

Swapnal Shahil

Mail: swapnalsahil@gmail.com

Portfolio: <https://swapnalshahil.github.io/>

Github: <https://github.com/swapnalshahil>

Linkedin: <https://www.linkedin.com/in/swapnalshahil/>

SCoRe Lab - Open MF

Google Summer of Code 2021

About Me

I am currently in the 2nd year of my Bachelor's Degree from the Indian Institute of Technology Guwahati (IIT G). I have been tinkering with the codes since the first year of my college. I love working on open-source projects and learn every time new things from projects and the community. I have good knowledge of React, Js, ES7, HTML, CSS with my interest in the web and I am contributing to this organization since December and will continue further in all my possible way.

Analytics API and UI Development

Research:

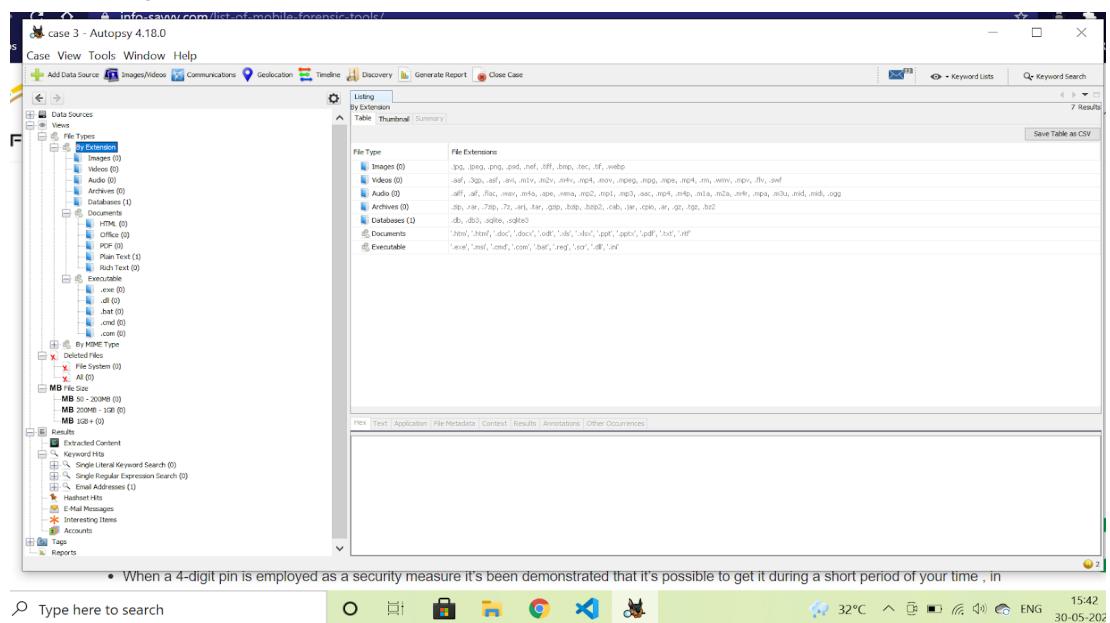
Available Analytics tool:

1. Sleuth Kit(+Autopsy):

- It's an open-source digital forensics toolkit that can be used for in-depth analysis of various file systems.

Features:

- List allocated and deleted ASCII and Unicode file names.
- Display file names and metadata structure
- Lookup file hashes in a hash database, such as the NIST NSRL, Hash Keeper, and custom databases that have been created with the 'md5sum' tool.
- Organize files according to their type.
- Multimedia - Extract EXIF from pictures and watch videos
- Give the final report in various modules like HTML, text, excel report.
- Also, it opens one case at a time and works on it.



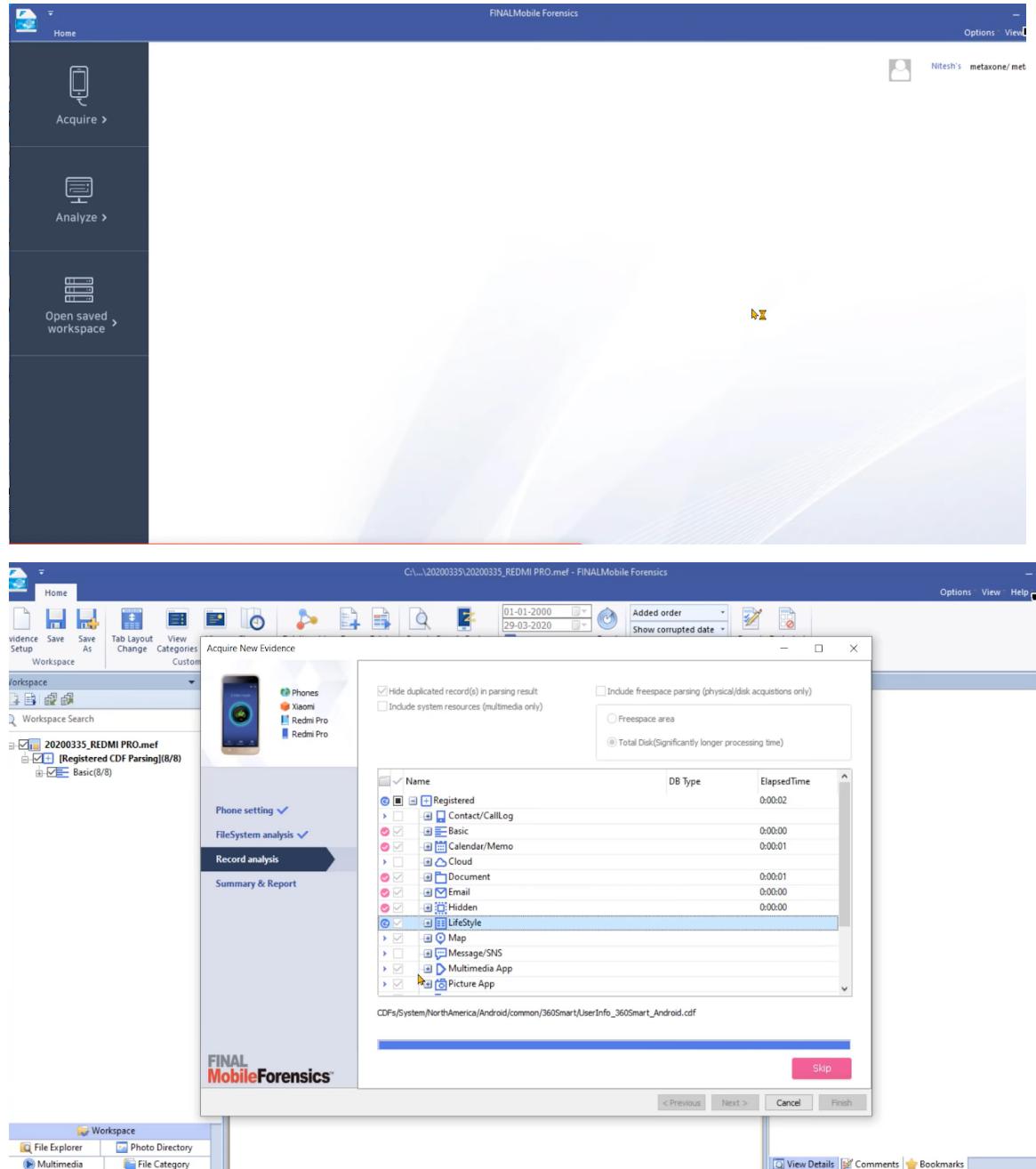
2.FINAL Mobile Forensics

- It simply captures data and analyzes it from mobile devices through logical and physical acquisitions.
- Not an open-source tool
- Subscription needed.

