future internet

MDPI

Article

Browser Forensic Investigations of Instagram Utilizing IndexedDB Persistent Storage

Furkan Paligu [1,*] and Cihan Varol [2]

[1] Computer Science Department, North American University, Stafford, TX 77477, USA
[2] Computer Science Department, Sam Houston State University, Huntsville, TX 77340, USA; cxv007@shsu.edu
* Correspondence: fpaligu@na.edu; Tel.: +1-832-230-5543

Abstract: Social media usage is increasing at a rapid rate. Everyday users are leaving a substantial amount of data as artifacts in these applications. As the size and velocity of data increase, innovative technologies such as Web Storage and IndexedDB are emerging. Consequently, forensic investigators are facing challenges to adapt to the emerging technologies to establish reliable techniques for extracting and analyzing suspect information. This paper investigates the convenience and efficacy of performing forensic investigations with a time frame and social network connection analysis on IndexedDB technology. It focuses on artifacts from prevalently used social networking site Instagram on the Mozilla Firefox browser. A single case pretest–posttest quasi-experiment is designed and executed over Instagram web application to produce artifacts that are later extracted, processed, characterized, and presented in forms of information suited to forensic investigation. The artifacts obtained from Mozilla Firefox are crossed-checked with artifacts of Google Chrome for verification. In the end, the efficacy of using these artifacts in forensic investigations is shown with a demonstration through a proof-of-concept tool. The results indicate that Instagram artifacts stored in IndexedDB technology can be utilized efficiently for forensic investigations, with a large variety of information ranging from fully constructed user data to time and location indicators.

Keywords: digital forensics; persistent storage; web browser forensics

Citation: Paligu, F.; Varol, C. Browser Forensic Investigations of Instagram Utilizing IndexedDB Persistent Storage. *Future Internet* **2022**, *14*, 188. https://doi.org/10.3390/fi14060188

Academic Editors: Joel Scanlan and Paul A. Watters

Received: 23 May 2022
Accepted: 14 June 2022
Published: 17 June 2022

1. Introduction

Technology is advancing very rapidly. As a result, the volume and complexity of digital data are growing at an accelerated rate. On average, an individual living in the U.S. spends approximately two hours on social media sites every day, while leaving a substantial number of digital artifacts on various sites [1]. The situation is not any different offshore. A striking example is the case of Max Schrems, an Austrian privacy activist, who made a request to see his personal information stored in Facebook servers and received 1222 pages of documents [2]. As the amount of data usage increases, so does the need to store it in innovative ways. This phenomenon creates a heavy burden on all individuals and forensic investigators, as the ways we communicate and utilize information are subject to constant change. Therefore, renovation and diversification of the technologies subject to digital forensics is a constant obligation.

IndexedDB is a fairly new technology utilizing NoSQL (Not Only SQL) transactional database systems to provide persistent storage in web browsers [3]. It can provide over 50 MB of storage for each origin (distinct protocol, domain, host, and URL to access a source on the web) [4]. This is over a minimum of 45 MB of increased data storage compared to previous Web Storage technology [5]. Due to advantageous storage capability and fast access to persistent data with the utilization of JavaScript and JSON (JavaScript Object Notation) objects [6], the utilization of IndexedDB is increasing rapidly [7]. One of the websites that recently joined this trend is the popular social media site Instagram, which has over 1074 billion users worldwide [8].

This paper scrutinizes the Instagram-generated data stored in IndexedDB technology for its digital forensic value. In particular, the extraction, processing, and presentation of artifacts are scrutinized for Mozilla Firefox implementation of IndexedDB and verified through another implementation utilized by Google Chrome. Additionally, a proof-of-concept tool is developed for the application of the given techniques. In this scope, the paper raises two research questions.

- Is it possible to obtain information about the actions of a suspect on Instagram through artifacts populated in IndexedDB storage?
- Are the existent IndexedDB data convenient for construction of time frame analysis for the actions of Instagram users?

The rest of the paper is structured as follows: presentation of related work, background on IndexedDB, methods of experimentation, results, discussion, description of the proof-of-concept tool, and conclusions.

2. Related Works

Instagram web browser artifacts left on personal computers have been covered in a notable number of studies [9–11]. However, these studies were cut short of web persistent storage forensics, which is a relatively new area and has produced only a handful of papers in the last decade [7,12–17]. Additionally, Instagram forensics has been the subject of several research papers with a broader scope of social media forensics. These papers covered various aspects such as network traffic analysis [18], mobile application investigations [19,20], user behavioral analysis [21], digital image/video analysis [22], and criminal impersonation [23].

