

# Fake Profile Generator Project

This Python project generates synthetic user data using Faker. It features a clean Tkinter GUI for testing and learning.

GithubRepo

# Team Members & Course

**Mariam Marwan**

ID: 231007951

**Ingy Hany**

ID: 231014596

**Salma Ahmed**

ID: 231014542

**Worood Abdou**

ID: 231017647

**Dan Mohamed**

ID: 231017562

**Zeina Mohamed**

ID: 231027513

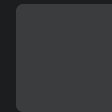
Course: Introduction to Cybersecurity (CCY2001)

# Project Objectives



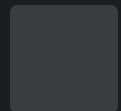
## Use Libraries

Faker and pyperclip for data generation and management.



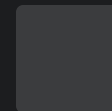
## Build GUI

Clean, user-friendly interface with Tkinter.



## Export Data

Support JSON and CSV formats.



## Extra Features

Dark mode, search, clipboard copy, and stats.

# Tools & Technologies

## Python 3.13

Core programming language.  
language.

Development done in Visual Studio Code.

## Faker Library

Generates realistic fake data.  
data.

## Tkinter

GUI framework for the  
interface.

## pyperclip

Clipboard support for copying  
copying data.

# Key Features



## Multiple Profiles

Generate many profiles with chosen fields.



## Export Options

Save profiles as CSV or JSON files.



## Live Preview

Edit and view profiles in real-time.



## Dark Mode

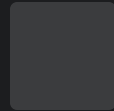
Toggle for better accessibility.

# Additional Features



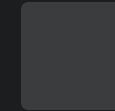
## Clipboard Copy

Copy output with one click.



## Search Function

Find keywords in preview text.



## Profile Statistics

Shows total and gender distribution.  
distribution.

# How It Works

1

## Launch App

Start the Fake Profile Generator.

2

## Input Number

Enter how many profiles to generate.

3

## Select Fields

Choose data fields like name, job, address.

4

## Choose Format

Pick CSV or JSON output.

5

## Generate & Save

Preview and export profiles.

# Project Code Structure

## Profile Generation

`generate_fake_profile()` creates profiles  
profiles with selected fields.

## Export Functions

`save_profiles_as_json()` and  
`save_profiles_as_csv()` handle exports.

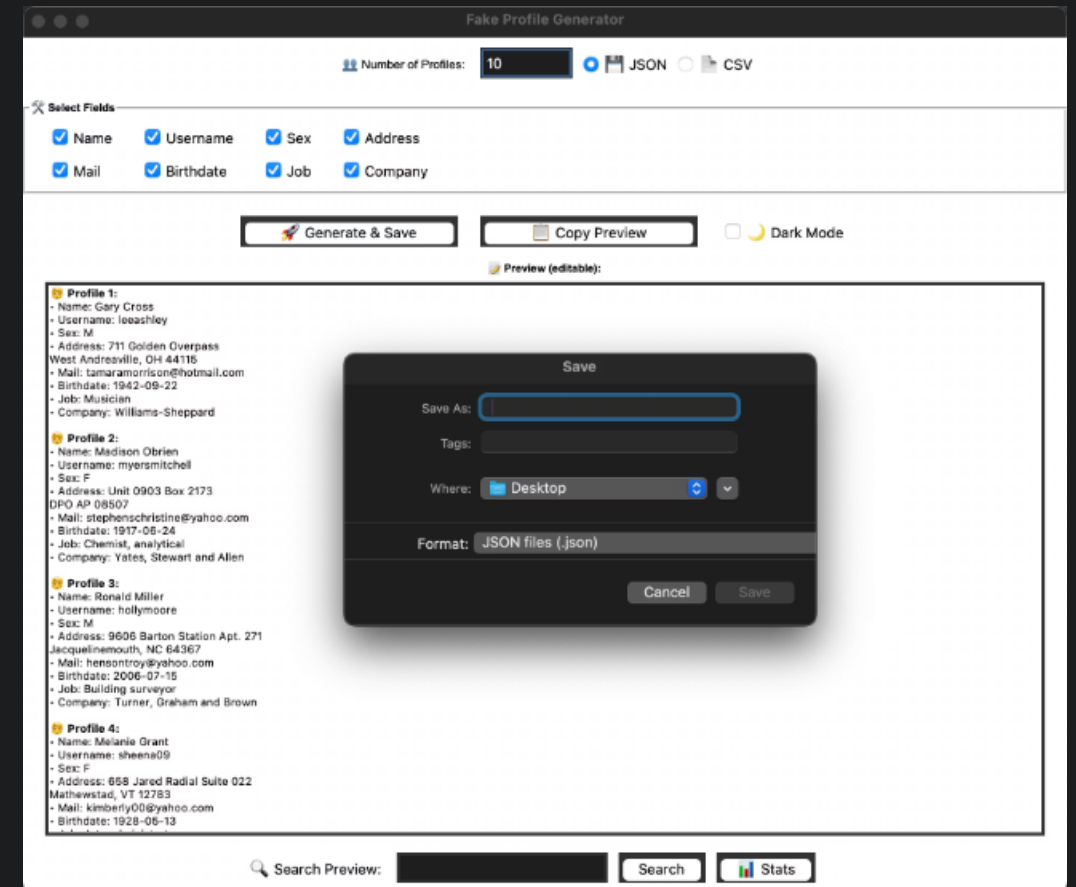
## GUI Controls

Toggle dark mode, search, copy, and update  
update stats.



# Example Output

```
{  
  "name": "John Doe",  
  "username": "johndoe95",  
  "sex": "M",  
  "address": "123 Elm Street, NY",  
  "mail": "johndoe@example.com",  
  "birthdate": "1995-08-12",  
  "job": "Software Developer",  
  "company": "TechCorp"  
}
```



# Conclusion

This project shows Python's power in GUI design and data handling.

Features like dark mode and live search enhance usability.

It is a valuable tool for developers, testers, and students.

```
from tkinter import *
from tkinter import ttk, messagebox, filedialog
from faker import Faker
import json
import csv
from decimal import Decimal
from datetime import date
import pyperclip

class CustomJSONEncoder(json.JSONEncoder):
    def default(self, o):
        if isinstance(o, Decimal):
            return str(o)
        elif isinstance(o, date):
            return o.isoformat()
        return super().default(o)

def generate_fake_profile(selected_fields, fake_instance):
    profile = fake_instance.profile()
    return {field: profile[field] for field in selected_fields if field in profile}

def save_profiles_as_json(profiles, file_path):
    with open(file_path, 'w') as f:
        json.dump(profiles, f, indent=4, cls=CustomJSONEncoder)

def save_profiles_as_csv(profiles, file_path):
    with open(file_path, 'w', newline='') as f:
        writer = csv.DictWriter(f, fieldnames=profiles[0].keys())
        writer.writeheader()
        writer.writerows(profiles)

def rate_and_save():
    n = int(entry_number.get())
    if n <= 0:
        raise ValueError
    try:
        messagebox.showerror("Invalid Input", "Please enter a valid number")
    except:
        pass
```