Features:

3

- Program is able to read its own generated phone images(MEFs)
- After data is loaded it auto-generates filename on basis of the model number of mobile devices.
- Results are generated in the HTML,pdf,excel.
- Display the file and all formats of files.



3.Nyuki Forensic Investigator (NFI)

- It's an open-source application that provides analysis for mobile device artifacts.

Features:

- Explore the device contacts, Facebook friends, LinkedIn connections, Whatsapp contacts
- Read through Facebook, LinkedIn, Skype chats
- List Telephony communications such as Calls, SMS messages
- Information related to Bluetooth devices, wifi networks, IP address leases.
- Extract ASCII and Unicode strings from Application files
- Files in a systematic way.

The screenshot shows the Silensec's Mobile Forensics Investigator (MFI) application interface. The main window title is "Silensec's Mobile Forensics Investigator (MFI)". The left sidebar has a tree view with nodes: "test", "Dashboard", "Catalogs", "Device Info", "Apps", "Networking" (which is selected), "WIFI Information", "Internet Browsing" (which is highlighted in grey), "DHCP Information", and "Device Logs". The main content area is titled "Internet Browsing" and "Browser History". It displays a table with the following data:

Date	Browser	Visits	Duration	Title
1410351535	Android Browser	10	N/A	[STABLE][2014.07.27] SuperSU v2.02 - XDA Forums
1410334188	Android Browser	2	N/A	YouTube
1410337408	Android Browser	1	N/A	Login to Twitter
1410338164	Android Browser	1	N/A	https://twitter.com/i/redirect?url=https%3A%2F%2Ftwitter.com%2Fwho_to_follow%2Fimport&t=1&sig=b224ab6acf5c55ffd6cb8c9bb24cce35ddb9a02d&id=
1410338176	Android Browser	1	N/A	https://twitter.com/who_to_follow/import
1410338276	Android	2	N/A	Request for Permission

Below the table, there are navigation buttons (back, forward, search) and a status bar at the bottom showing "Browsing History" and "1/4".

Silensec's Mobile Forensics Investigator (MFI)

Telephony

Calls SMS Messages

Message ID	Status	Type	Date Send	Local Date	From/To	Seen	Message
1	OK	Incoming	1409903678	1409936841	AIRTEL	Yes	Dear Customer, your daily club20 offer did not auto renew, you do not have sufficient funds, Recharge your account to continue enjoying the offer.
2	OK	Incoming	N/A	1410341667	AIRTEL	Yes	Thank you for the inforation
3	OK	Incoming	1410341867	1410342771	+15627312219	Yes	WhatsApp code 331-118
4	OK	Incoming	N/A	1410342822	+15627312219	Yes	331-118
6	OK	Incoming	N/A	1410357906	Airtel	Yes	Free Airtime just for you! Use your airtime & instantly get 5times bonus today! Dial* 141# for your daily Target.. Do not miss out on this great offer.
7	OK	Incoming	N/A	1410358102	Viber	Yes	Your Viber code is: 8822. Close this message and enter the code into Viber to activate your account

Telephone call log

2/4

Silensec's Mobile Forensics Investigator (MFI)

Internet Chats

Facebook App Chats LinkedIn Messages Skype Messages Whatsapp Messages

Date	Sender	Type	Attachments	Text
1408539718	Dummyj Jumaa	REGULAR	None	Hey nyathiwa to picha no ber manade yawa
0	N/A	BEFORE_FIRST_SENTINEL	None	None
1410513100	Juma Fredrick	REGULAR	None	hellow can u see mee ???
1410512997	Juma Fredrick	REGULAR	None	how are u today??
0	N/A	BEFORE_FIRST_SENTINEL	None	None
1410514049	Juma Fredrick	REGULAR	None	if you get to see me, alert me for a chart..cook??

Chats

3/4

1. Correlation between different data present in the database:-

- One-to-one(1:1): Ex- Account No. and Account holder name. Each account can have only one account holder.
- One-to-many(1:N): Ex- Like a customer can order different things from the store.
- Many-to-many(N:N): Ex - In a company N employees can work on M projects.

2. Analytics:-

A. Graphical: It basically shows data in a graphical way which saves lots of time.

Types:

- Centrality analysis: Estimates how important a node is for the connectivity of the network.
- Connectivity analysis: Determine how strongly or weakly connected two nodes are.
- Path analysis: Examines the relationships between nodes. Mostly used in shortest distance problems.

B. String-based:

Most of the digital investigations rely on textual evidence, this is obviously due to the fact that most of the stored digital data is linguistic. Ex- logged conversation, a lot of important text-based evidence can be gathered while dumping strings from images (smartphone memory dumps) and can include emails, instant messaging, address books, browsing history, and so on. Most of the currently available digital forensic tools rely on matching and indexing algorithms to search textual evidence at a physical level so that they search every byte to locate specific text strings.

C. Pattern Based:

Pattern Analytics can be defined as a discipline of Big Data that enables business leaders to understand how different variables of the business interact and are linked with each other. Variables can be of any kind and within any data source, structured as well as unstructured. Machine Learning(ML) is mainly used to handle such big data.

3. Data can be presented in one of the three ways:

- a) Text presentation: Text can be used to provide interpretation or emphasize certain data. If quantitative information to be conveyed consists of one or two numbers, it is more appropriate to use written language than tables or graphs.
- b) Table presentation: Tables convey information that has been converted into words or numbers in rows and columns. Easy to understand. Tables are the most appropriate for presenting individual information and can present both quantitative and qualitative information.
- c) Graph presentation: Graphs are effective for presenting large amounts of data, they can be used in place of tables to present small sets of data. A graph format that best presents information must be chosen so that readers and reviewers can easily understand the information.

