil sem **for _ in range()**, kjer podčrtaj pomeni, da ne potrebujemo števca. Zanka se zažene tolikokrat, dokler ne doseže vrednosti length. Če length nebi spremenil v integer, se tukaj pojavi težava, ko program zaženemo.

```
password = ''.join(random.choice(chars)for _ in range(length))
password_field.delete(0,END) # pobriše polje od zacetka do konca
password_field.insert(0, password) # prikaze kodo v polju
```

Slika 10: Random choice

Ko program zaženemo, lahko večkrat generiramo kodo. Da to lahko deluje, se mora polje vsakič ponovno pobrisati. Ko uporabnik ponovno pritisne na gumb generate password, se celotno polje pobriše in se v polju pokaže novo geslo. Uporabil sem ukaza **.delete** in **.insert**. Ukaz delete pobriše polje od 0 do konca niza, insert pa v polju prikaze niz.

Zdaj sem moral le še povezati funkcijo in pa gumb generate_password. To sem naredil z ukazom **Button** iz modula tkinter, ki ima parameter command. Za ta parameter sem nastavil funkcijo generate_password, kar pomeni, da vsakič, ko uporabnik klikne na gumb, zažene funkcijo, ki pobriše polje in v polju prikaže nov niz.

```
generate_button = Button(root,text="Generate Password",command=generate_password)
generate_button.pack()
```

Slika 11: Generate button

Potreboval sem le še funkcijo, ki kopira geslo in jo povezati z gumbom copy password. V tej funkciji sem uporabil ukaz **.get**, ki iz polja password vzame geslo. Če v polju obstaja geslo, ga s pomočjo modula pyperclip, z ukazom **pyperclip.copy()** kopira.

```
def copy_password():
    password = password_field.get()
    if password:
        pyperclip.copy(password)
    copy_button = Button(root,text="Copy Password",command=copy_password)
    copy_button.pack()
```

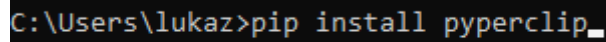
Slika 12: Funkcija copy password

Funkcijo sem z gumbom povezal na isti način, kot sem povezal funkcijo generate_password, ampak sem za command nastavil funkcijo copy_password. Ko uporabnik pritisne na gumb copy, se zažene funkcija copy_password.

Na koncu kode sem potreboval še ukaz **root.mainloop()**, ki pusti okno odprto, dokler ga uporabnik ne zapre in s tem program ustavi.

3.2 NAVODILA ZA UPORABO

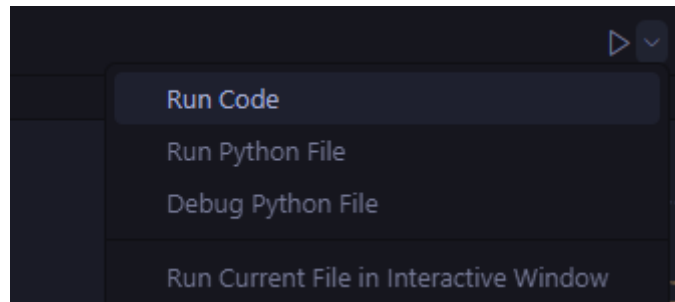
Uporabnik si program naloži in ga zažene. Za program je potrebno, da si uporabnik naloži modul pyperclip in python. Če uporabnik modula nima naloženega, mora odpreti cmd in napisati komando ***pip install pyperclip***.



```
C:\Users\lukaz>pip install pyperclip_
```

