

SECTION-K18KK

FINAL REPORT

Topic-AI POWERED IMAGE RECOGNIZER
FROM SOCIAL MEDIA
INT-404 AI

By-

NAME	REGISTRATION NO.	ROLL NO.
MOHAMMAD AMIR IDREESI	11811970	70
AAKASH SAXENA	11805347	66

Program:Btech Cse(Hons.) School:

School of Computer Science and Engineering.

Faculty: Mr. Sagar Pande

Lovely Professional University
Jalandhar, Punjab, India.



LOVELY
PROFESSIONAL
UNIVERSITY

Transforming Education Transforming India

ACKNOWLEDGEMENT

In performing our assignment, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much Pleasure. We would like to show our gratitude MR. SAGAR PANDE for giving us a good guideline for assignment throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment. In addition, a thank you to Professor MR. SAGAR PANDE, who introduced us to the Methodology of work, and whose passion for the “underlying structures” had lasting effect. Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our assignment. We thank all the people for their help directly and indirectly to complete our assignment.

INTRODUCTION

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

Image recognition is a computer vision technique that allows machines to interpret and categorize what they “see” in images or videos. Often referred to as “image classification” or “image labelling”, this core task is a foundational component in solving many computer vision-based machine learning problems.

But how does image recognition actually work? What are the different approaches, what are its potential benefits and limitations, and how might you use it in your business?

Why is image recognition important?

Interactive marketing/Creative campaigns Image recognition is one of the most foundational and widely applicable computer vision tasks.

Image recognition’s broad and highly generalizable functionality can enable several transformative user experiences, including but not limited to:

- Automated image organization
- User-generated content moderation
- Enhanced visual search
- Automated photo and video tagging

IMAGE RECOGNIZER FROM SOCIAL MEDIA

Performing intelligence gathering is a time-consuming process, it typically starts by attempting to find a person's online presence on a variety of social media sites. While this is an easy task for a few, it can become incredibly tedious when done at scale. What if it could be automated and done on a mass scale with hundreds or thousands of individuals?

SOCIAL MAPPER

(Abstract)

Social Mapper is an open source tool that searches for profile information from social media sites, such as Facebook, Instagram, LinkedIn, Google+, V Kontakte and microblogging websites like Weibo and Douban. The tool uses names and photos as input to scan social media profiles of the people on mass. Target names and photos can be provided to the tool via csv file containing target names and URLs of the target photos, or it can be a folder containing named images. It takes approximately 60–70 seconds to scan one profile based on provided names and photos.

Social Mapper has a variety of uses in the security industry, for example the automated gathering of large amounts of social media profiles for use on targeted phishing campaigns. Facial recognition aids this process by removing false positives in the search results, so that reviewing this data is quicker for a human operator.

At a low level, Social Mapper works by running through 3 main stages. The first is target parsing, it creates a list of targets based on the input you give it. A social mapper target consists of a name and a

picture of that person. These can be provided via links in a csv file, images in a folder or via people registered to a company on LinkedIn.

Once the targets are processed, stage 2 of social mapper kicks in and it starts searching for these people online. It does this by instrumenting the Firefox browser, logging into the afore mentioned supported social media sites and begins searching for targets by name. It pulls out the top results from this search (usually between 10 and 20) and starts downloading the profile pictures and performing facial recognition checks to try and find a match. It's possible to tweak the way it performs via various parameters when the tool starts with options such as: if the program should keep searching after an initial match is found for a better one, and to change the thresholds of the facial recognition to remove more false positives at the risk of missing legitimate profiles.

Once all the social media sites have been checked, stage 3 of the tool kicks in and it starts generating your reports and data. Social Mapper has a variety of output; it generates a csv file with links to the profile pages of the target list and a more visual HTML report that can be handy for quickly checking and verifying the results.

IMPLEMENTATION

WEEK-1 (PLANNING) -Includes outline and prerequisite for the project.