Chang and Yen [9] conducted a study that analyzes the Instagram artifacts on Android and Windows 10 platforms. It further diversified the artifacts with the examination of different web browsers. A broad range of artifacts from pictures to user activities such as liking, commenting, and tagging was analyzed. The study pointed out that the artifacts available in different browsers show significant differences. This is attributed to the variant privacy mechanisms of web browsers. Despite covering valuable sources such as log files, history records, and temporary network file artifacts, the study does not provide considerable information on persistent Web Storage technologies such as IndexedDB.

Ghafarian and Keskin [10] had a particular forensic focus on the artifacts of Facebook and Instagram obtained from Windows hibernation files. A detailed discussion is provided for social media forensics in the cases where it is not possible to obtain information from the suspect's hard drive. The paper shares detailed techniques for creating a connection between the suspect and the extracted artifacts. However, IndexedDB and similar persistent storage technologies are not covered in the study.

Jadoon et al. [11] scrutinized the forensic resistance of Tor browser in a Windows virtual environment. The experiment utilized a list of activities to populate data that included Instagram user activities such as following, visiting accounts, and liking pictures. The artifacts of visited links were successfully recovered, while artifacts from comments and liked pictures could not be recovered. The examined technologies are listed as the registry and main memory of the virtual drive, which did not include persistent storage technologies.

Kimak et al. [12] gave a detailed examination of IndexedDB in Mozilla Firefox and Google Chrome browsers. The digital forensic tool Encase was utilized to create a forensic image and subsequently extract and examine the subject artifacts. Significant issues such as clear-text (unencrypted) data storage are presented with potential solutions. The paper examines IndexedDB within the context of IndexedDB API, which was later updated to further versions of IndexedDB API 2.0 in 2018 [13] and the working draft of Indexed Database API 3.0 in 2021 [14]. Additionally, the content under scrutinization is limited to generic artifacts that are now outdated.

Paligu et al. [7] conducted research focused on IndexedDB forensics and security without targeting Instagram artifacts. The paper included a proof-of-concept tool called BrowStExPlus to demonstrate the findings. Paligu and Varol [15] expanded this work

to demonstrate the forensic investigation techniques of IndexedDB for WhatsApp web artifacts contained in the Google Chrome browser. A tool called BrowSwEx was developed to extract and present the artifacts. However, the SQLite implementation of IndexedDB was only partially covered in the first study. Additionally, the specifications of IndexedDB were changed with the new version of the API.

Walnycky et al. [16] acquired and analyzed artifacts from network packets and Android end devices utilizing content from twenty different applications including the Instagram Android application. The research aimed at a full or partial reconstruction of data artifacts. The images sent from the Instagram application were successfully intercepted in the network traffic. This paper covers only the Android application of Instagram and not information about persistent storage analysis.

Al Mushcab and Gladyshev [17] investigated efficient techniques to eliminate challenges in forensic analysis of Instagram and Path social media applications on iPhone devices. The process included a complete process of data population, image acquisition, and automatic examination. It is concluded that the internal memory of the iPhone 5s device contained artifacts from the social media applications, even though they are not installed in the internal memory.

Pambayun and Raidi [18] particularized the mobile investigations of Instagram on Android devices. The research applied the investigation stages of the Digital Forensics Research Workshop (DFRWS) with the utilization of the OXYGEN mobile forensics tool and JSON Viewer. The paper concludes on the sufficiency of the forensic tools to obtain artifacts from chat sessions that are extracted from both internal and external memory. The paper presents a systematic look into Android device investigations of Instagram. However, the shared methodology is slightly short on detail to provide an extension to other platforms.

Seo et al. [19] called attention to information exposure on Instagram pages with potential threats such as impersonation, fraud, and copyright violation. A reverse-engineering method is applied to obtain personal information from Instagram pages and data sources such as cache, SQLite files. Several tools including ADB (Android Debug Bridge) and DB Browser for SQLite were utilized during the examination. It is concluded that the user behavior on Instagram can be analyzed by arbitrary users with little effort. The applied method is also suggested as a digital forensic examination technique.

Douglas [20] conducted a detailed investigation on the images handled by Instagram. Various best practices such as the selection of image quality during extraction are shared to provide a framework for obtaining reliable data. A cross-platform experiment, questioning how an image from an Android device would look when uploaded to an Apple device, is made available as Supplementary Data. As the primary outcome, the essentials of a set of techniques to identify whether an image is original and authentic or downloaded from Instagram are laid out. Additionally, the characteristics of Instagram processed images are presented as a framework for future research.