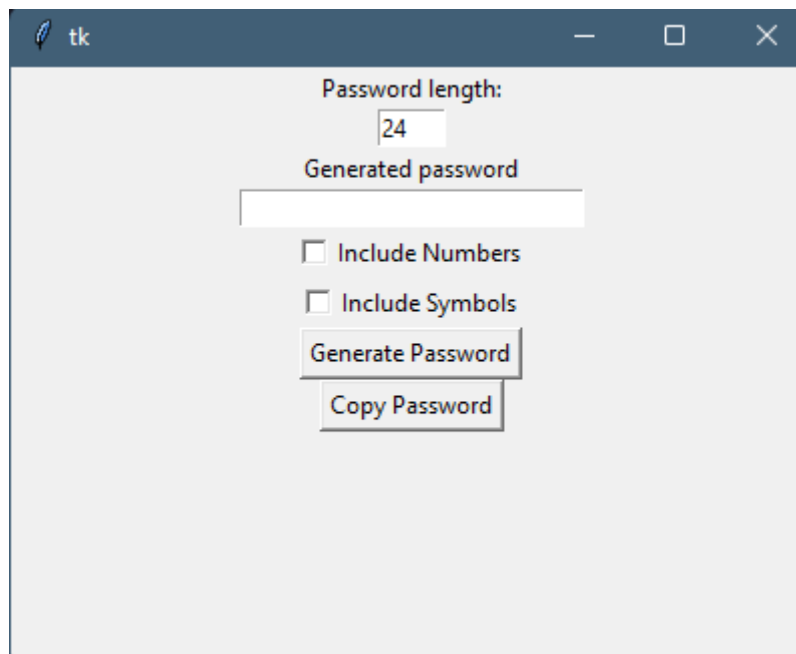
Slika 13: Modul pyperclip

Ko uporabnik naloži modul pyperclip, lahko program uporablja. Program lahko odpre v kateremkoli python editorju.



Slika 14: Zaženi program

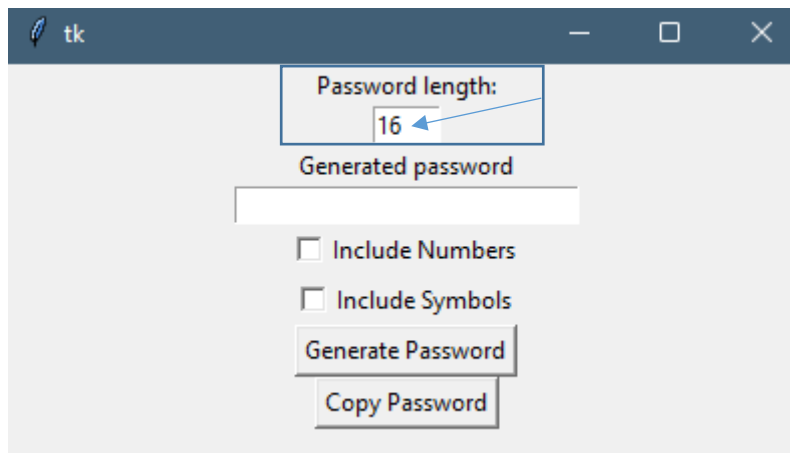
Jaz sem za programiranje uporabljal Visual Studio code, ampak načeloma za vse python editorje izgleda podobno zgornji sliki.



Slika 15: Program

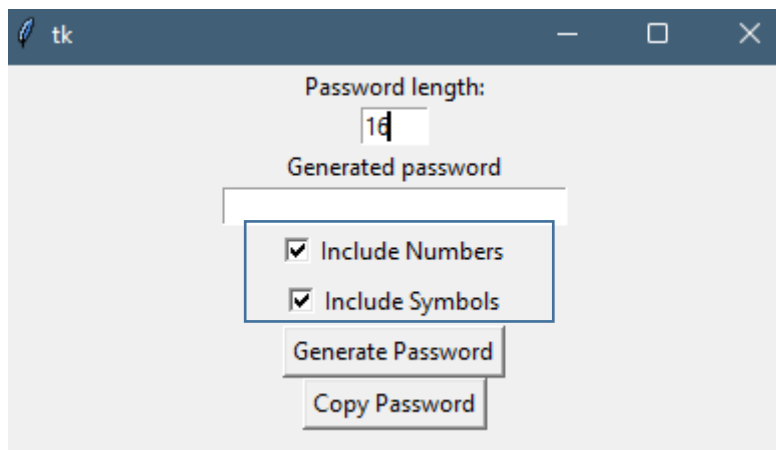
Ko uporabnik zažene program se odpre to okno.