WEEK-2(RESOURCE COLLECTION) -Includes learning all the concept related to project, which consists about the libraries and its implementations.

WEEK-3(CODE WRITING) -Includes writing the code. And also preparing for the necessary modules for the project.

WEEK-4(PRE-SUBMISSION REPORT) -includes short summary of what we have covered till 4th week.

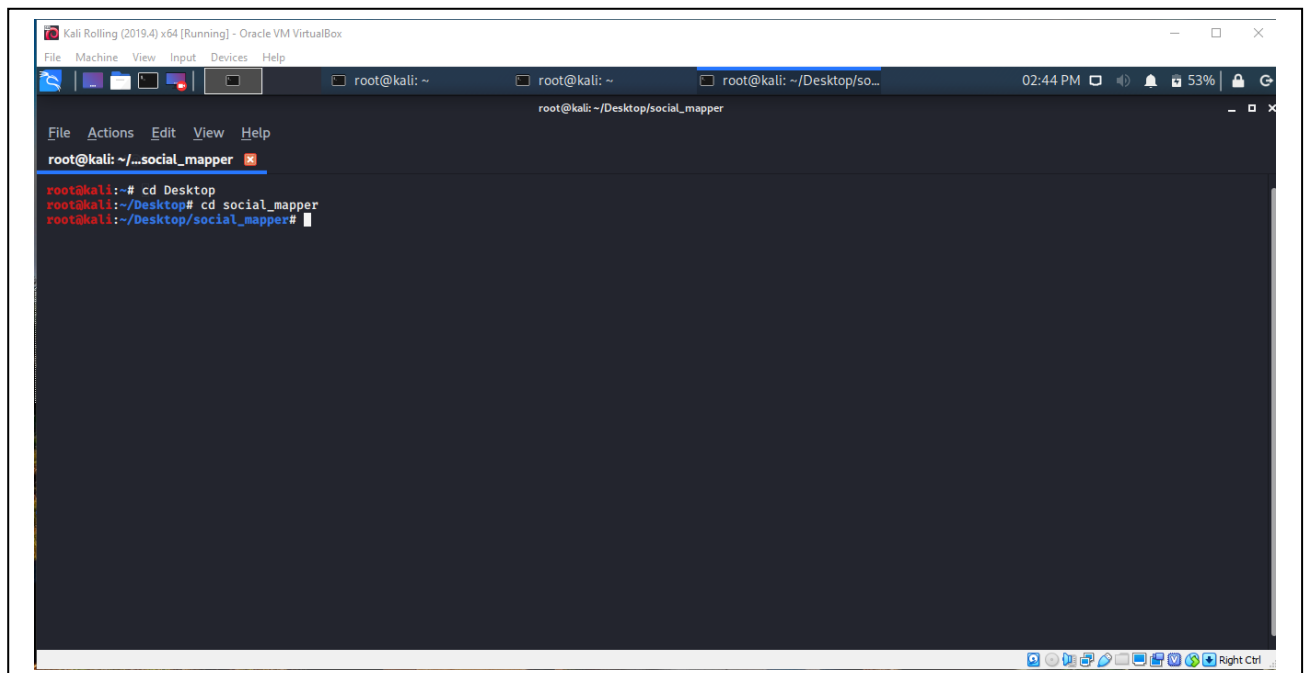
WEEK-5(COMPLETION OF CODE) -Code completion and started coding for the modules.

WEEK-6(TESTING) -Testing the projects with multiple cases and trying to fix all the errors.

WEEK-7(FINAL REPORT PREPARATION) -Final presentation preparation and GitHub upload with completion of the project.