Project Goals

1. Generate dummy dataset from a rooted device.
2. Extraction of the text messages.
3. Extraction of browser history.
4. Extraction of Bluetooth files.
5. Extracting data (EXIF) from images(like location, date, and time)
6. Extraction of location from the device.
7. Keyword search - Implementation of keyword search to get all the cases that have the searched keyword.
8. Implementing keyword search from cases.
9. Maximum occurred common word/data between two cases.
10. Returning all the common words between two cases.
11. Filtering Cases by Date or range of date.
12. Creating APIs for the above.
13. Tag association and tag-based search like searching cases that are tagged with theft to get all the theft-related cases.
14. Customized keyword search <kw1> + <kw2> + ... + <kwN>
15. Analytics based on location.
16. Implementing analytics on the front end.

17. Presentation of observed analytics - graph, library, an API will be developed which will send the database data to the graph generating library to create required graphs and that will show in our OpenMF frontend.
18. Presentation of observed analytics between two cases using graphs and tables.
19. Update the project wiki.
20. Will write blogs and explain the project on LinkedIn and Medium. Also, I have an idea to make video tutorials explaining the project and work, which further publish on YouTube and LinkedIn to gain the crowd on the Website.

Timeline

Note: Each cell contains a time period of 1 week.

Week No.	Date	Work
0)	17 May - 7 June	Community Bonding (take feedback from mentors, Modify timeline and project details)
Coding officially begins!		
1)	7 June - 13 June	<ul style="list-style-type: none"> ● Generate dummy dataset from a rooted device. ● Extraction of the text messages. ● Creating API which will show the extracted text messages. ● Implementation from the command line so that we can directly extract SMS.
2)	14 June - 20 June	<ul style="list-style-type: none"> ● Extraction of browser history ● Extraction of Bluetooth files/data. ● Creating APIs which will show the extracted files for the above. ● Implementation from the command line to extract browser history

		and Bluetooth data directly.
3)	21 June - 27 June	<ul style="list-style-type: none"> • Extracting data (EXIF) from images(like location, date, and time) • Extraction of location from the device • Creating APIs which will show the extracted files for the above. • Implementation from the command line to extract EXIF and location data directly.
4)	28 June - 4 July	<ul style="list-style-type: none"> • Keyword search - Implementation of keyword search to get all the cases that have the searched keyword. • Implementing keyword search from cases. • Creating APIs for the above.
5)	5 July - 11 July	<ul style="list-style-type: none"> • Maximum occurred common word/data between two cases. • Returning all the common words between two cases. • Creating APIs for the above.

Evaluations (July 12- July 16)

6)	12 July - 18 July	<ul style="list-style-type: none"> • Taking feedback from mentors and working on bugs(if any). • Leftover work(if any) • Filtering Cases by Date or range of date. • Creating API for that. • Implementing frontend from week 4th,5th, and 6th in the analytics section.
7)	19 July- 25 July	<ul style="list-style-type: none"> • Tag association and tag-based search like searching cases that are tagged with theft to get all the theft-related cases. • Customized keyword search <kw1> + <kw2> + ... + <kwN> • Analytics based on location • Frontend implementation for this week.
8)	26 July - 1 Aug	<ul style="list-style-type: none"> • Presentation of observed analytics - graph, library, an API will be developed which will send the database data to the graph generating library to create required graphs and that will show in our OpenMF frontend.
9)	2 Aug - 8 Aug	<ul style="list-style-type: none"> • Presentation of observed analytics between two cases using graphs and tables.
10)	9 Aug - 15 Aug	<ul style="list-style-type: none"> • Leftover work (if any) • Writing Wiki.

		• Writing Medium Blog for documenting all progress and work done.
Final Evaluation week starts (Aug 16- Aug 23)		
Final results of Google Summer of Code 2021 announced (August 31)		

Implementation:

(7 June - 13 June)

Setting up the project and generated some dummy data which will be important in the next work period. Also, working on the extraction part to extract text messages (SMS) from android devices and Implementation from the Postman and CMD so that SMS can be directly extracted through dedicated command. To make this useful on the frontend and analysis, An API is also created that will provide a data path for SMS of the desired case stored in the database.

```

Command Prompt
Extracting current db from: /data/data/com.samsung.advp.imssettings/databases/ims.db
Extracting current db from: /data/data/com.sec.android.app.myfiles/databases/myfiles.db
Extracting current db from: /data/data/com.wsomacp/databases/wsomacp.db
Extracting current db from: /data/data/com.google.android.apps.docs/appdocs.db
databases extraction completed...
smsdb does not exist
[['Shell permissions', 'root(su)'], ['ADB serial', '3adad83b\r\n'], ['Manufacturer', 'samsung'], ['Model', 'SM-G600FY'], ['IMEI', ''], ['Android version', '6.0.1'], ['Build name', 'MMB29M.G600FYXXU1BRD2'], ['Wifi MAC', 'e4:5d:75:b5:20:56'], ['Local time', '2021-06-09 22:22:59 India Standard Time'], ['Android time', '2021-06-09T14:22:59Z'], ['Account ', ['com.google: tswap4693@gmail.com', 'com.samsung.android.email: tswap4693@gmail.com', 'com.facebook.auth.login: Facebook', 'com.samsung.android.coreapps: +91 821 006 701']]
Saved report successfully

(venv) C:\Users\Swapnil Shahil\git_projects\OpenMF>python converter.py extractsms mmssms.db tsv
Tables in database are :
android_metadata
pdu
sqlite_sequence
addr
part
rate
drm
sms
raw
incomplete_lms
attachments
sr_pending
im
ft
ft_retry
im_threads
cmas
wpm
canonical_addresses
threads
pending_msgs
mychannels
words
words_content
words_segments
words_segdir
spam_pdu

```

(Extraction of text messages using CMD)

