PYTHON PROGRAMMING INTERNSHIP
PROJECT 1

Project Title:Random Username Genarator

Objective: The goal of this project is to create Python program that generates unique and fun usernames suitable for social media or gaming platforms.This project will help full you practice basic Python concepts such as working with lists,randominzation,and file handling.

Project Features:

1.Combine Random Adjectives and Nouns:

Generate usernames by combining a list of pre-defined adjectives and nouns**.**

```
Import random

#Lists of adjectives and nouns

Adjectives=[“quick”,”lazy”,”sleepy”,”happy”,”bright”,”silent”,”brave”,”clever’’]

Nouns=[“fox”,”dog”,”cat”,”river”,”mountain”,”tree”,”shadow”,”star”]

#function to generate random adjectives-nouns combinations def

generate_phrases(num_phrases=5):

phrases = [] for_in

range(num_phrases):

adjectives=random

.choice(adjectives) noun=random

.choice(nouns)

phrases.append(f”{adjective}[noun}”

) return phrases

#Generate and print 5 random phrases

Random_phrase = generate_phrases(5) For

phrase in random_phrases:
```

Print(phrase)

2.Custamization Options:

-Allows users to include numbers oe special characters in generated usernames

-Optionally set the length or structure of the username

Import random Import

string

Def generate_username(length, use_uppercase, use_lowercase,use_numbers,use_special_char):

#Define character pools

Uppercase_pool =string.ascii_uppercase

Lowercase_pool =string,ascii_lowercase Numbers_pool

=string.digits

special_chars_pool=!@#$%^&*(){}[]:;<>,.?/”’

#combine polls based on user preferences

Character_pool=”” If

use_uppercase:

Character_pool+=uppercase_pool If

use_lowercase:

Character-pool+=lowercase_pool If

use-numbers:

Character_pool+=numbers_pool

If use_special_chars:

Character_pool+=special_chars_pool #Ensure

at least one character poolis selected If not

character_pool:

raise ValueError(“you must select at least one type of character!”)

#Generate a random username

Return’’.join(random.choice(character_pool)for_in range (length)) def

main():

print(“Welcome to the custom username generator!”) try:

length=int(input(“Enter the desired lengthof the username:”))

use_uppercase=input(“Include uppercase letters?(y/n):”).lower()==’y’

use_lowercase=input(“Include lowercase letters?(y/n):”).lower()==’y’

use_numbers=input(“Include numbers?(y/n):”).lower()==’y’ use_

special_ char=input(“Include special character?(y/n):”).lower()==’y’

username=generate_username(length,use_uppercase,use_lowercase,use_numbers,use_special_chars) print(f”\nGenerated

Username:{username}”) except ValueError

as e:

print(f“Error:{e}”) if__name__==”__main__”:

main()

3.Save username to a File:

-Save the generated username to a text file for future use or sharing.

Import random

Import string Import os def

generate_username(length,use_uppercase,use_lowercase,use_numbers,use_special_chars):

#Define character pools

uppercase_pool=string.ascii_uppercase lowercase_pool=string.ascii_lowercase

number_ppol=string.digits special_chars_pool=!@#$%^&*()_+{}[]:;<>,.?/~`”

```
#combine pools based on user preferences character_pool="" if
use_uppercase:
character_pool+=uppercase_pool if
use_iowercase:
character-pool+=lowercase_pool if
use_numbers:
character_pool+=numbers_pool
if use_special_chars:
character_pool+=special_chars_pool #Ensure
at least one charater pool is selected if not
character_pool:
raise ValueError("you must select at least one type of character!")
#Generate a random username
return",.join(random,choice(character_pool)for_in range(length)
def save_username_to_file(username,filename="username.txt"):
try:
with open(filename,"a") as file: file.write(username+"\n")
print(f"Username saved to `{os.path.abspath(filename)}'.")
except Exception as e:
print(f"Error saving username:{e}") def
main():

print("Welcome to the custom Username Generator!") try;
length=int(input("Enter the desired length of the username:"))
use_uppercase=input("Include uppercase letters?(y/n):").lower()=='y'
```