RESULT

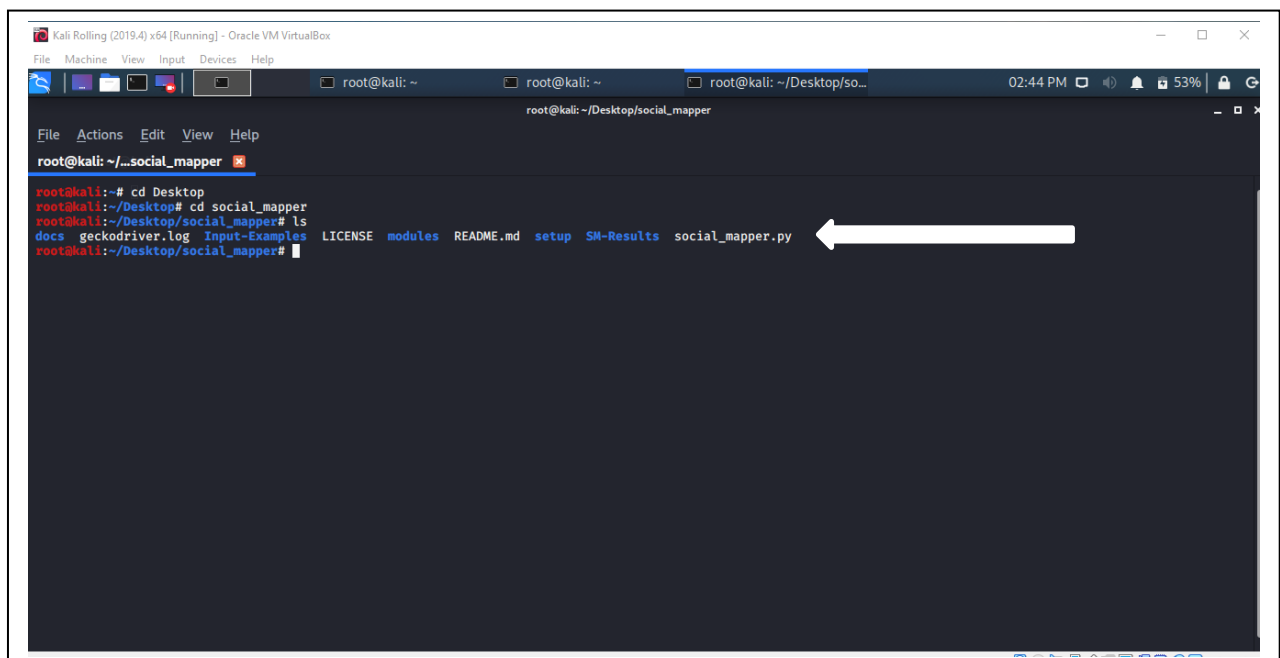
Step-1 Clone the project from GitHub and make sure to meet all the requirements, using the command **git clone https://github.com/amir7587/AI-Powered-Image-Recognizers-from-Social-Media.git**



A screenshot of a terminal window within a Kali VM. The terminal shows the following commands and output:

```
root@kali: ~/social_mapper
root@kali:~# cd Desktop
root@kali:~/Desktop# cd social_mapper
root@kali:~/Desktop/social_mapper#
```

Here social_mapper.py is the program which we have to run.