_id	thread_id	address	person	date	date_sent	protocol	read	status	type	reply_path_present
1										
2	3	JA-JioSVC	None	1622951068463	1622951052000	0	1	-1	1	0
3	To know 'How to Set up Jio Network' on mobile, click https://youtu.be/o18LboDilho									For seamless data experience
4	To know your number, track balance & usage, give a missed call to 1299. 1299									
5	7	4	[REDACTED]	None	1622961172315	1622961171000	0	1	-1	1
6	8	4	[REDACTED]	None	1622961182594	1622961181000	0	1	-1	1
7	9	4	[REDACTED]	None	1622961188679	1622961188000	0	1	-1	1
8	10	4	[REDACTED]	None	1623059523761	0	None	1	-1	5
9	11	4	[REDACTED]	None	1623061396257	0	None	1	-1	5
10	12	4	[REDACTED]	None	1623076273359	1623076271000	0	1	-1	1
11	13	5	56161179	None	1623125457024	0	None	1	-1	5
12	14	6	59029411	None	1623128868808	0	1	-1	1	0
13	15	7	JK-620016	None	1623138017359	1623138015000	0	1	-1	1
14	16	8	JX-JioPay	None	1623166083721	1623166077000	0	1	-1	1
15	इस रिचार्ज में आपको मिलता है : Benefits: 1. UNLIMITED DATA - 24 GB (1 GB/Day) 2. Unlimited Voice Calls 3. 100 SMS/									
16	ट्रॉज़ेन आईडी : [REDACTED]									
17	अपने मोजूदा और आगामी प्लान के बारे में जानने के लिये लिंक करें : https://www.jio.com/dl/my_plans									
18	रीचार्ज की अपना अनुबंध शेयर करने के लिए लिंक करें https://www.jio.com/en-in/jio-rech-exp-survey-hindi?custid=1253002									
19	अपना मोजूदा बैलेंस, प्लान की जानकारी और आकर्षक रिचार्ज प्लान के बारे में जानने के लिये 1991 पर कॉल करें। +917012075009									
20	17 9 JK-JioPay None 1623166109827 1623166077000 0 1 -1 1 0 None Recharge of Rs. 149.0									
21	Entitlement: Benefits: 1. UNLIMITED DATA - 24 GB (1 GB/Day) 2. Unlimited Voice Calls 3. 100 SMS/Day 4. Complimentary 100 SMS/Day									
22	Transaction ID: [REDACTED]									
23	To view details of your current and upcoming plan, click https://www.jio.com/dl/my_plans									
24	To share your recharge experience, click https://www.jio.com/en-in/jio-rech-exp-survey?custid=125300211281									
25	Dial 1991, to know your current balance, validity, plan details and for exciting recharge plans.									+917012075009
26	18 10 [REDACTED] None 1623166133051 0 None 1 -1 2 None None Hey mom None 0 0 -1									

(Extracted text messages(SMS))

(14 June - 20 June)

Work on the extraction part of the browser and write scripts for the same. For this, research all the tables from the database and try to gather all the necessary information which will be useful for mobile forensic. Implementation from POSTMAN and CMD to directly extract browser data through dedicated command and write an API that will provide the data path to history.tsv, which will contain all the necessary information of the desired case stored in a database.

```
Tables in database are ::  
meta  
urls  
sqlite_sequence  
visits  
visit_source  
keyword_search_terms  
downloads  
downloads_url_chains  
downloads_slices  
segments  
segment_usage  
typed_url_sync_metadata
```

Tables from the chrome history database

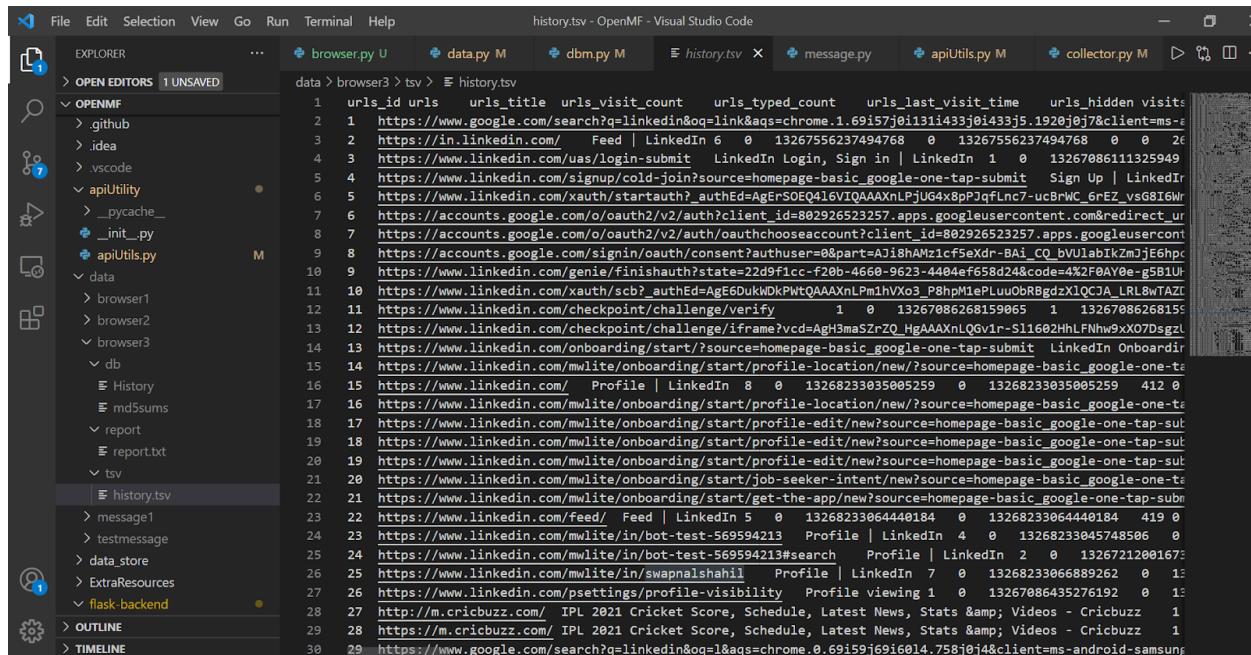
```
(venv) C:\Users\Swapnil Shahil\Documents\OpenMF>python collector.py --option browser --session_name browser4
* daemon not running; starting now at tcp:5037
* daemon started successfully
Shell permissions: root(su)

General Device Information

ADB serial: 3adad83b

Device model: samsung SM-G600FY
IMEI:
Android version: 6.0.1
Build number: MB29M.G600FYXXU1BRD2
Wi-Fi MAC: e4:5d:75:b5:20:56
Local time: 2021-06-17 11:39:12 India Standard Time
Android time: 2021-06-17
Sync'ed Accounts.
com.google: tswap4693@gmail.com
com.samsung.android.email: tswap4693@gmail.com
com.samsung.android.coreapps: [REDACTED]
('h', '--help')
Starting data extraction plan for given options
Running data extraction for selected options :
Options to parse are : ['browser']
Session name : browser4
Extracting all common databases ...
Extracting current db from: /data/data/com.google.android.apps.messaging/databases/bugle_db
Extracting current db from: /data/data/com.google.android.dialer/databases/dialer.db
Extracting current db from: /data/data/com.android.providers.contacts/databases/calllog.db
Extracting current db from: /data/data/com.android.providers.contacts/databases/voicemail.db
Extracting current db from: /data/data/com.google.android.apps.messaging/databases/bugle_db
Extracting current db from: /data/data/com.google.android.dialer/databases/dialer.db
Extracting current db from: /data/data/com.android.providers.contacts/databases/calllog.db
Extracting current db from: /data/data/com.android.providers.contacts/databases/voicemail.db
```

