

# Swapnal Shahil

Mail: [swapnalsahil@gmail.com](mailto:swapnalsahil@gmail.com)

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

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

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

# 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 Python, React, Js, ES7, HTML, CSS with my interest in the web, and I have been contributing to this organization since December and will continue further in all my possible way.

## Analytics API and UI Development

### Research:

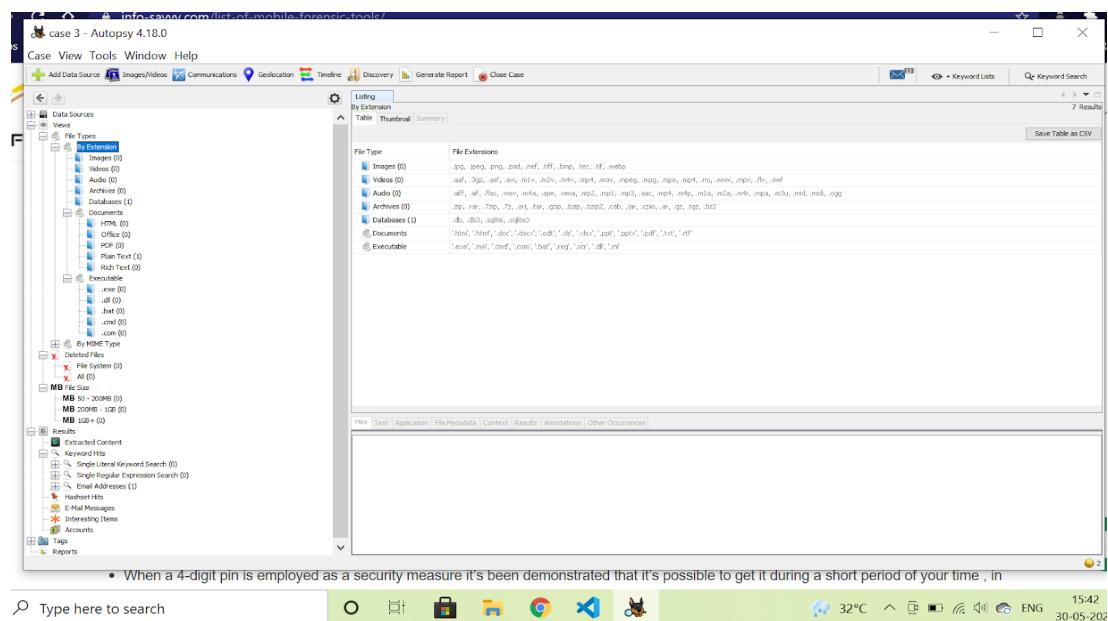
Available Analytics tool:

#### 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.

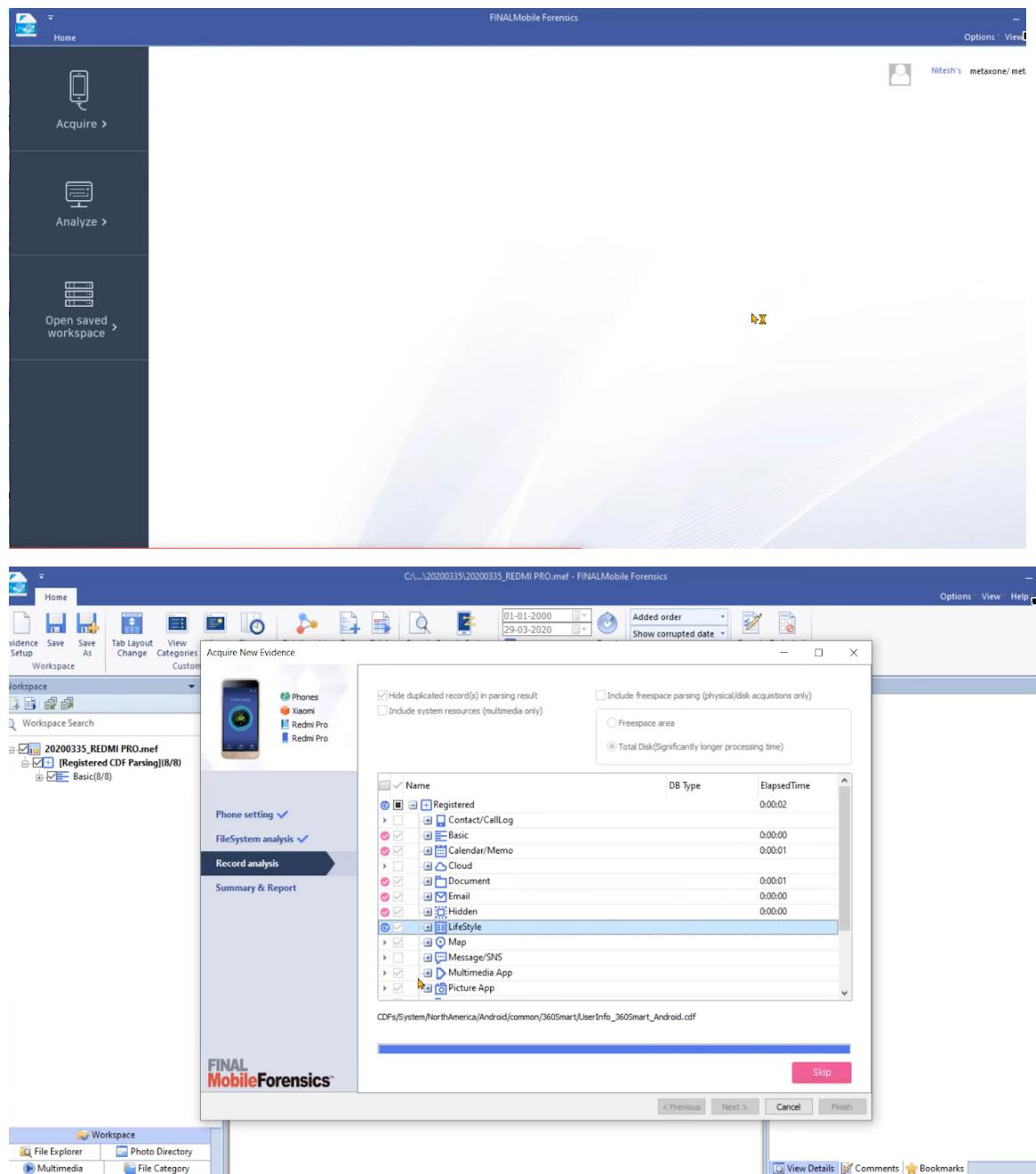


## 2.FINAL Mobile Forensics

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

## Features:

- 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 "Catalogs" section with "Device Info", "Apps", "Networking" (which is selected), "Wifi Information", "Internet Browsing" (which is also selected), and "DHCP Information". Below these are "Device Logs" and "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&id=1&sig=b224ab6acf5c55ffd6cb8c9bb24cce35ddb9a02d&id=;
1410338176	Android Browser	1	N/A	https://twitter.com/who_to_follow/import
1410338276	Android Browser	2	N/A	Request for Permission

At the bottom of the main content area, there are navigation buttons (back, forward, search) and a status message "Browsing History". The footer contains links for "Home", "Settings", and "About".

telephony communications such as Calls, SMS messages  
are individual application stores  
individual application file attributes, types, content and application usage timeline  
et ASCII and Uni  
Application usage  
h through applic  
more to come...

**Silensec's Mobile Forensics Investigator (MFI)**

test

Dashboard Catalogs

- Device Info
- Apps
- Networking
- Device Logs
- Communications
- Telephony

2/4 Telephone call log

Message ID	Status	Type	Date Send	Local Date	From/To	Seen	Message
1	OK	Incoming	14099903678	14099936841	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 information
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 5 times 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

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

communications such as Calls, SMS messages  
dual application stores  
individual application file attributes, types, content and application usage timeline  
and Uni  
ion usage  
h applic  
come...

**Silensec's Mobile Forensics Investigator (MFI)**

test

Dashboard Catalogs

- Device Info
- Apps
- Networking
- Device Logs
- Communications
- Telephony

3/4 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??

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

## 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"> <li>● Generate dummy dataset from a rooted device.</li> <li>● Extraction of the text messages.</li> <li>● Creating API which will show the extracted text messages.</li> <li>● Implementation from the command line so that we can directly extract SMS.</li> </ul>
2)	14 June - 20 June	<ul style="list-style-type: none"> <li>● Extraction of browser history</li> <li>● Extraction of Bluetooth files/data.</li> <li>● Creating APIs which will show the extracted files for the above.</li> <li>● Implementation from the command line to extract browser history</li> </ul>

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

#### Evaluations (July 12- July 16)

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

		• 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.

```

c:\ 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
database extraction completed...
smdb 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', '2021-06-09\r'], ['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)

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor Area:** The main area displays the contents of the 'sms.tsv' file, which is a tab-separated values file containing text message data. The columns include: \_id, thread\_id, address, person, date, date\_sent, protocol, read, status, type, reply\_path\_present, and several fields starting with 'seamless data exp'. The data consists of multiple rows of message details.
- Explorer Bar:** Shows the project structure under 'OPENIMF' (extractsms > tsv). The 'sms.tsv' file is currently selected.
- Bottom Status Bar:** PROBLEMS 1, OUTPUT, DEBUG CONSOLE, TERMINAL, and a PowerShell icon.

(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

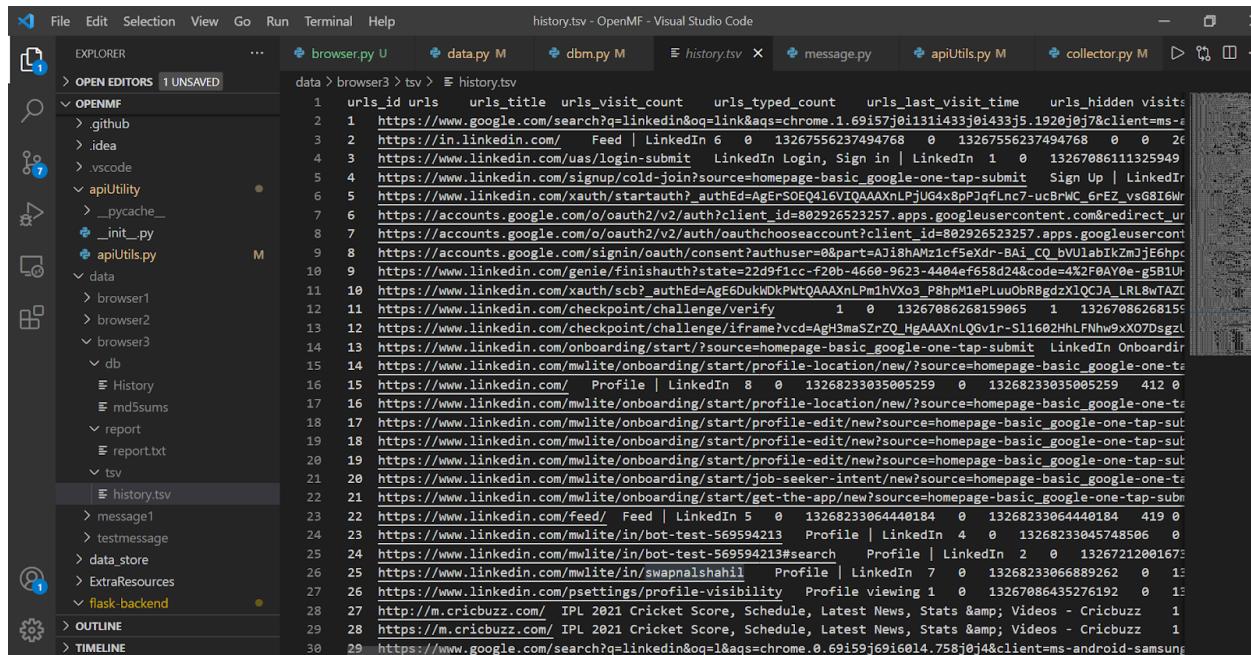
```
(venv) C:\Users\Swapnil Shahil\Documents\OpenMF>python collector.py --option browser --session_name browser4
* daemon not running; starting now at tcp:5037
* daemon started successfully
Shell permissions: root(su)