Zarei et al. [21] analyzed the impersonators targeting Instagram users to construct a technique of impersonation identification. A large amount of data are utilized with both fake and genuine accounts, where users were divided into clusters based on their profile characteristics. The paper presents findings that indicate a considerable amount of political interest in bot-like clusters. It is also discussed that the interest levels of bot-like clusters in news agencies and sports stars were not significantly different.

Kumar and Karabiyik [22] focused on the Instagram artifacts of Android in vanish mode, where the messages, memes, and pictures disappear when the chat is closed. The question of whether the vanished messages could be obtained by forensic investigators is addressed. Artifacts are extracted from sources such as direct.db and rendered videos from the Android file storage location of Instagram. It is discussed that messages, reels, videos, and some other media items can be uncovered from the direct.db-wal file. The study is focused exclusively on the Android artifacts, and IndexedDB storage of web browsers is not addressed.

Quan et. al. [23] focused on Instagram photos and their sensor pattern noise (SPN), which is described as a feasible device fingerprint. Two groups of filters were applied, where SPN was either attenuated or preserved. About twenty-thousand images were experimented on, and over 96% of accuracy was achieved in differentiating filters. The study gives good insight into the Instagram image filters. However, web forensics are not covered.

The overall focus of research on Instagram web and mobile application forensics is diversified in miscellaneous aspects, including both user interaction and data artifacts. However, to the best of our knowledge, there is not any work conducted on IndexedDB and Instagram that can be a practical guide for cases under investigation. Additionally, no studies have focused on the extraction of IndexedDB 3.0 artifacts from SQLite implementation of Mozilla Firefox.

3. Background for IndexedDB and LevelDB

Even though legacy client-side storage technologies have been present in web browsers for twenty-five years [24], their structure has undergone a major change with the introduction of persistent storage. As was pointed out in the literature review section, the coverage of these technologies in academic work is less than ideal. In this section, a background is provided for the emphasized technologies and their relevant extensions to provide a better understanding.

3.1. Client-Side and Server-Side Storage

Two principal options are available for storing information for a web origin: client-side storage and server-side storage. Both options carry advantages and disadvantages [25,26]. Predominantly, the security of the client-side information is inevitably dependent on the user and the web browser [26]. Nevertheless, server-side storage utilizing client-side code such as JavaScript must send data forth and back, a process that generates increased network traffic. Furthermore, the scalability of the applications is dependent on the choice of storage techniques. When a substantial amount of data are stored on the server, the performance is directly proportional to the changes in the number of active users [27]. The most suitable option depends on the specifications of the application. A commonly accepted practice is to utilize server-side storage for sensitive information when less vulnerable information is stored on the client-side [28].

Legacy client-side storage technologies such as cookies provide small-scale capacity compared to newer technologies such as Web Storage and IndexedDB [29]. Therefore, their utilization and the types of data they contain differ considerably [30]. Table 1 gives a summary of the client-side storage technologies with their capacities. The emergence of persistent storage technologies with fast access to larger sizes of data profoundly changed the content of the data stored on the client-side. Therefore, its significance for forensic researchers has been heightened over time.

Table 1. Client-Side Technologies and Storage Capacity.

Technology	Storage Size	Notes
Cookies	4 KB	Legacy client-side storage technology
Web Storage	5 MB	Predecessor of IndexedDB that is still actively in use
Session Storage	5 MB	Non-persistent storage
IndexedDB	>50 MB	Bridge between Web Storage and Cache API
Cache API	>500 MB	Currently does not have considerable content in Alexa top 20 websites

3.2. IndexedDB

IndexedDB is a persistent NoSQL transactional database technology that takes advantage of client-side storage for web applications. It is fast and highly efficient, since B-trees are heavily utilized in its structure. B-trees enable fast manipulation of data on databases of considerable size [31]. With IndexedDB, entire databases can be employed for each web origin in a consistent structure, even within different platforms. This consistency is ensured by the specifications shared by W3C (World Wide Web Consortium) [32]. Currently, W3C published three versions of IndexedDB, with the latest one, IndexedDB API 3.0, being released in 2021 [14]. Moreover, it was highly adopted by major web browsers in a relatively short amount of time [33]. Table 2 presents the dates that major web browsers started their support for IndexedDB technology. These browsers constitute over 83 percent of their market [34]. The implementation technologies of the browsers are also listed in Table 2.