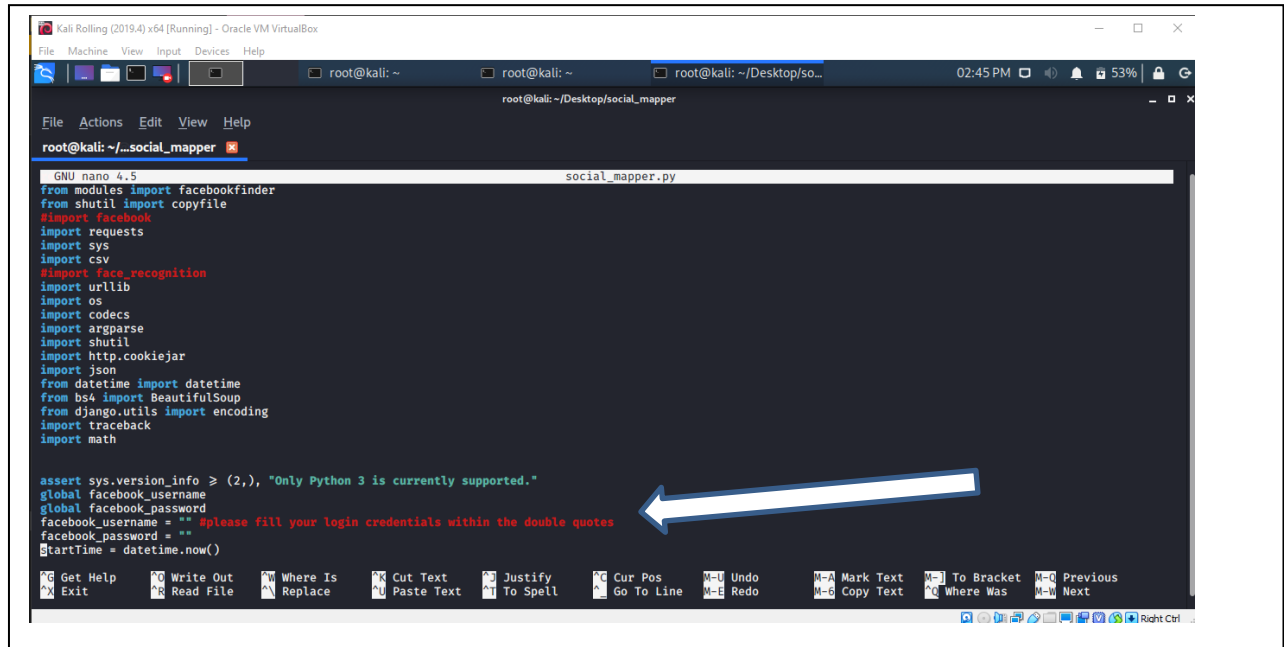
A screenshot of a terminal window within a Kali VM, showing the same directory as the previous image. The following command and output are shown:

```
root@kali:~/Desktop/social_mapper# ls
docs  geckodriver.log  Input-Examples  LICENSE  modules  README.md  setup  SM-Results  social_mapper.py
```

A white arrow points to the file `social_mapper.py` in the output.