General Device Information

ADB serial: 3adad83b

Device model: samsung SM-G600FY
IMEI:
Android version: 6.0.1
Build number: MB29M.G600FYXU1BRD2
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'd 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/callog.db
[REDACTED]
```

### Extraction of browser data using CMD



```
File Edit Selection View Go Run Terminal Help history.tsv - OpenMF - Visual Studio Code
EXPLORER OPEN EDITORS 1 UNSAVED
OPENMF .github .idea .vscode apiUtility __pycache__ _init_.py apiUtils.py M
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.1.69157j01311433j01433j5.1920j0j7&client=ms-e
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=AgErSOEQ416VlQAAAXnLPjU64x8pP1qfLn7-uclBrWC_6rEZ_vsg816w
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
9 8 https://accounts.google.com/signin/oauth/consent?authuser=0&part=Aj18AMz2cf5fxdr-BAi_CQ_bVUlabbkZmjE6hp
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=AgE6DUKWDkPWTQAAAXnPm1hvXo3_P8hPM1ePluuObRBgdzx1LQCA_LRL8wTAZC
12 11 https://www.linkedin.com/checkpoint/challenge/verify 1 0 13267086268159065 1 13267086268159065
13 12 https://www.linkedin.com/checkpoint/challenge/iframe?vcd=AgH3maS7rzQ_HgAAAXnLQGvir-S11602HhLNh9xX07DsgzL
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/in/swapnilshahil Profile | LinkedIn 7 0 13268233066889262 0 13
27 26 https://www.linkedin.com/psettings/profile-visibility Profile viewing 1 0 13267086435276192 0 13
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=chrome.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.

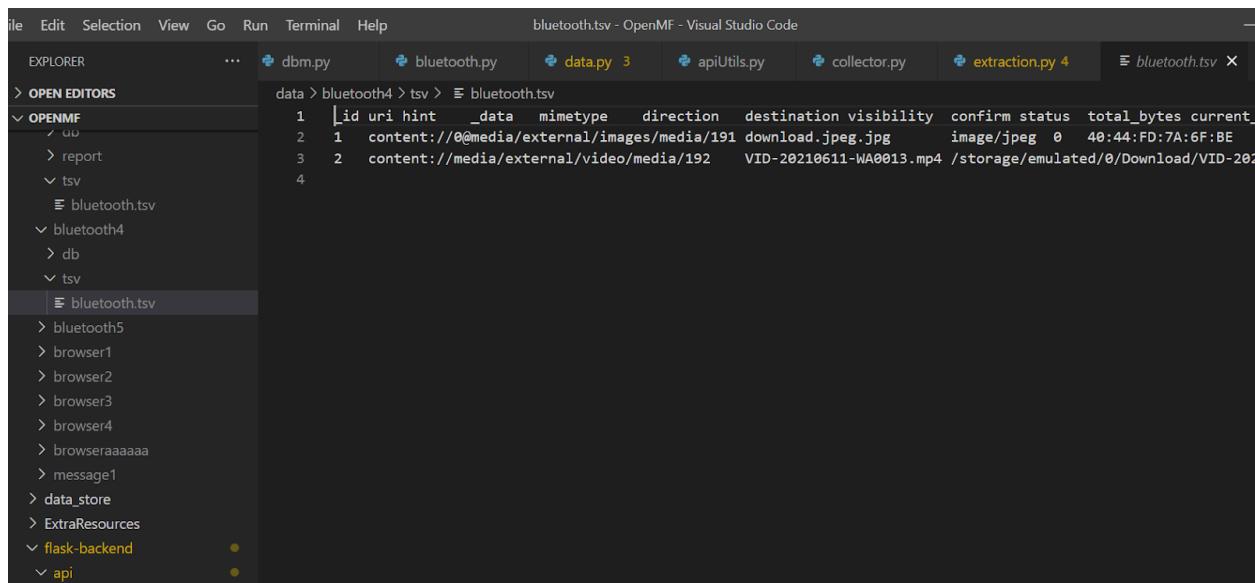
```
(venv) C:\Users\Swapnil Shahil\Documents\OpenMF>python collector.py --option bluetooth --session_name bluetooth
Shell permissions: root(su)

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'ed 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/mmsms.db
Extracting current db from: /data/data/com.android.providers.telephony/databases/mmsms.db
```

