*import* random

*import* string

*import* tkinter *as* tk

*from* tkinter *import* ttk, messagebox, filedialog

class PasswordGenerator:

    def \_\_init\_\_(*self*, *master*):

        self.master = master

        master.title("Password Generator")

        master.geometry("400x500")

        self.frame = ttk.Frame(master, *padding*="10")

        self.frame.grid(*row*=0, *column*=0, *sticky*=(tk.W, tk.E, tk.N, tk.S))

*# Password length*

        ttk.Label(self.frame, *text*="Password Length:").grid(*column*=0, *row*=0, *sticky*=tk.W)

        self.length\_var = tk.IntVar(*value*=11)

        ttk.Spinbox(self.frame, *from\_*=9, *to*=16, *textvariable*=self.length\_var, *width*=5).grid(*column*=1, *row*=0, *sticky*=tk.W)

*# Character types*

        self.use\_lowercase = tk.BooleanVar(*value*=True)

        self.use\_uppercase = tk.BooleanVar(*value*=True)

        self.use\_digits = tk.BooleanVar(*value*=True)

        self.use\_special = tk.BooleanVar(*value*=True)

        ttk.Checkbutton(self.frame, *text*="Lowercase", *variable*=self.use\_lowercase).grid(*column*=0, *row*=1, *sticky*=tk.W)

        ttk.Checkbutton(self.frame, *text*="Uppercase", *variable*=self.use\_uppercase).grid(*column*=1, *row*=1, *sticky*=tk.W)

        ttk.Checkbutton(self.frame, *text*="Digits", *variable*=self.use\_digits).grid(*column*=0, *row*=2, *sticky*=tk.W)

        ttk.Checkbutton(self.frame, *text*="Special Characters", *variable*=self.use\_special).grid(*column*=1, *row*=2, *sticky*=tk.W)

*# Number of passwords*

        ttk.Label(self.frame, *text*="Number of Passwords:").grid(*column*=0, *row*=3, *sticky*=tk.W)

        self.num\_passwords = tk.IntVar(*value*=1)

        ttk.Spinbox(self.frame, *from\_*=1, *to*=10, *textvariable*=self.num\_passwords, *width*=5).grid(*column*=1, *row*=3, *sticky*=tk.W)

*# Generate button*

        ttk.Button(self.frame, *text*="Generate Passwords", *command*=self.generate\_passwords).grid(*column*=0, *row*=4, *columnspan*=2, *pady*=10)

*# Results area*

        self.results\_text = tk.Text(self.frame, *wrap*=tk.WORD, *width*=40, *height*=12)

        self.results\_text.grid(*column*=0, *row*=5, *columnspan*=2, *pady*=10)

*# Save button*

        ttk.Button(self.frame, *text*="Save Passwords", *command*=self.save\_passwords).grid(*column*=0, *row*=6, *columnspan*=2, *pady*=10)

    def generate\_password(*self*):

        lowercase = string.ascii\_lowercase *if* self.use\_lowercase.get() *else* ''

        uppercase = string.ascii\_uppercase *if* self.use\_uppercase.get() *else* ''

        digits = string.digits *if* self.use\_digits.get() *else* ''

        special = string.punctuation *if* self.use\_special.get() *else* ''

        all\_characters = lowercase + uppercase + digits + special

*if* not all\_characters:

*return* "Error: No character set selected"

        password = []

*if* self.use\_lowercase.get():

            password.append(random.choice(string.ascii\_lowercase))

*if* self.use\_uppercase.get():

            password.append(random.choice(string.ascii\_uppercase))

*if* self.use\_digits.get():

            password.append(random.choice(string.digits))

*if* self.use\_special.get():

            password.append(random.choice(string.punctuation))

*for* \_ *in* range(self.length\_var.get() - len(password)):

            password.append(random.choice(all\_characters))

        random.shuffle(password)

*return* ''.join(password)

    def generate\_passwords(*self*):

        self.results\_text.delete(1.0, tk.END)

*for* i *in* range(self.num\_passwords.get()):

            password = self.generate\_password()

            self.results\_text.insert(tk.END, f"Password {i+1}: {password}\n\n")

    def save\_passwords(*self*):

        content = self.results\_text.get(1.0, tk.END).strip()

*if* not content:

            messagebox.showwarning("No Passwords", "Generate passwords before saving.")

*return*

        file\_path = filedialog.asksaveasfilename(*defaultextension*=".txt", *filetypes*=[("Text files", "\*.txt"), ("All files", "\*.\*")])

*if* file\_path:

*with* open(file\_path, 'w') *as* file:

                file.write(content)

            messagebox.showinfo("Success", f"Passwords saved to {file\_path}")

*if* \_\_name\_\_ == "\_\_main\_\_":

    root = tk.Tk()

    app = PasswordGenerator(root)

    root.mainloop()