Fill your login credentials within the double quotes

First open the social_mapper.py by typing **nano social_mapper.py**

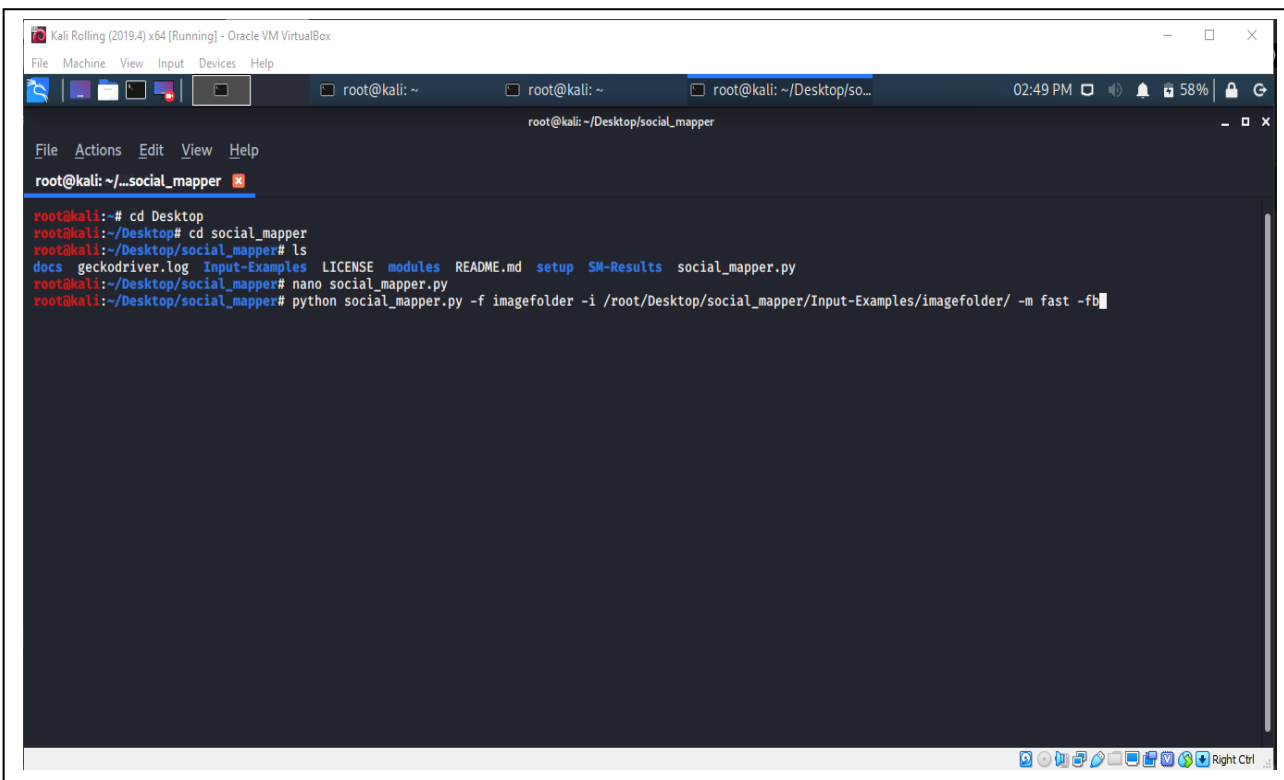


```
GNU nano 4.5 social_mapper.py
from modules import facebookfinder
from shutil import copyfile
#import facebook
import requests
import sys
import csv
#import face_recognition
import urllib
import os
import codecs
import argparse
import shutil
import http.cookiejar
import json
from datetime import datetime
from bs4 import BeautifulSoup
from django.utils import import encoding
import traceback
import math

assert sys.version_info >= (2,), "Only Python 3 is currently supported."
global facebook_username
global facebook_password
facebook_username = "" #please fill your login credentials within the double quotes
facebook_password = ""
startTime = datetime.now()

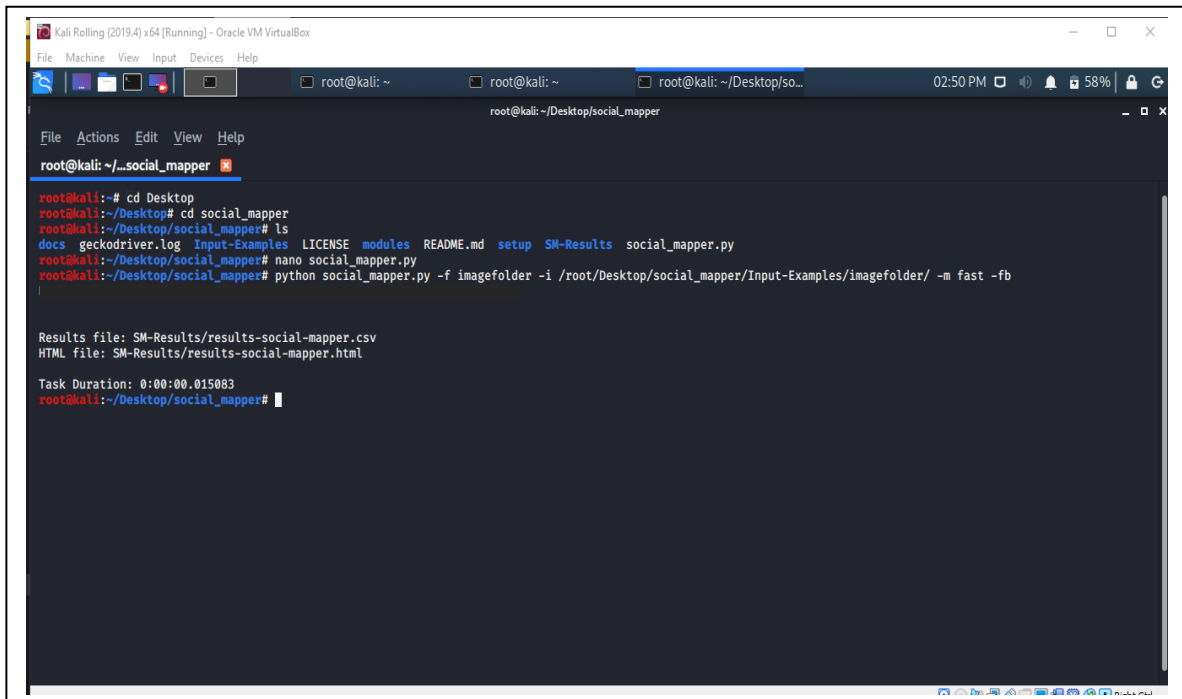
Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos  Undo  Mark Text  To Bracket  Previous
Exit      Read File  Replace  Paste Text  To Spell  Go To Line  Redo  Copy Text  Where Was  Next
```