## Extraction of Bluetooth data using CMD



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.

```

File Edit Selection View Go Run Terminal Help mediadata.tsv - OpenMF - Visual Studio Code
EXPLORER OPENMF dbm.py M mediadata.tsv media.py U
OPENMF mediadata.tsv
media3
db
report
tsv
mediadata.tsv
media4
db
report
tsv
mediadata.tsv
message1
test1
data_store
ExtraResources
flask-backend
api
_pycache_
helpers
models
routes
_pycache_
_init_.py
mediadata.tsv
1 _id _data _size datetaken mime_type title description _display_name picasa_id orientation latitude longitude
2 20 /storage/emulated/0/Samsung/Image/5. Nightscape.jpg 903589 2014-08-01T19:48:14 image/jpeg 5. Nightscape
3 21 /storage/emulated/0/Samsung/Image/6. Bridge.jpg 522139 2014-08-01T19:48:14 image/jpeg 6. Bridge
4 22 /storage/emulated/0/Samsung/Image/7. Starlight.jpg 491165 2014-08-01T19:48:14 image/jpeg 7. Starlight
5 93 /storage/emulated/0/DCIM/Camera/20210530_111405.jpg 1360519 2021-05-30T11:14:05 image/jpeg 20210530_111405
6 97 /storage/emulated/0/WhatsApp/Media/WhatsApp Images/IMG-20210531-WA0001.jpg 241874 2021-05-31T11:36:07 image/jpeg
7 161 /storage/emulated/0/DCIM/Facebook/FB_IMG_1622884832764.jpg 44863 2021-06-04T16:23:52 image/jpeg FB_IMG_1622884832764
8 185 /storage/emulated/0/Pictures/Messenger/received_3853411084782291.jpeg 31927 2021-06-07T09:59:53 image/jpeg
9 189 /storage/emulated/0/DCIM/Camera/20210610_112615.jpg 1734850 2021-06-10T11:26:15 image/jpeg 20210610_112615
10 190 /storage/emulated/0/Download/download.webp 7052 2021-06-12T19:19:32 image/webp download
11 191 /storage/emulated/0/Download/download.jpeg.jpg 12734 2021-06-12T19:31:57 image/jpeg download.jpeg

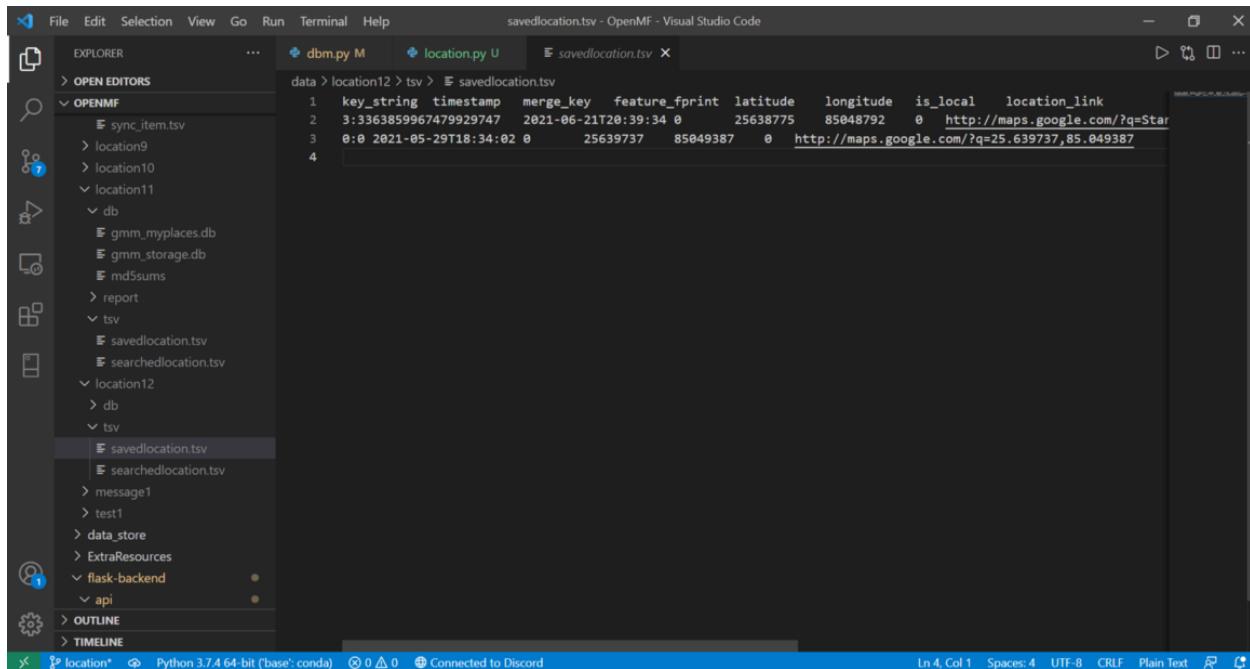
```

File media\* Python 3.7.4 64-bit (base: conda) 0 Connected to Discord

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



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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows a tree view of project files and folders. Key items include:
  - OPEN EDITORS:** dbm.py (marked with a blue dot), location.py (marked with a blue dot), savedlocation.tsv.
  - OPENMF:** sync\_item.tsv, location9, location10, location11 (expanded), db (expanded), gmm\_myplaces.db, gmm\_storage.db, md5sums, report, tsv (expanded), savedlocation.tsv, searchedlocation.tsv.
  - location12:** db, tsv (expanded), savedlocation.tsv, searchedlocation.tsv.
  - data:** location12 > tsv > savedlocation.tsv.
- Editor:** The 'savedlocation.tsv' file is open, displaying the following data:

key_string	timestamp	merge_key	feature_fprint	latitude	longitude	is_local	location_link
3:3363859967479929747	2021-06-21T20:39:34	0	25638775	85048792	0	<a href="http://maps.google.com/?q=Star">http://maps.google.com/?q=Star</a>	
0:0	2021-05-29T18:34:02	0	25639737	85049387	0	<a href="http://maps.google.com/?q=25.639737,85.049387">http://maps.google.com/?q=25.639737,85.049387</a>	