Extraction of browser data using CMD



```
data > browser3 > tsv > history.tsv
1 urls_id urls urls_title urls_visit_count urls_typed_count urls_last_visit_time urls_hidden visits
2 1 https://www.google.com/search?q=linkedin&oq=link&aq=chrome.1.69157j0131i433j01433j5.1920j0j7&client=ms-edges
3 2 https://in.linkedin.com/ Feed | LinkedIn 6 0 13267556237494768 0 13267556237494768 0 0 26
4 3 https://www.linkedin.com/uas/login-submit LinkedIn Login, Sign in | LinkedIn 1 0 13267086111325949
5 4 https://www.linkedin.com/signup/cold-join?source=homepage-basic_google-one-tap-submit Sign Up | LinkedIn
6 5 https://www.linkedin.com/xauth/startauth?authEd=AgErSOEQ416VIOQAAAXnLPjU64x8pP3qfLn7-uclBWC_6rEZ_vSG816Wr
7 6 https://accounts.google.com/o/oauth2/v2/auth?client_id=802926523257.apps.googleusercontent.com&redirect_ur
8 7 https://accounts.google.com/o/oauth2/v2/auth/oauthchooseaccount?client_id=802926523257.apps.googleusercontent.com&
9 8 https://accounts.google.com/signin/oauth/consent?authuser=&part=Aj18AMz1cf5exdr-BAi_CQ_bVULab1kZmjE6hpc
10 9 https://www.linkedin.com/genie/finishauth?state=22d9f1cc-f20b-4660-9623-4404ef658d24&code=4%2F0AY0e-g5B1U
11 10 https://www.linkedin.com/xauth/scb?authEd=AgE6DUkWOkPWTqAAAXnPlm1hvXo3_P8hPM1ePluuObR8gdzX1QCJA_LRL8wTAZc
12 11 https://www.linkedin.com/checkpoint/challenge/verify 1 0 13267086268159065 1 13267086268159065
13 12 https://www.linkedin.com/checkpoint/challenge/iframe?vcid=AgH3maS7rZQ_HgAAAAXnLQGvIr-S11602HhLFNhw9xX07DsgzL
14 13 https://www.linkedin.com/onboarding/start/?source=homepage-basic_google-one-tap-submit LinkedIn Onboardin
15 14 https://www.linkedin.com/mwlite/onboarding/start/profile-location/new/?source=homepage-basic_google-one-ta
16 15 https://www.linkedin.com/_Profile | LinkedIn 8 0 13268233035005259 0 13268233035005259 412 0
17 16 https://www.linkedin.com/mwlite/onboarding/start/profile-location/new/?source=homepage-basic_google-one-ta
18 17 https://www.linkedin.com/mwlite/onboarding/start/profile-edit/new?source=homepage-basic_google-one-tap-su
19 18 https://www.linkedin.com/mwlite/onboarding/start/profile-edit/new?source=homepage-basic_google-one-tap-su
20 19 https://www.linkedin.com/mwlite/onboarding/start/profile-edit/new?source=homepage-basic_google-one-tap-su
21 20 https://www.linkedin.com/mwlite/onboarding/start/job-seeker-intent/new?source=homepage-basic_google-one-ta
22 21 https://www.linkedin.com/mwlite/onboarding/start/get-the-app/new?source=homepage-basic_google-one-tap-su
23 22 https://www.linkedin.com/feed/_Feed | LinkedIn 5 0 13268233064440184 0 13268233064440184 419 0
24 23 https://www.linkedin.com/mwlite/in/bot-test-569594213 Profile | LinkedIn 4 0 13268233045748506 0
25 24 https://www.linkedin.com/mwlite/in/bot-test-569594213#search Profile | LinkedIn 2 0 13267212001673
26 25 https://www.linkedin.com/mwlite/swapsnalsahil Profile | LinkedIn 7 0 1326823306889262 0 13267086435276192 0
27 26 https://www.linkedin.com/psettings/profile-visibility Profile viewing 1 0 13267086435276192 0 13267086435276192 0
28 27 http://m.cricbuzz.com/ IPL 2021 Cricket Score, Schedule, Latest News, Stats & Videos - Cricbuzz 1
29 28 https://m.cricbuzz.com/ IPL 2021 Cricket Score, Schedule, Latest News, Stats & Videos - Cricbuzz 1
30 29 https://www.google.com/search?q=linkedin&oq=1&qs=chrome.0.69i59j69i6014.758j0j4&client=ms-android-samsu
```

Extracted data in history.tsv

Later work on the extraction of Bluetooth data from an android device and implement this to extract from Postman and CMD, same as the browser part. From this, we will get all the

information like which data is transferred from the current android device to whatever connected device, data path, time when data is shared, direction and destination, total bytes, and device name.

```
(venv) C:\Users\Swapnil Shahil\Documents\OpenMF>python collector.py --option bluetooth --session_name bluetooth5
Shell permissions: root(su)

General Device Information

ADB serial: 3adad83b

Device model: samsung SM-G600FY
IMEI:
Android version: 6.0.1
Build number: MMB29M.G600FYXXU1BRD2
Wi-Fi MAC: e4:5d:75:b5:20:56
Local time: 2021-06-17 14:03:02 India Standard Time
Android time: 2021-06-17
Sync'd Accounts.
com.google: tswap4693@gmail.com
com.samsung.android.email: tswap4693@gmail.com
com.samsung.android.coreapps: [REDACTED]
{'--help', '-h'}
Starting data extraction plan for given options
Running data extraction for selected options :
Options to parse are : ['bluetooth']
Session name : bluetooth5
Extracting all common databases ...
Extracting current db from: /data/data/com.google.android.apps.messaging/databases/bugle_db
Extracting current db from: /data/data/com.google.android.dialer/databases/dialer.db
Extracting current db from: /data/data/com.android.providers.contacts/databases/callog.db
Extracting current db from: /data/data/com.android.providers.settings/databases/settings.db
Extracting current db from: /data/data/com.android.providers.contacts/databases/contacts2.db
Extracting current db from: /data/data/com.sec.android.provider.logsprovider/databases/logs.db
Extracting current db from: /data/data/com.android.providers.telephony/databases/mmssms.db
Extracting current db from: /dbdata/databases/com.android.providers.telephony/databases/mmssms.db
```

Extraction of Bluetooth data using CMD

```
data > bluetooth4 > tsv > bluetooth.ts
1 | id uri hint _data
2 | 1 content://@media/external/images/media/191 download.jpeg.jpg image/jpeg 0 40:44:FD:7A:6F:BE
3 | 2 content://media/external/video/media/192 VID-20210611-WA0013.mp4 /storage/emulated/0/Download/VID-20210611-WA0013.mp4 video/mp4 0 40:44:FD:7A:6F:BE
```

Extracted data in bluetooth.tsv(above)

(21 June - 27 June)

Work on the extraction of EXIF data from images and location data from an android device. EXIF stands for Exchangeable image file format and refers to the basic metadata that is generated and stored by a camera or device whenever you take a photo. In simple words whenever we click a photo some data get embedded within our image and include pieces of information like date when it was taken, date if modified with any software, software id, a location where the pic was clicked, time, description, bucket id, bucket name, album, tags, width, height and many more.