After filling the login credentials run the program by typing the following code



```
root@kali: ~/social_mapper
root@kali:~# cd Desktop
root@kali:~/Desktop# cd social_mapper
root@kali:~/Desktop/social_mapper# ls
docs  geckodriver.log  Input-Examples  LICENSE  modules  README.md  setup  SM-Results  social_mapper.py
root@kali:~/Desktop/social_mapper# nano social_mapper.py
root@kali:~/Desktop/social_mapper# python social_mapper.py -f imagefolder -i /root/Desktop/social_mapper/Input-Examples/imagefolder/ -m fast -fb
```


Here the search is complete



```
Kali Rolling (2019.4) x64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: ~
root@kali: ~
root@kali: ~/Desktop/so... 02:50 PM 58%
root@kali: ~/Desktop/social_mapper

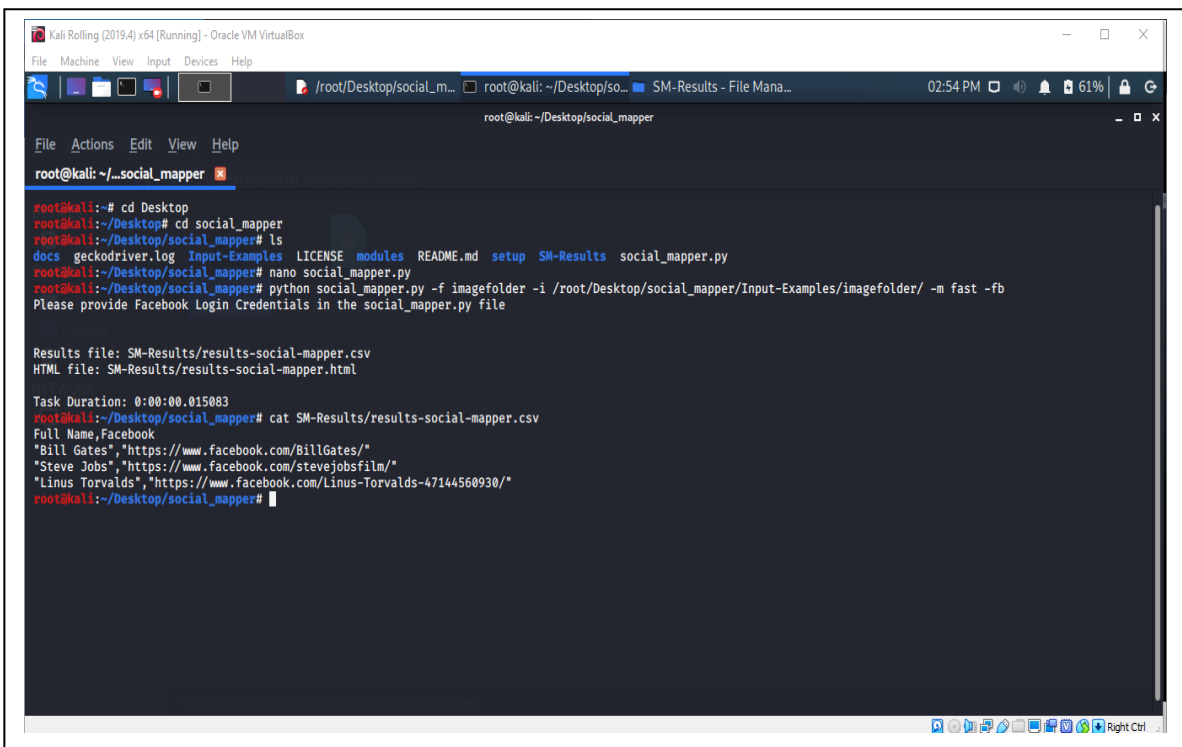
File Actions Edit View Help
root@kali: ~/...social_mapper x

root@kali:~# cd Desktop
root@kali:~/Desktop# cd social_mapper
root@kali:~/Desktop/social_mapper# ls
docs  geckodriver.log  Input-Examples  LICENSE  modules  README.md  setup  SM-Results  social_mapper.py
root@kali:~/Desktop/social_mapper# nano social_mapper.py
root@kali:~/Desktop/social_mapper# python social_mapper.py -f imagefolder -i /root/Desktop/social_mapper/Input-Examples/imagefolder/ -m fast -fb

Results file: SM-Results/results-social-mapper.csv
HTML file: SM-Results/results-social-mapper.html

Task Duration: 0:00:00.015083
root@kali:~/Desktop/social_mapper#
```

Profile link is stored in .csv file we can see it by typing the below command and see the result



```
Kali Rolling (2019.4) x64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/root/Desktop/social_m... root@kali: ~/Desktop/so... SM-Results - File Mana... 02:54 PM 61%
root@kali: ~/Desktop/social_mapper

File Actions Edit View Help
root@kali: ~/...social_mapper x

root@kali:~# cd Desktop
root@kali:~/Desktop# cd social_mapper
root@kali:~/Desktop/social_mapper# ls