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. A POST request is being made to `http://127.0.0.1:5000/keyword/search`. The request body is set to `JSON` and contains the following JSON payload:

```

1
2 "keyword" : "samsung"
3

```

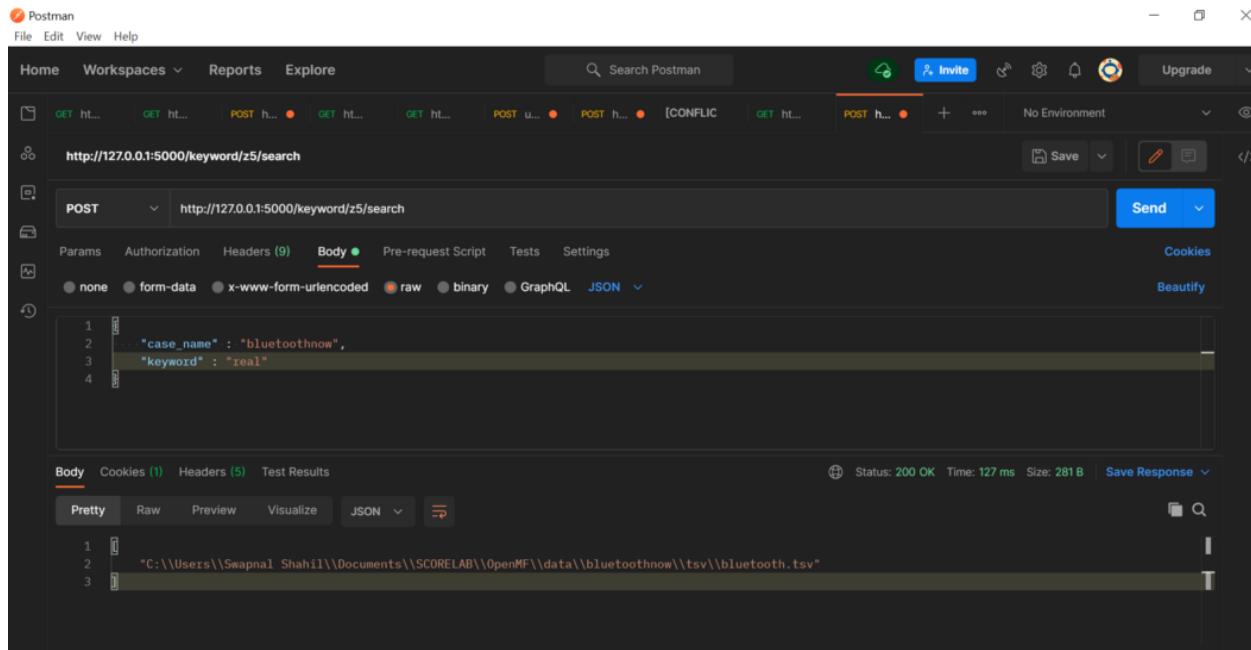
The response status is `200 OK`, time taken is `56 ms`, and the size is `698 B`. The response body is displayed in pretty JSON format, showing multiple file paths where the keyword "samsung" was found:

```

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

```

Keyword search from the whole database (above)



Keyword search from a particular case (above)

## (5 July - 11 July)

Work on the analytics part of this project and create APIs respectively to find common words between the two cases and the maximum word common between the cases with their frequencies.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

http://127.0.0.1:5000/common/Case1/bluetoothnow

POST http://127.0.0.1:5000/common/Case1/bluetoothnow

Params Authorization Headers (9) Body (selected) Pre-request Script Tests Settings

Body (Pretty, Raw, Preview, Visualize, JSON)

```

1
2   "case1": "Case1",
3   "case2": "bluetoothnow"
4

```

Body Cookies (1) Headers (5) Test Results

Status: 200 OK Time: 42 ms Size: 688 B Save Response

Common words between two cases.(above)

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

http://127.0.0.1:5000/common/words/browser2/Case1

POST http://127.0.0.1:5000/common/words/browser2/Case1

Params Authorization Headers (9) Body (selected) Pre-request Script Tests Settings

Body (Pretty, Raw, Preview, Visualize, JSON)

```

1
2   "case1": "browser2",
3   "case2": "Case1"
4

```

Body Cookies (1) Headers (5) Test Results

Status: 200 OK Time: 25 ms Size: 70.17 KB Save Response

Maximum common word with frequency.

## (12 July - 18 July)

Work on the analytics part to filter the cases by creating API and implementing third, fourth, fifth, and sixth-week work on the OpenMF front end. Use Redux for state management

The screenshot shows the 'Common Words' section of the OpenMF application. On the left, there is a sidebar with various navigation options: 'Common word' (selected), 'Key Word Search', 'Filter', 'File Explorer', 'case Tree', 'Analytics', 'Home', and 'Contact'. The main content area has two input fields labeled 'case1 \*' and 'case2 \*'. Below these fields is a green button labeled 'FIND COMMON WORDS'. A placeholder text 'Please provide Case name!' is visible below the input fields. At the bottom of the page, there is a footer bar with the text 'SCoRe Lab' and 'Contact us: [Email](#)'.