Če hoče uporabnik spreminjati dolžino kode, želeno dolžino vpiše v to polje:



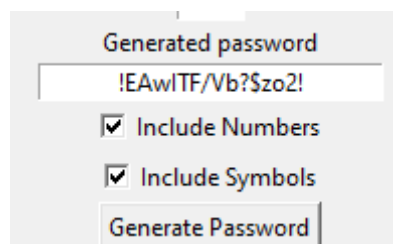
Slika 16: Spreminjanje dolžine gesla

Nato lahko obkljuka številke in simbole, če jih hoče uporabiti v svojem geslu:



Slika 17: Izbiranje znakov

Ko uporabnik nastavi vse po svojih preferencah, pritisne na gumb generate in geslo se prikaže v polju:

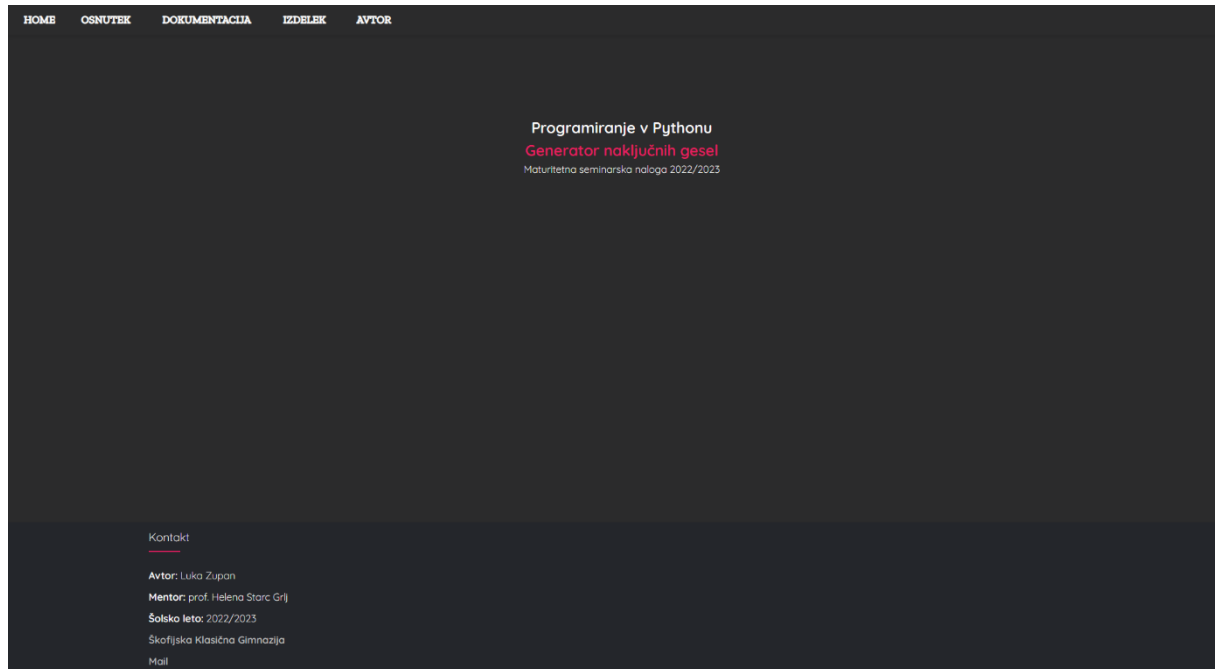


Slika 18: Generirano geslo

Geslo nato uporabnik kopira in ga uporabi, kot geslo za poljubni uporabniški račun in ostane varen na spletu.

3.3 PREDSTAVITEV NALOGE NA SPLETU

Celotno nalogo in vse datoteke v povezavi z nalogo sem predstavil tudi na spletu. Naredil sem spletno stran v programskem jeziku Html in jo oblikoval z jezikom CSS. Spletno stran sem naredil v programu Visual Studio Code in jo s pomočjo programa Fillezilla naložil na splet.