docs  geckodriver.log  Input-Examples  LICENSE  modules  README.md  setup  SM-Results  social_mapper.py
root@kali:~/Desktop/social_mapper# nano social_mapper.py
root@kali:~/Desktop/social_mapper# python social_mapper.py -f imagefolder -i /root/Desktop/social_mapper/Input-Examples/imagefolder/ -m fast -fb
Please provide Facebook Login Credentials in the social_mapper.py file

Results file: SM-Results/results-social-mapper.csv
HTML file: SM-Results/results-social-mapper.html

Task Duration: 0:00:00.015083
root@kali:~/Desktop/social_mapper# cat SM-Results/results-social-mapper.csv
Full Name,Facebook
"Bill Gates","https://www.facebook.com/BillGates/"
"Steve Jobs","https://www.facebook.com/stevejobsfilm/"
"Linus Torvalds","https://www.facebook.com/Linus-Torvalds-47144560930/"
root@kali:~/Desktop/social_mapper#
```

```
Kali Rolling (2019.4) x64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mozilla Firefox /root/Desktop/social_mapper root@kali: ~/Desktop/ SM-Results - File Man... 02:56 PM 66%
root@kali: ~/Desktop/social_mapper

File Actions Edit View Help
root@kali: ~/social_mapper

root@kali: ~/Desktop/social_mapper# python social_mapper.py -f imagefolder -i /root/Desktop/social_mapper/Input-Examples/imagefolder/ -m fast -fb

Results file: SM-Results/results-social-mapper.csv
HTML file: SM-Results/results-social-mapper.html

Task Duration: 0:00:00.002426
root@kali: ~/Desktop/social_mapper# firefox SM-Results/results-social-mapper.html
Sandbox: seccomp sandbox violation: pid 1777, tid 1777, syscall 315, args 1777 139817092671040 56 0 139817092769728 139817092671040.
Sandbox: seccomp sandbox violation: pid 1844, tid 1844, syscall 315, args 1844 140380587291904 56 0 140380574499936 140380587291904.
Sandbox: seccomp sandbox violation: pid 1820, tid 1820, syscall 315, args 1820 140078065394624 56 0 140078052959904 140078065394624.
```