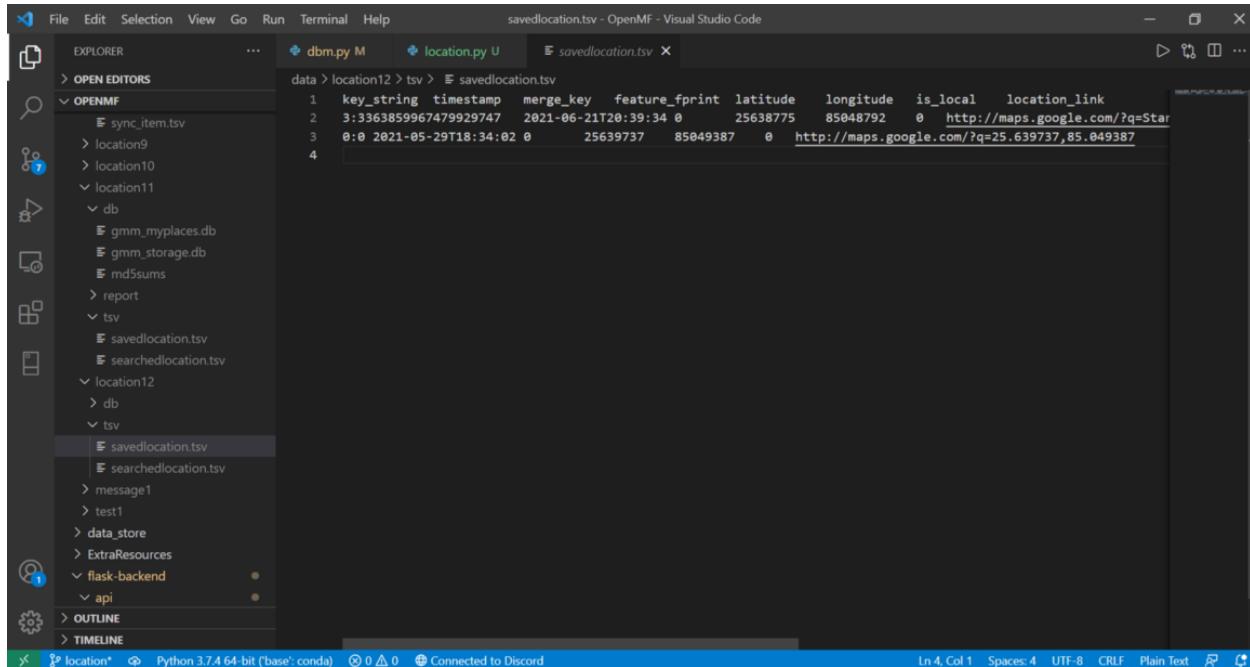
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it displays a tree view of the project structure:
 - OPEN EDITORS:** Shows `dbm.py`, `mediadata.tsv`, and `media.py`.
 - DATA:** `media4 > tsv > mediadata.tsv`
 - OPENMF:** Contains:
 - `mediadata.tsv`
 - `media3`: Contains `db`, `report`, and `tsv`. The `tsv` folder contains `mediadata.tsv`.
 - `media4`: Contains `db`, `report`, and `tsv`. The `tsv` folder contains `mediadata.tsv`.
 - `message1`
 - `test1`
 - `data_store`
 - `ExtraResources`
 - `flask-backend`: Contains `api`, `routes`, and `__init__.py`. The `api` and `routes` folders contain `__pycache__` subfolders.
- Code Editor:** The main area shows the contents of `mediadata.tsv` as a CSV file. The first few rows are:

	_id	data	_size	datetaken	mime_type	title	description	_display_name	picasa_id	orientation	latit
1	20	/storage/emulated/0/Samsung/Image/5.	Nightscape.jpg	983589	2014-08-01T19:48:14	image/jpeg	5.	Nightscape			
2	21	/storage/emulated/0/Samsung/Image/6.	Bridge.jpg	522139	2014-08-01T19:48:14	image/jpeg	6.	Bridge			
3	22	/storage/emulated/0/Samsung/Image/7.	Starlight.jpg	491165	2014-08-01T19:48:14	image/jpeg	7.	Starlight			
4	93	/storage/emulated/0/DCIM/Camera/20210530_111405.jpg	1360519	2021-05-30T11:14:05	image/jpeg	20210530_111405					
5	97	/storage/emulated/0/WhatsApp/Media/WhatsApp Images/IMG-20210531-WA0001.jpg	241874	2021-05-31T11:36:07	image/jpeg	IMG-20210531-WA0001					
6	161	/storage/emulated/0/DCIM/Facebook/FB_IMG_1622804032764.jpg	44063	2021-06-04T16:23:52	image/jpeg	FB_IMG_1622804032764					
7	185	/storage/emulated/0/Pictures/Messenger/received_385341084782291.jpeg	31927	2021-06-07T09:59:53	image/jpeg	received_385341084782291					
8	189	/storage/emulated/0/DCIM/Camera/20210610_112615.jpg	1734850	2021-06-10T11:26:15	image/jpeg	20210610_112615					
9	190	/storage/emulated/0/Download/download.webp	7052	2021-06-12T19:19:32	image/webp	download					
10	191	/storage/emulated/0/Download/download.jpeg	12734	2021-06-12T19:31:57	image/jpeg	download.jpeg					
11											
12											
- Bottom Status Bar:** Shows tabs for `media*`, `Python 3.7.4 64-bit (base: conda)`, and `Connected to Discord`. It also shows file statistics (Ln 1, Col 1, Spaces: 4, CRLF) and icons for Plain Text and Prettier.