use_lowercase=input(“Include lowercase latters?(y/n):”).lower()==’y’

use_numbers=input(“Include numbers?(y/n):”).lower()==’y’ use

_special_chars=input(“Include special characters?(y/n):”).lower()=’y’

username=generate_username(length,use_uppercase,use_lowercase,use_numbers,use_special_ch

ars) print(f“\nGenerated Username:{username}”) save=input(“Do you wantto save this

username to a file for future use or sharing?(y/n):”).lower() if save ==’y’:

save_username_to_file(username)

else:

print(“Username not saved.”) except

ValueError as e:

print(f“Error:{e}”) if__name__==“__main__”:

main()

4.Interactive User Input:

-Include options for users to specify their preferences(e.g,add numbers,special characters,or both) import random

import string def

get_user_preferences():

print(“Choose Input Preferences:”)

print(“A)Numbers only”) print(“B) Special character

only”) print(“c) Both numbers and special characters”)

choice=input(“Enter your

choice(A/B/C):”).strip().upper() if choice not

in[‘A’,’B’,’C’]: print(“Invalid choice.please try again.”)

```
return get_user_preferences() return choice def

get_numbers_range():

start=int(input(“Enter the start of the range:”)) end=int(input(“Enter the end

of the range:”)) return list(rang(start,end+1)) def get_special_character():

chars=input(“Enter the special characters you want to include(e.g.,!@#$%):”)

return list(chars) def generate_output(preferences,length,allow_repeats):

result=[] ifperferences[“type”]==”numbers”:

source=preferences[“numbers”] elif

preferences[“type”]==”special_characters”:

source=preferences[“special_characters”] else:#Both

source=preferences[“numbers”]+preferences[“special_characters”]

if allow_repeats:

result=random.choices(source,k=length)
else: if

len(source)<length:

print(“Error:Not enough unique elements to generate the output without repeats.”) return

[]

result=random.sample(source,length)

return result def

main():

#Get preferences

Choice=get_user_perferences()

Preferences ={}  if

choice==’A’: # numbers only
```

Preferences[“type”]=”numbers”

Preferences[“numbers”]=get_numbers_range() elif choice

==’B’ : #special characters only preferences[“type”]

=”special_characters” preferences[“special_characters”] =

get_special_characters() else: #Both

preferences[“type”]=”both” preferences[“numbers”] =

get_numbers_range() preferences[“special_characters”] =

get_spwecial_characters()

Get output preferences

Length =int(input(“Enter the length of the output:”)) allow_repeats =

input(“Allow repeated entries?(yes/no):”).strip.lower()==”yes”

#Generate output

Output = generate_output(preferences,length,allow_repeats)

if output:

print(Generated Output:”,”.join(map(str,output))) if__main__==

“__main__”:

main()

Tips for Success:

1.Plan Your Code:

-Break the project into small tasks such as defining adjective/noun lists,generating random combinatins,and saving output to a file. **Step 1:Define Input Options** adjectives =[“happy”,”bright”,”quick”] nouns = [“cat”,’sun”,”river”] numbers = list(range(0,10)) special_characters =[“!”,”@”,”#”,”$”] **Step 2:User Input for preferences** # Example function to get user preferences def get_user_input(): print(“Choose components for your

random combination:”) print(“1) Adjectives”) print(“2) Nouns”) print(“3)Numbers”) print(“4) Special Characters”) print(“5) All of the above”) choice = input(“Enter your choices (e.g.,1,2,3:”).split(“,”) return choices

def generate_combination(source_list,length,all_repeats):

if allow_repeats:

return’’.join(random.choices(source_list,k=length)) else:

if len(source_list)<length:

print(“Not enough elements for the desired length without repetition.”)

return None return’’.join(random.sample(source_list,length)) **Step 4:save output to a file** def save_to_file(output):

with open(“generated_output.txt”,”a”)as file:

file.write(f‘{output}- generated on {datetime.now()}\n”)

print(“output saved to ‘generated_output.txt’”) **Step 5:Integration** def main():