Common Word Page (above)

OpenMF

managementswapnal  
managementswapnal@gmail.com

Common word

Key Word Search

Filter

File Explorer

case Tree

Analytics

Home

Contact

SCoRe Lab

Contact us:

Keyword Search

keyword \*

case \*

FIND CASES

Please add keyword!

Keyword Search Page (above)

OpenMF

managementswapnal  
managementswapnal@gmail.com

Common word

Key Word Search

Filter

File Explorer

case Tree

Analytics

Home

Contact

SCoRe Lab

Contact us:

Filter Cases

From Date  
2021-07-01

To Date  
2021-07-10

Sat, Jul 10

July 2021

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

CANCEL OK

Filter Cases (above)

## (19 July - 25 July)

Work on “Tag association” and tag-based search like searching cases that are tagged with theft to get all the theft-related cases and on customized keyword search.

The image contains two screenshots of the OpenMF web application interface, showing different search and filtering options.

**Top Screenshot (Keyword Search):**

- The title bar shows multiple tabs: Email - SW..., scorelab/OpenMF - Gitter, Implementation of some analyti..., OpenMF, localhost:3000/keywordsearch, Apps, Rest, webd, Courses, GSoC, scorelab/scorelab..., Home | Google Su..., SCoRe Lab Open M..., OpenMF GSoC21..., Internship QuickLink..., and Reading list.
- The main header includes links for HOME, PROFILE, MEMBERS, CONTACT US, and DASHBOARD.
- The left sidebar has a user profile for "managementwapnal" and "managementswapnal@gmail.com". It includes icons for Common word, Key Word Search (which is selected), Filter, File Explorer, case Tree, Analytics, Home, Contact, and SCoRe Lab.
- The central search area has two input fields: "keyword \*" and "case \*". Below them is a green "FIND CASES" button. A placeholder text "Please add keyword!" is visible.
- The bottom right corner features a KAPWING logo with the text "Contact us: 🌐".

**Bottom Screenshot (Filter Cases):**

- The title bar shows multiple tabs: Email - SW..., scorelab/O..., [Frontend] / x, OpenMF/filter, OpenMF, javascript..., React Sele..., javascript..., localhost:3000/filter, Apps, Rest, webd, Courses, GSoC, scorelab/scorelab..., Home | Google Su..., SCoRe Lab Open M..., OpenMF GSoC21..., Internship QuickLink..., and Reading list.
- The main header includes links for HOME, PROFILE, MEMBERS, CONTACT US, and DASHBOARD.
- The left sidebar has a user profile for "managementwapnal" and "managementswapnal@gmail.com". It includes icons for Common word, Key Word Search, Filter (which is selected), File Explorer, case Tree, Analytics, Home, Contact, and SCoRe Lab.
- The central search area has a dropdown menu set to "Tags" and a search field for "tags \*". Below it is a green "FIND CASES" button. The text "No Case Found" is displayed.
- The bottom right corner features a KAPWING logo with the text "Contact us: 🌐".

Postman

File Edit View Help

Home Workspaces Reports Explore Search Postman Invite Upgrade

GET http... GET http... POST http... GET http... GET http... POST http... [CONFLICT] [CONFLICT] POST http... + No Environment

<http://127.0.0.1:5000/keyword/search/tags>

POST http://127.0.0.1:5000/keyword/search/tags

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 "tags": "theft"
```

Send Cookies Beautify

Body Cookies (1) Headers (5) Test Results Status: 200 OK Time: 44 ms Size: 530 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   "C:\\Users\\Swapnil Shahil\\Documents\\SCORELAB\\OpenMF\\flask-backend\\api\\routes\\..../data/POSTMANforreport2",
3   "C:\\Users\\Swapnil Shahil\\Documents\\SCORELAB\\OpenMF\\flask-backend\\api\\routes\\..../data/tagcase",
4   "C:\\Users\\Swapnil Shahil\\Documents\\SCORELAB\\OpenMF\\flask-backend\\api\\routes\\..../data/tagcase_cmd"
5 ]
```

Find and Replace Console Bootcamp Runner Trash

Postman

File Edit View Help

Home Workspaces Reports Explore Search Postman Invite Upgrade

GET http... GET http... POST http... GET http... GET http... POST http... [CONFLICT] [CONFLICT] POST http... POST http... + No Environment

<http://127.0.0.1:5000/keyword/custom/search>

POST http://127.0.0.1:5000/keyword/custom/search

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 "keyword": "samsung,2021-07-19,theft"
```

Send Cookies Beautify

Body Cookies (1) Headers (5) Test Results Status: 200 OK Time: 81 ms Size: 530 B Save Response

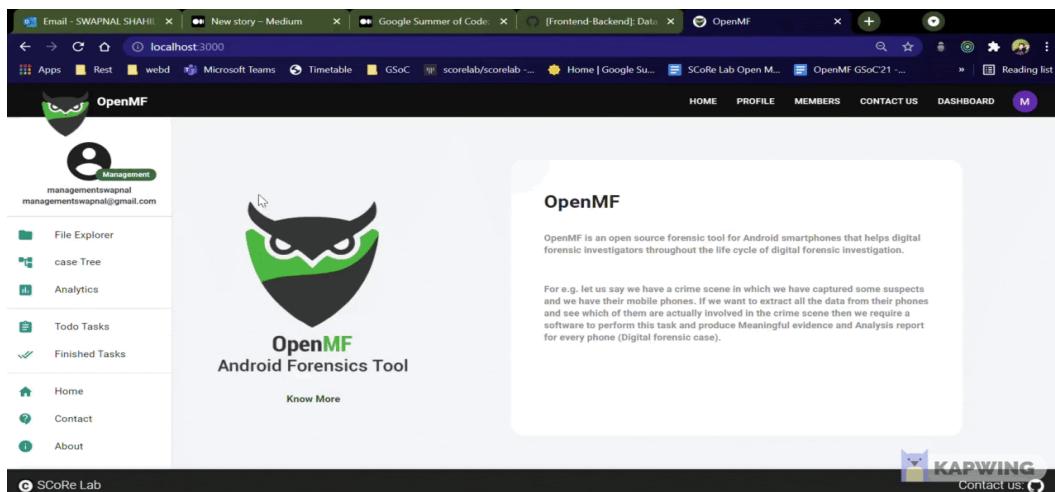
Pretty Raw Preview Visualize JSON