Data from Images (above)

Later work on the location part to get all the saved locations and searched locations. Create APIs for the above and implement from CMD and POSTMAN both.



```

File Edit Selection View Go Run Terminal Help
OPEN EDITORS
OPENMF
  sync_item.tsv
  location9
  location10
  location11
    db
      gmm_myplaces.db
      gmm_storage.db
      md5sums
    report
    tsv
      savedlocation.tsv
      searchedlocation.tsv
  location12
    db
    tsv
      savedlocation.tsv
      searchedlocation.tsv
      message1
      test1
    data_store
    ExtraResources
      flask-backend
        api
    OUTLINE
    TIMELINE
dbm.py M location.py savedlocation.tsv
data > location12 > tsv > savedlocation.tsv
1   key_string    timestamp    merge_key    feature_fprint    latitude    longitude    is_local    location_link
2   3:363859967479929747 2021-06-21T20:39:34 0       25638775    85048792    0   http://maps.google.com/?q=Star
3   0:0 2021-05-29T18:34:02 0       25639737    85049387    0   http://maps.google.com/?q=25.639737,85.049387
4

```

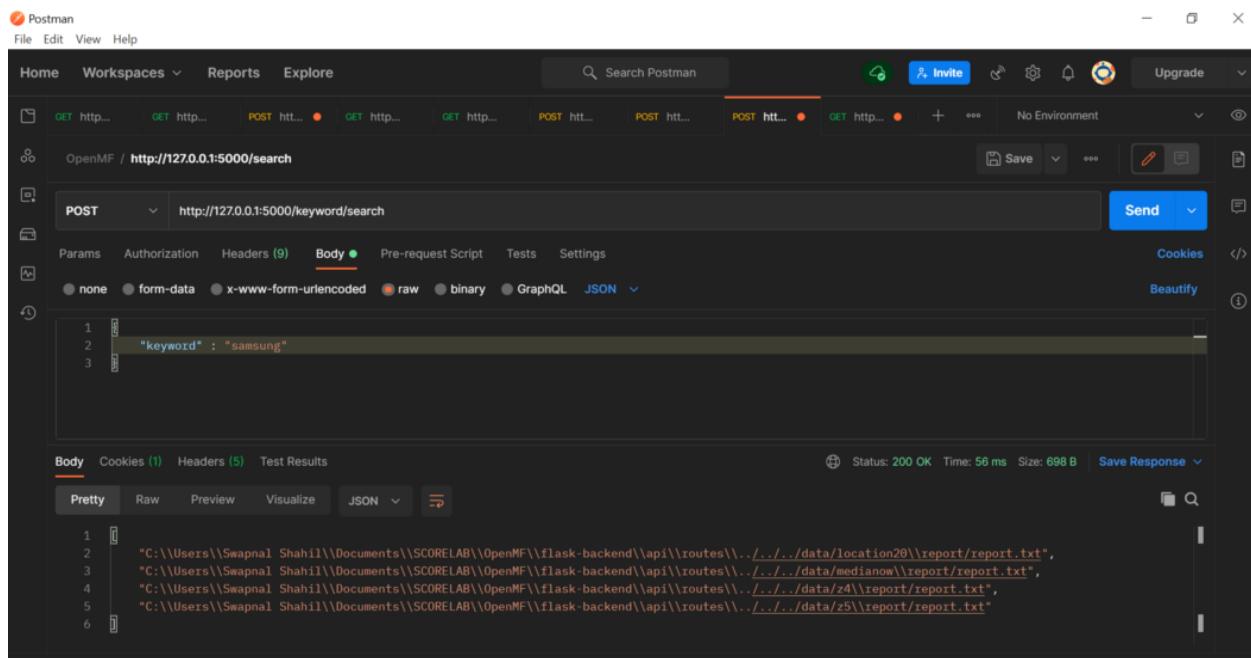
Ln 4, Col 1 Spaces: 4 UTF-8 CRLF Plain Text

Saved location Data (above)

(28 June - 4 July)

Work on keyword search to get all the cases that have the searched keyword. Basically main purpose for this was to find a word from the database if the admin wants to search while

analyzing the mobile forensic case. For this, make two APIs that have some similar kind of function but have different use in their own place. One API search the keyword from the whole database so that analyst can relate one case with other cases and another API works within the case to get the files which have that keyword.



The screenshot shows the Postman application interface. The top navigation bar includes 'File', 'Edit', 'View', 'Help', 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar says 'Search Postman'. The main workspace shows a collection named 'OpenMF / http://127.0.0.1:5000/search'. A POST request is selected with the URL 'http://127.0.0.1:5000/keyword/search'. The 'Body' tab is active, showing a JSON payload:

```

1
2 "keyword" : "samsung"
3

```

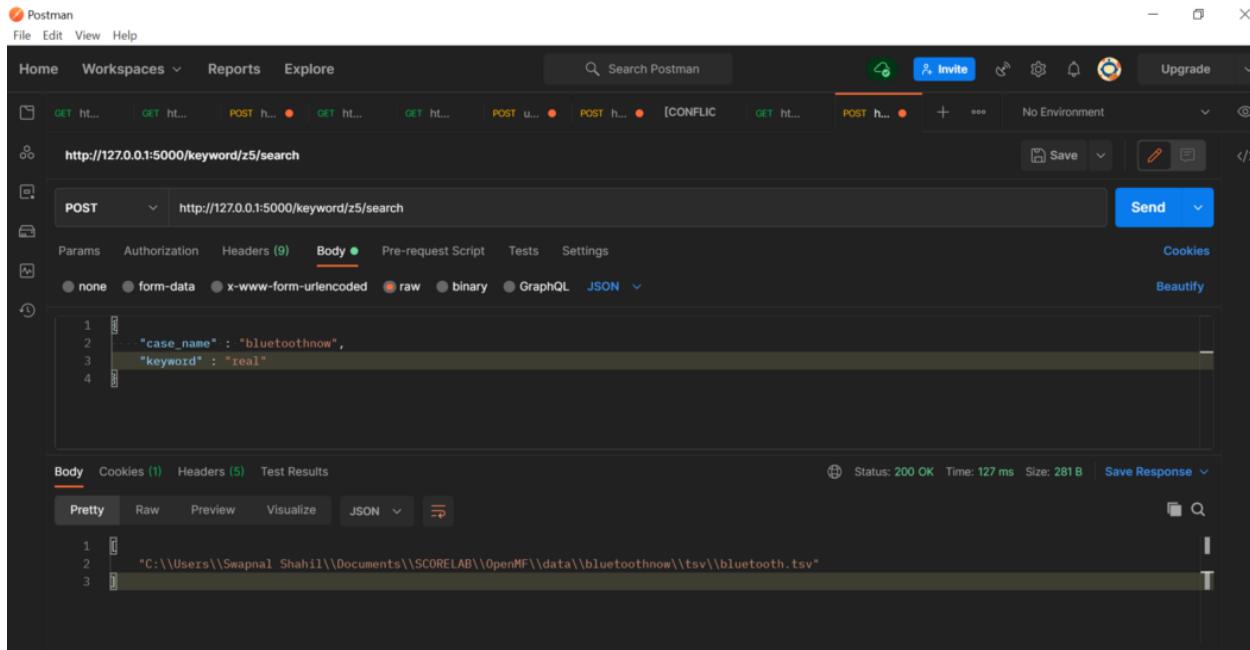
The response status is 200 OK, time 56 ms, size 698 B. The 'Pretty' tab in the results panel displays the following JSON output:

```

1
2 [
3     "C:\\Users\\Swapnal Shahil\\Documents\\SCORELAB\\OpenMF\\flask-backend\\api\\routes\\..../data/location20\\report/report.txt",
4     "C:\\Users\\Swapnal Shahil\\Documents\\SCORELAB\\OpenMF\\flask-backend\\api\\routes\\..../data/medianowl\\report/report.txt",
5     "C:\\Users\\Swapnal Shahil\\Documents\\SCORELAB\\OpenMF\\flask-backend\\api\\routes\\..../data/z4\\report/report.txt",
6     "C:\\Users\\Swapnal Shahil\\Documents\\SCORELAB\\OpenMF\\flask-backend\\api\\routes\\..../data/z5\\report/report.txt"
]