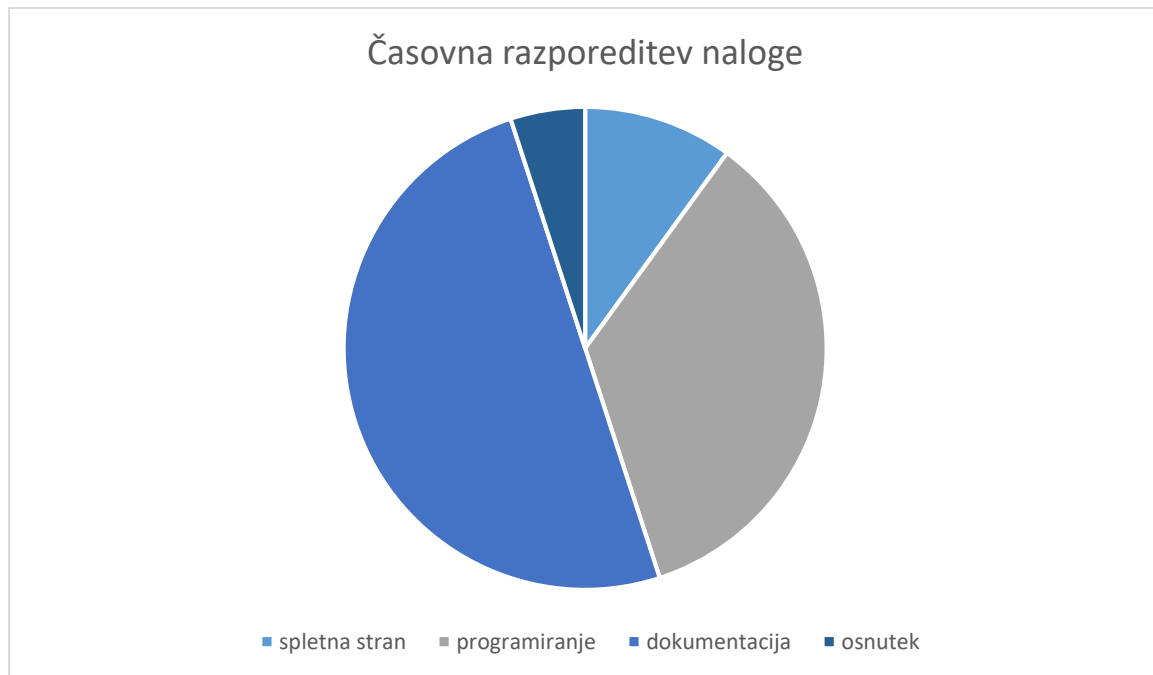
Slika 19: Spletna stran

Na zgornji sliki je začetna stran moje celotne spletne strani.

3.4 PORABA ČASA

Pri seminarski nalogi sta mi največ časa vzela učenje programiranja s tkinterjem in pisanje dokumentacije. Veliko časa sem potreboval, da sem lahko podatke, ki jih uporabnik vpiše v tkinter okno prenesel v samo funkcijo. Ta del mi je predstavljal največjo težavo pri sami nalogi. Dokumentacija mi je prav tako vzela veliko časa, saj sem dokumentiral posamezne dele celotne kode in dodajal še slike zaslona.

Izdelava same spletne strani mi je tudi vzela nek del časa, ampak pri tem nisem imel težav.



Graf 1: Časovna razporeditev

4 ZAKLJUČEK

Zadovoljen sem s svojim končnim izdelkom. Mislim, da sem dosegel svoja pričakovanja in cilje, ki sem si jih zadal. Izdelek, ki sem ga izdelal je uporaben in mislim, da bo prav prišel tudi drugim uporabnikom, ki hočejo ostati varni na spletu. Menim, da sem s svojim projektom vsaj malo pripomogel k temu, da bo manj ljudi imelo težave z vdori v uporabniške račune.

4.1 MOŽNE IZBOLJŠAVE

Mislim, da je moj program uporaben in prijazen uporabnikom. Mogoče bi lahko spremenil to, da lahko uporabnik program uporablja, brez da naloži modul pyperclip, kar mi na žalost ni uspelo.

5 VIRI

- Python (online) [Uporabljeno 1. 1. 2023] Dostopno na:
<https://www.python.org/downloads/>
- Tkinter tutorial (online) [Uporabljeno 2.1.2023] Dostopno na:
https://www.youtube.com/watch?v=XaVp2l6Z_Dc
- W3Schools (online) [Uporabljeno 2.1.2023] Dostopna na:
<https://www.w3schools.com/>
- Tkinter (online) [Uporabljeno 3.1.2023] Dostopna na:
<https://docs.python.org/3/library/tkinter.html>
- Stack Overflow (online) [Uporabljeno 23.3.2023] Dostopno na:
<https://stackoverflow.com/>