```
1 [
2   "C:\\\\Users\\\\Swapnil Shahil\\\\Documents\\\\SCORELAB\\\\OpenMF\\\\flask-backend\\\\api\\\\routes\\\\..../data/POSTMANforreport2",
3   "C:\\\\Users\\\\Swapnil Shahil\\\\Documents\\\\SCORELAB\\\\OpenMF\\\\flask-backend\\\\api\\\\routes\\\\..../data/tagcase",
4   "C:\\\\Users\\\\Swapnil Shahil\\\\Documents\\\\SCORELAB\\\\OpenMF\\\\flask-backend\\\\api\\\\routes\\\\..../data/tagcase_cmd"
5 ]
```

Find and Replace Console Bootcamp Runner Trash

## (26 July - 1 Aug)

Work on Data Visualization in the OpenMF frontend and try to present observed analytics through graphs, tables, maps of each case in the report section of OpenMF Analytics.



```

1
2   "case_name": "location_update5"
3

```

Status: 200 OK Time: 61 ms Size: 1.08 KB Save Response

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help' options. Below the navigation bar is a search bar labeled 'Search Postman'. The main workspace shows a list of recent requests and an environment named 'No Environment'. A specific POST request to 'http://127.0.0.1:5000/report/generalinfo' is selected. The 'Body' tab is active, displaying the following JSON payload:

```
1  {
2     "case_name": "location_update5"
3 }
```

Below the body, there are tabs for 'Params', 'Authorization', 'Headers (9)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is currently selected. On the right side, there are buttons for 'Save', 'Send', 'Cookies', and 'Beautify'. The response status is shown as 'Status: 200 OK Time: 24 ms Size: 984 B Save Response'. At the bottom, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. The bottom right corner shows the date '01-08-2021' and time '11:54'.

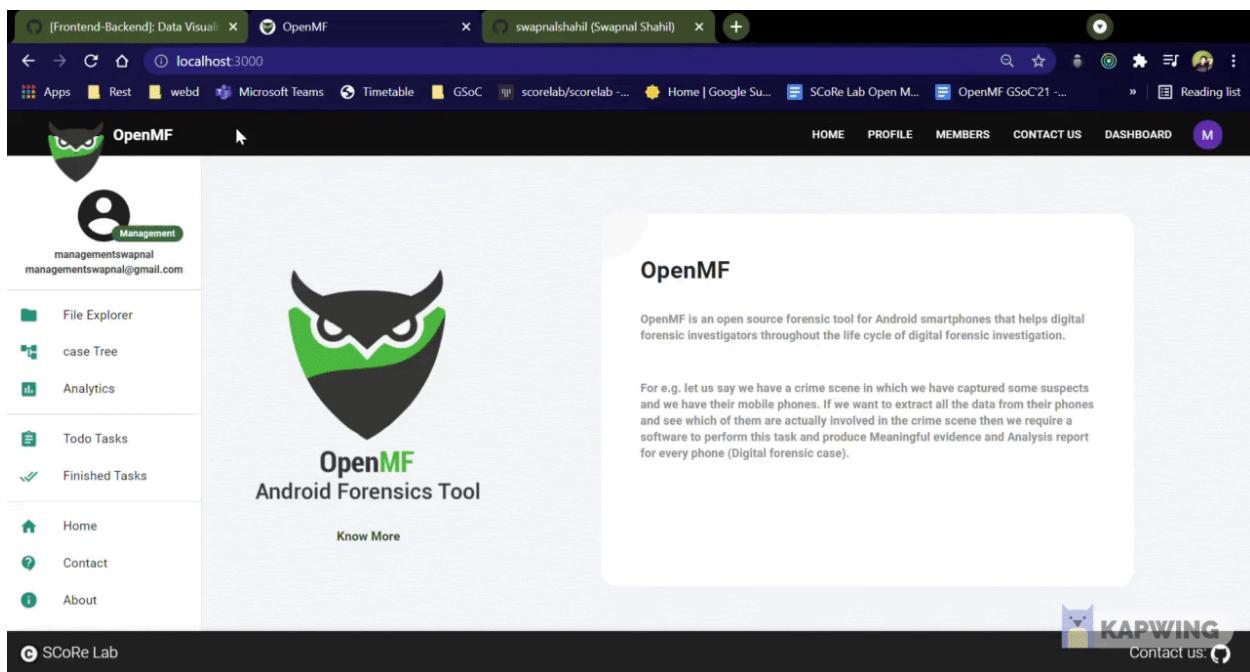
The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', and 'Help' options. Below the navigation bar is a search bar labeled 'Search Postman'. The main workspace shows a list of recent requests and a central panel for the current request. The current request is a 'POST' to 'http://127.0.0.1:5000/report/browserdata'. The 'Body' tab is selected, showing the following JSON payload:

```
1 | {
2 |   "case_name": "location_update5"
3 | }
```

Below the body, there are tabs for 'Body', 'Cookies (1)', 'Headers (5)', and 'Test Results'. The status bar at the bottom indicates a 'Status: 200 OK', 'Time: 56 ms', and 'Size: 52.54 KB'. The bottom navigation bar includes icons for 'Bootcamp', 'Runner', 'Trash', and a search bar.

## (2 Aug - 8 Aug)

Worked on Data visualization between two cases. Presentation of observed analytics between two cases using graphs, maps, and tables. For this, Create some APIs for representation on the OpenMF frontend. These will be responsible for fetching common details like call logs, text messages, locations, and browser histories from the extracted cases.



Also, create a location page that points to all the locations present in the extracted case and routed to the analytics section in the OpenMF frontend.

**OpenMF**

OpenMF is an open source forensic tool for Android smartphones that helps digital forensic investigators throughout the life cycle of digital forensic investigation.

For e.g. let us say we have a crime scene in which we have captured some suspects and we have their mobile phones. If we want to extract all the data from their phones and see which of them are actually involved in the crime scene then we require a software to perform this task and produce Meaningful evidence and Analysis report for every phone (Digital forensic case).

**KAPWING**  
Contact us: [@](#)

Also, Create an analytics landing page that briefly provides an overview of OpenMF Analytics.

**OpenMF Analytics**

OpenMF analytics is a mobile forensic analytics service offered by SCoRe Lab (Sustainable Computing Research Lab) that helps to analyse data extracted from Android Devices.

- Common Words : Search common between the data.
- Keyword Search : Search keyword to relate the cases.
- Filter : Filter cases date wise.
- Location : Find the searched location from case.
- General Info : Get the general informations of the device.
- Tag Association : Give tags to cases.
- Data Visualization : Visualize data using maps, graphs and tables.

Contact us: [@](#)

## (9 Aug - 15 Aug)

Create Wiki for SCoRe Lab OpenMF and document the whole work.

## 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/223>  
“Data visualization between two cases”
- [Merged] <https://github.com/scorelab/OpenMF/pull/224>  
“Analytics based on location”
- [Merged] <https://github.com/scorelab/OpenMF/pull/229>  
“Added some missing comments”
- [Merged] <https://github.com/scorelab/OpenMF/pull/218>  
“Data Visualization of Cases”
- [Merged] <https://github.com/scorelab/OpenMF/pull/210>  
“Implementation of some analytics API on front end OpenMF”
- [Merged] <https://github.com/scorelab/OpenMF/pull/207>  
“Create Filter API to get Cases by Date or range of date.”
- [Merged] <https://github.com/scorelab/OpenMF/pull/200>  
“List all common word/data between two cases”
- [Merged] <https://github.com/scorelab/OpenMF/pull/195>  
“Keyword Search”

- [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/186>  
“Extraction of location data from the android device”
- [Issue] <https://github.com/scorelab/OpenMF/issues/170>  
“Extract all data from rooted Android device not working”
- [Issue] <https://github.com/scorelab/OpenMF/issues/193>  
“Keyword search (for getting cases)”
- [Issue] <https://github.com/scorelab/OpenMF/issues/194>  
“Keyword search from cases”
- [Issue] <https://github.com/scorelab/OpenMF/issues/198>  
“Maximum common word/data between two cases.”
- [Issue] <https://github.com/scorelab/OpenMF/issues/199>  
“List all common word/data between two cases.”
- [Issue] <https://github.com/scorelab/OpenMF/issues/201>  
“Add comments in the worked file”
- [Issue] <https://github.com/scorelab/OpenMF/issues/205>  
“Create Filter API to get Cases by Date or range of date.”
- [Issue] <https://github.com/scorelab/OpenMF/issues/206>  
 “[Frontend]: Implementation on analytics section in management.”
- [Issue] <https://github.com/scorelab/OpenMF/issues/208>  
“Bug in API for getting list of cases”
- [Issue] <https://github.com/scorelab/OpenMF/issues/211>  
“Tag association”
- [Issue] <https://github.com/scorelab/OpenMF/issues/212>  
“Customized Keyword search”

- [Issue] <https://github.com/scorelab/OpenMF/issues/214>  
“Report.txt did not get stored when extracting from POSTMAN”
- [Issue] <https://github.com/scorelab/OpenMF/issues/217>  
“Data visualization”
- [Issue] <https://github.com/scorelab/OpenMF/issues/213>  
“Analytics based on location”
- [Issue] <https://github.com/scorelab/OpenMF/issues/219> “Data visualization between two cases”
- [Issue] <https://github.com/scorelab/OpenMF/issues/222>  
“Bug in the extraction of call logs and message”
- [Issue] <https://github.com/scorelab/OpenMF/issues/201>  
“Add comments in the worked file”
- [Issue] <https://github.com/scorelab/OpenMF/issues/230>  
“Wiki for OpenMF”

## Personal Information:

- Name: Swapnil Shahil
- Email: [swapnalsahil@gmail.com](mailto:swapnalsahil@gmail.com)
- GitHub: swapnalsahil
- LinkedIn: [swapnalsahil](https://www.linkedin.com/in/swapnalsahil)
- Twitter: [@eulersgamma](https://twitter.com/eulersgamma)
- Duo: swapnalsahil@gmail.com
- Gitter nickname: swapnalsahil
- 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](mailto:swapnalsahil@gmail.com)

Gitter: swapnalshahil

Mob: 9205711402

### **3. What is your GitHub username(s):**

[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.