```

Keyword search from the whole database (above)



Keyword search from a particular case (above)

SCoRe Lab Contributions:

I have been contributing to this organization since last year and will continue as per need.

- [Merged] <https://github.com/scorelab/OpenMF/pull/192>
“Extraction of images data”
- [Merged] <https://github.com/scorelab/OpenMF/pull/182>
“Extract Bluetooth data”
- [Merged] <https://github.com/scorelab/OpenMF/pull/181>
“Extraction of browser history”
- [Merged] <https://github.com/scorelab/OpenMF/pull/175>
“Extraction of text messages(SMS)”

- [Merged] <https://github.com/scorelab/OpenMF/pull/47>
"Frontend documentation #47"
- [Merged]<https://github.com/scorelab/OpenMF/pull/44>
"fixed typo in README.md #44"
- [Open]<https://github.com/scorelab/OpenMF/pull/157>
"[Frontend] enabled lazy loading #157"
- [Open]<https://github.com/scorelab/OpenMF/pull/117>
"[frontend] Know more button fixed #117"
- [Issue]<https://github.com/scorelab/OpenMF/issues/46>
"Documentation needs update! #46"
- [Issue]<https://github.com/scorelab/OpenMF/issues/116>
"Know More Button not working in login page #116"
- [Issue]<https://github.com/scorelab/OpenMF/issues/174>
"Extraction of text messages"
- [Issue] <https://github.com/scorelab/OpenMF/issues/177>
"Extraction of Bluetooth data"
- [Issue] <https://github.com/scorelab/OpenMF/issues/178>
"Extraction of browser history"
- [Issue] <https://github.com/scorelab/OpenMF/issues/187>
"Extraction of EXIF data from images (like location, date, time, etc.)"
- [Issue] <https://github.com/scorelab/OpenMF/issues/170>
"Extract all data from rooted Android device not working"

Personal Information:

- Name: Swapnil Shahil

- Email: swapnalsahil@gmail.com
- GitHub: swapnalshahil
- LinkedIn: [swapnalsahil](https://www.linkedin.com/in/swapnalsahil/)
- Twitter: [@eulersgamma](https://twitter.com/eulersgamma)
- Duo: swapnalsahil@gmail.com
- Gitter nickname: swapnalshahil
- First language: Hindi, proficient in English
- Time zone: Indian Standard Time (GMT +5:30)
- Contact: +91 9205711402
- University: Indian Institute of Technology, Guwahati
- Year of study: 2nd year B. Tech.
- Major: Chemical Science and Technology (Batch of 2023)

Reference:

- [/scorelab/OpenMF/](#)

Questions

1. Are you a SCoRe contributor/ Have you contributed to SCoRe before?

Yes, I have contributed to SCoRe Lab Organization since December 2020. Yes, My contributions are in **OpenMF** and **Fact-Bounty** projects

2. How can we reach you (eg: email) if we have questions about your application?

Email: swapnalsahil@gmail.com

Gitter: swapnalshahil

Mob: 9205711402

3. What is your GitHub username(s):

[swapnalshahil](https://github.com/swapnalshahil)

Project Specific Questions

4. Which SCoRe GSoC project are you applying for (please submit separate applications for each project):

OpenMF - Analytics API

5. What do you plan to accomplish over this summer for this project?

- Generate dummy dataset from a rooted device.
- Extraction of the text messages.
- Extraction of browser history.
- Extraction of Bluetooth files.
- Extracting data (EXIF) from images(like location, date, and time)
- Extraction of location from the device.
- Keyword search - Implementation of keyword search to get all the cases that have the searched keyword.
- Implementing keyword search from cases.
- Maximum occurred common word/data between two cases.
- Returning all the common words between two cases.
- Filtering Cases by Date or range of date.
- Creating APIs for the above.

- Tag association and tag-based search like searching cases that are tagged with theft to get all the theft-related cases.
- Customized keyword search <kw1> + <kw2> + ... + <kwN>
- Analytics based on location.
- Implementing analytics on the front end.
- Presentation of observed analytics - graph, library, an API will be developed which will send the database data to the graph generating library to create required graphs and that will show in our OpenMF frontend.
- Presentation of observed analytics between two cases using graphs and tables.
- Writing Wiki and blogs.

6. If you have your own project to propose, please describe it here:

Not now but maybe in the future.

7. List down any plans you have during this summer.

I don't have any other plan for this summer, so I can give my whole day for this project.

8. Education:

- **University:** Indian Institute of Technology Guwahati
- **Department:** Chemical Science and Technology
- **Current year of study:** 2nd year of B.Tech
- **Programming courses:**
 - Object-Oriented Programming
 - Data Structures and Algorithm
 - Interactivity with JavaScript

- ReactJs
- Python and C++ Programming
- Git and Github

9. Do you have work experience in programming?

Yes, I did some projects of mine till now but contributed to many organizations on regular basis. One of my projects (project-corona) shows the current number of COVID19 { active cases, death count, recovered count } using bar graph of all countries.

10. Do you have previous open source experience? Briefly describe what you have done.

From my 1st year of my B. Tech I am involved with open-source. Also, I participated in Hactoberfest and able to create more than 10 valuable PRs. In my free time, I contribute to reactjs.org (Facebook React Community).

11. Tell one interesting fact about yourself.

I am a very social person and I love photography.