**Project Report: Random Username Generator**

**1. Project Title**

Random Username Generator

**2. Objective**

The goal of this project is to create a Python program that generates unique and customizable usernames suitable for use on social media, gaming platforms, or other online applications. The program should allow users to define their preferences for the generated usernames, including the option to include numbers, special characters, and specify a maximum username length.

**3. Technologies Used**

* **Python 3.x**: The programming language used to implement the solution.
* **Python Libraries**:
  + **random**: To generate random choices for adjectives, nouns, numbers, and special characters.
  + **string**: To access punctuation (special characters) used in the usernames.

**4. Features**

The **Random Username Generator** provides the following features:

1. **Adjective and Noun Combination**: Generates usernames by combining a random adjective and a noun from predefined lists.
2. **Customization**: Users can customize the generated usernames by:
   * Including random numbers.
   * Adding special characters.
   * Specifying a maximum length for the generated usernames.
3. **File Saving**: The program offers the option to save the generated usernames to a text file for future use or sharing.
4. **User Interaction**: Users can interact with the program to define their preferences, such as the number of usernames to generate and whether to include numbers and special characters.

**5. Code Breakdown**

**Adjective and Noun Lists**

The project starts by defining two lists, adjectives and nouns, which contain pre-defined words that will be used to form the usernames:

adjectives = [

"Brave", "Calm", "Daring", "Eager", "Fierce", "Gentle",

"Happy", "Jolly", "Kind", "Lively", "Mighty", "Noble",

"Quiet", "Royal", "Witty", "Zesty", "Cheerful", "Bold"

]

nouns = [

"Lion", "Falcon", "Phoenix", "Tiger", "Wolf", "Eagle",

"Panther", "Dragon", "Bear", "Hawk", "Fox", "Shark",

"Panda", "Knight", "Crusader", "Wizard", "Samurai", "Ninja"

]

**Username Generation**

The function generate\_username() is responsible for generating the random usernames by combining an adjective and noun, and adding optional components like numbers and special characters.

def generate\_username(include\_numbers=True, include\_special\_chars=True, username\_length=None):

adjective = random.choice(adjectives)

noun = random.choice(nouns)

username = adjective + noun

if include\_numbers:

username += str(random.randint(0, 9999))

if include\_special\_chars:

username += random.choice(string.punctuation)

if username\_length and username\_length < len(username):

username = username[:username\_length]

return username

**Saving Usernames to a File**

The function save\_usernames\_to\_file() takes a list of generated usernames and writes them to a file:

def save\_usernames\_to\_file(usernames, file\_name="usernames.txt"):

with open(file\_name, "w") as file:

for username in usernames:

file.write(username + "\n")

**Main Program Flow**

The main program flow, contained in the main() function, interacts with the user to get their preferences (how many usernames to generate, whether to include numbers and special characters, etc.) and calls the necessary functions to generate and display the usernames.

def main():

print("Welcome to the Random Username Generator!")

num\_usernames = int(input("How many usernames would you like to generate? "))

include\_numbers = input("Include numbers? (yes/no): ").strip().lower() == "yes"

include\_special\_chars = input("Include special characters? (yes/no): ").strip().lower() == "yes"

username\_length = input("Specify maximum username length (or press Enter to skip): ").strip()

username\_length = int(username\_length) if username\_length.isdigit() else None

usernames = [

generate\_username(include\_numbers, include\_special\_chars, username\_length)

for \_ in range(num\_usernames)

]

print("\nGenerated Usernames:")

for username in usernames:

print(username)

save\_option = input("\nWould you like to save these usernames to a file? (yes/no): ").strip().lower()

if save\_option == "yes":

file\_name = input("Enter the file name (default: usernames.txt): ").strip() or "usernames.txt"

save\_usernames\_to\_file(usernames, file\_name)

print("Thank you for using the Random Username Generator!")

**6. User Input and Output**

The program asks for user input to customize the username generation:

* Number of usernames to generate.
* Whether to include numbers.
* Whether to include special characters.
* Maximum username length (optional).

The generated usernames are displayed on the screen, and the user is given the option to save them to a text file.

**Sample Output:**

Welcome to the Random Username Generator!

How many usernames would you like to generate? 5

Include numbers? (yes/no): yes

Include special characters? (yes/no): yes

Specify maximum username length (or press Enter to skip):

Generated Usernames:

BraveWolf!2345

MightyTiger#567

HappyLion@1234

RoyalBear\*678

BoldDragon!234

Would you like to save these usernames to a file? (yes/no): yes

Enter the file name (default: usernames.txt): usernames\_output.txt

Usernames saved to usernames\_output.txt.

Thank you for using the Random Username Generator!

**7. Conclusion**

The **Random Username Generator** project successfully demonstrates how to generate unique and customizable usernames using Python. The program is highly flexible, allowing users to define their preferences for numbers, special characters, and username length. The option to save the generated usernames to a text file adds practical value. This project has provided a solid foundation in working with Python's built-in libraries, user input, randomization, and file handling.

**8. Future Improvements**

* **Expanded Word Lists**: Add more adjectives and nouns to increase the variety of generated usernames.
* **Error Handling**: Improve the program's handling of invalid inputs (e.g., non-numeric entries for username length).
* **GUI Integration**: Create a graphical user interface (GUI) to make the tool more user-friendly.
* **Unique Usernames**: Ensure that generated usernames are unique within a session to avoid repetition.