#Step1:Define lists

Adjectives=[“happy”,”bright”,”quick”]

nouns=[“cat”,”sun”,”river”] numbers

= list(range(0,10))

special_characters=[“!”,”@”,”#”,”$”]

#Step 2:User input

choice=get_user_input() source_list=[] if

“1” in choices: source_llist+=adjectives

if”2” in choices : source_list+=nouns

if “3” in choices : source_list+= numbers if “4” in

choices : source_list+=special_characters

 # Step 3:Generate combination length =

int(input(“Enter the length of the output:”))

Allow_repeats=input(“Allow repeats? (yes/no):”).lower()==”yes”

result=generate_combination(source_list,length,allow_repeats)

if result:

print(“Generated Output:”,result) **# Step 4:Save output** if

input(“Save output to a file? (yes/no):”).lower()==”yes”:

save_to_file(result)

2.Use Python Libraries:

-Leverage Python’s random module to generate random choice for adjectives and nouns.

-Use the open() function for file handling to save username import random #Listof

adjectives adjectives =

[“quick”,”lazy”,”bright”,”sleepy”,”beautiful”,”noisy”,”clam”,”happy”]

#List of nouns

Nouns=[“dog”,”cat”,”mountain”,”river”,”city”,”forest”,”bird”,”star”]

#Random select an adjective and a noun random-

adjective=random.choice(adjectives) random

_noun=random.choice(nouns) #Combine them into a phrase

random-phrase=f”The {random_adjeactive}{random_noun}”

#print the random phrase

Print(random_phrase)

3.Error Handling:

-Ensure the program handles user input gracefully(e.g.,invalid options).

def get_integer_input(): while True:

try:

user_input= int(input(“please enter a number:”))

return user_input except ValueError:

print(“Invalid input!please enter a valid number.”)

def get_menu_choice(): print(“Menu Options:”)

print (“1.Option One”) print(“2.Option Two”)

print(“Exit”) While True:

try:

choice=int(input(“please enter select an option(1-3):”)

if choice not in[1,2,3]: raise ValueError as e:

print(f‘Error:{e}.please try again.”)

4.Test Thoroughly:

-Test the program with different inputs and combinations to ensure it works as expected.

Please enter a number:abc

Invalid input!please enter a valid number.

Please enter a number:5

Please select an option(1-3):4

Error:Invalid option,please try again.

Please select an option(1-3):2

5.Make it User- Friendly:

-Use clear prompts and messages to guide the user through the program. def

get_integer_input():

```
# Clear message to the user explaining the expected input

Print(“Welcome! Please enter a valid number to continue.”) While

True:

Tey:

#Request input from the user with  clear prompt

User_input=int(input(“Enter a number(e.g.,5):”))

Print(f“Thank you!Ypu entered{user_input}.”)

Return user_input except ValuError:

#If the input is not valid , print an error and ask again

Print(“Oops!That’s not valid number.please enter a valid integer.”) def

get_menu_choice():

#Display the menu with clear options for the user

print(“/n---Menu---”)  print  (“1.Option  One”)

print(“2.Option  Two”)  print(“3.Exit”)  While

True:

Try:

# prompt userfor menu choice choice=int(input(“/nplease select an option

by entering the number(1-3):”))

# Check if choice is within the valid range if

choice not in[1,2,3]:

raise  ValueError(“Invalid  option!”)  print(f“  You

selected  Option{choice}.proceeding…”)  return

choice exept ValueError as e:
```

```
#If the choice is invalid, provide helpful feedback

print(f“Error:{e}please choose between 1 and 3.”) def

main():

print(“Welcome to the program ! let’s get started.”)

#Get the user input for a number get_integer_input()

#Get the user choice from the menu

Choice=get_menu_choice()

#Handle the user’s choice if

choice==1:

print(“You chose Option One.proceeding with Option One…”) elif

chose ==2:

print(“You chose Option Two.proceeding with Option Two…”)   elif

choice==3:

print(“Goodbye Exiting the program.”) if__name__==“__main__”:

main()
```

Name : M.Mahalakshmi

Roll.No : 213N1A04A4

Branch : ECE

College : PVKKIT Ananthapuram.