Here another output is in .html file which can give the detailed output of the program

Kali Rolling (2019.4) x64 [Running] - Oracle VM VirtualBox

Mozilla Firefox

/root/Desktop/social_mapper

file:///root/Desktop/social_mapper/SM-Results/results-social-mapper.html

Photo	Name	LinkedIn	Facebook	Twitter	Pinterest	Instagram
	Bill Gates	-	Vkontakte	Weibo	Douban	
	Linus Torvalds					
	Steve Jobs					

LIBRARIES USED

- **Selenium-** Selenium library:
 - Used for Automation
 - Control Web driver
 - Perform actions like – element clicks, refresh page, go to website link, etc.
- **Time-Python's time module** provides a function for getting local **time** from the number of seconds elapsed since the epoch called `localtime()`. Python `time.time()` The `time()` function returns the number of seconds passed since epoch. For Unix system, January 1, 1970, 00:00:00 at UTC is epoch (the point where *time* begins).
- **bs4-** BeautifulSoup 4. BeautifulSoup is a **Python library** for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree.
- `__future__` - `future` is a real module, and serves three purposes:
 - To avoid confusing existing tools that analyze import statements and expect to find the modules they're importing.
 - To ensure that [future statements](#) run under releases prior to 2.1 at least yield runtime exceptions.
 - To document when incompatible changes were introduced, and when they will be — or were — made mandatory.
- **Traceback-** This *module* provides a standard interface to extract, format and print stack traces of *Python* programs. It exactly mimics the behavior of the *Python* interpreter.
- **CODECS-** This module defines base classes for standard Python codecs (encoders and decoders) and provides access to the internal Python codec registry, which manages the codec and error handling lookup process.

- **Argparse-** The *argparse* module makes it easy to write user-friendly command-line interfaces. ... The *argparse* module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.
- **Shutil_**-module in Python provides many functions of high-level operations on files and collections of files. ... This module helps in automating process of copying and removal of files and directories
- **Traceback_** A *traceback* is a report containing the function calls made in your code at a specific point.
- **Urllib-** Urllib module is the URL handling module for python. It is used to fetch URLs (Uniform Resource Locators). It uses the URL open function and is able to fetch URLs using a variety of different protocols. **Urllib** is a package that collects several modules for working with URLs, such as: ... *parse* for parsing URLs.
- **Os-** The **OS** module in **Python** provides a way of using operating system dependent functionality. The functions that the **OS** module provides allows you to interface with the underlying operating system that **Python** is running on – be that Windows, Mac or Linux.

Team Responsibilities

AMIR	<ul style="list-style-type: none">➤ Resource collection➤ Write code of social_mapper.py➤ Debugging➤ Write report and prepare contents.➤ Learn & implement all the libraries used in the project
AAKASH	<ul style="list-style-type: none">➤ Information gathering➤ Prepared the module facebook_finder.py➤ Debugging➤ Prepare Report content➤ Prepare for the various libraries required in the module

REFERENCES

- Tallwalker.com
- Dzone.Ai
- Social Mapper
- YouTube

THANK YOU