Project 2: Random Password Generator

Project Explanation

Objective

The aim of this project is to build a Random Password Generator that creates secure and customizable passwords. It strengthens understanding of Python's string manipulation, randomness, and security concepts.

Tools Used

- Python (core language)
- secrets module (for secure random generation)
- string module (to handle character sets)
- pyperclip module (optional, for clipboard support)

Features Implemented

1. User-defined length: User specifies the desired password length.
2. Character customization: User chooses whether to include:
   - Uppercase letters
   - Numbers
   - Special characters (symbols)
3. Strong security:
   - Ensures at least one character from each selected category.
   - Uses the 'secrets' module instead of 'random' for cryptographic safety.
   - Password is shuffled to remove predictability.
4. Clipboard copy support: If pyperclip is installed, the generated password is automatically copied.

Working

- The program asks the user for the desired password length.
- Then it prompts for preferences (uppercase, numbers, symbols).
- Based on selections, it builds a secure random password.
- The final password is displayed and optionally copied to the clipboard.

Applications

- Secure account creation
- Cybersecurity awareness
- Prevents weak or predictable passwords

Source Code

```
import string
import secrets

# Try to import pyperclip for clipboard functionality (optional)
try:
    import pyperclip
    CLIPBOARD = True
```

```
except ImportError:
    CLIPBOARD = False


def generate_password(length=12, use_upper=True, use_numbers=True, use_symbols=True):
    """
    Generate a strong random password based on user preferences.
    Uses the 'secrets' module for cryptographic security.
    """

    if length < 4:
        raise ValueError("Password length should be at least 4 characters.")

    # Base character set (always lowercase letters)
    char_set = string.ascii_lowercase

    if use_upper:
        char_set += string.ascii_uppercase
    if use_numbers:
        char_set += string.digits
    if use_symbols:
        char_set += string.punctuation

    if not char_set:
        raise ValueError("No character sets selected! Please enable at least one option.")

    # Ensure at least one character from each selected category
    password = []
    if use_upper:
        password.append(secrets.choice(string.ascii_uppercase))
    if use_numbers:
        password.append(secrets.choice(string.digits))
    if use_symbols:
        password.append(secrets.choice(string.punctuation))
    password.append(secrets.choice(string.ascii_lowercase))  # always at least one lowercase

    # Fill remaining slots with random choices
    while len(password) < length:
        password.append(secrets.choice(char_set))

    # Shuffle to remove predictability
    secrets.SystemRandom().shuffle(password)

    return ''.join(password)


# Main program
if __name__ == "__main__":
    print("■ Custom Random Password Generator ■")

    # Handle input safely
    try:
        length = int(input("Enter desired password length: "))
    except ValueError:
        print("■ Please enter a valid number for length.")
        exit(1)

    # Optional user preferences
    use_upper = input("Include uppercase letters? (yes/no): ").strip().lower() == "yes"
    use_numbers = input("Include numbers? (yes/no): ").strip().lower() == "yes"
    use_symbols = input("Include special characters? (yes/no): ").strip().lower() == "yes"

    try:
        password = generate_password(length, use_upper, use_numbers, use_symbols)
        print("\n■ Generated Secure Password:\n" + "="*40)
        print(password)
        print("="*40)

        # Copy to clipboard if pyperclip available
        if CLIPBOARD:
            pyperclip.copy(password)
            print("■ Password copied to clipboard!")
        else:
            print("■■ Install 'pyperclip' module to enable clipboard copy (pip install pyperclip).")

    except ValueError as e:
        print("■ Error:", e)
```

Output Screenshot