Table 2. IndexedDB Support and Technology in Web Browsers.

Browser	Support	Underlying Technology
Google Chrome	2012	LevelDB
Mozilla Firefox	2011	SQLite
Microsoft Edge	2015	LevelDB
Opera	2013	LevelDB
Internet Explorer	Only partial support	.dat file format

SQLite is prevalently seen in the preponderance of storage technologies, including the ones providing fundamental functionality to the browsers such as history and bookmarks. Similarly, it is the underlying technology for IndexedDB in Firefox browsers. Contrarily, Google Chrome constructed IndexedDB over LevelDB technology [35,36]. In various benchmark and experimental research, LevelDB has proved more efficient and secure compared to SQLite in terms of fast operations on key-value pairs and batch updates [37]. Additionally, LevelDB provides extra security for IndexedDB by keeping its .ldb files locked in accordance with the SOP (Same Origin Policy) [38]. As is seen in Table 2, the pioneering implementation of Google Chrome has been adapted through most of the modern browsers for IndexedDB, making SQLite the second-most-popular implementation. The API standards of IndexedDB published by W3C provide standard management for the developers through JavaScript code. In other words, even though the underlying technology of the IndexedDB might differ between browsers, its control is identical.

There is a level hierarchy in the IndexedDB API. The highest level in this hierarchy is a database. All databases are associated with a version that helps the server keep the storage up to date with updates. If developers introduce updates to the database, a function called onupgradeneeded is called to regenerate the database and its lower-level structures. Subsequently, databases contain object stores that are analogous to tables in traditional databases [39]. An example of IndexedDB control through JavaScript code can be seen in Code 1 and Code 2 with the API 3.0 specifications. Code 1 demonstrates the initiation process in three steps of establishing a database, creating an object store, and populating it with data. A data retrieval is similarly easy with the utilization of the read function over transactions. The function access is handled with onsuccess and onerror functions as demonstrated in Code 2, with steps of establishing a transaction, creating a request, and handling the return.

Code 1. Initiation of Database

```
// Step 1—establishing a database
const firstRequest = indexedDB.open("IndexedDBDemonstration");
let DemonstrationDatabase;
// onupgradeneeded is called when database does not exist or an upgrade is needed.
firstRequest.onupgradeneeded = function() {
const DemonstrationDatabase = firstRequest.result;
// Step 2—creating an object store. keyPath defines how the data will be indexed
const firstStore = DemonstrationDatabase.createObjectStore("users", {keyPath: "id"});
// Adding additional index
const nameIndex = firstStore.createIndex("by_name", "name", {unique: true});
// Step 3—adding data
firstStore.put({name: "John Doe", id: 120134});
firstStore.put({name: "Jane Doe", id: 120135});};
// onsuccess is called when database already exist and does not require any upgrade
firstRequest.onsuccess = function() {
DemonstrationDatabase = firstRequest.result;
};
```

Code 2. Data Retrieval

```
// Step 1—establishing a transaction
const firstTransaction = DemonstrationDatabase.transaction("users", "readonly");
const secondStore = firstTransaction.objectStore("users");
// Step 2—creating a request
const secondRequest = secondIndex.get("John Doe");
// Step 3—handle the returned data or error
// Without error
secondRequest.onsuccess = function() {
const ourResults = secondRequest.result;
// Handling data returned
};
// With error
secondRequest.onerror = function(event) {
// Handling the error
};
```

4. Materials and Methods

The research questions in this paper are tested with two hypotheses.

Hypothesis 1 (H1). *Data generated in IndexedDB storage can be utilized to obtain information about the actions of a suspect on Instagram.*

Hypothesis 2 (H2). *Instagram data artifacts in IndexedDB storage are suitable for utilization in time frame analysis for the actions of its users.*

4.1. Experimental Design

A single-case pretest–posttest quasi-experiment was carried out to test the hypotheses. The methodology applied in the paper resembles the experimental design put forward by Cook and Campbell [40]. A single subject was measured before and after being put through an adapted treatment. A comparison of the measurements is provided to disclose the change obtained by the treatment. Artifacts inherently present in the IndexedDB storage locations were discovered with the pretest experiment. The artifacts generated by the application of the treatment were identified with the comparison of the results obtained after the pretest and the treatment. In simpler terms, the difference between the

measurements is the proof that the discovered artifacts are the outcome of the treatment. The experimental process is displayed in Figure 1.

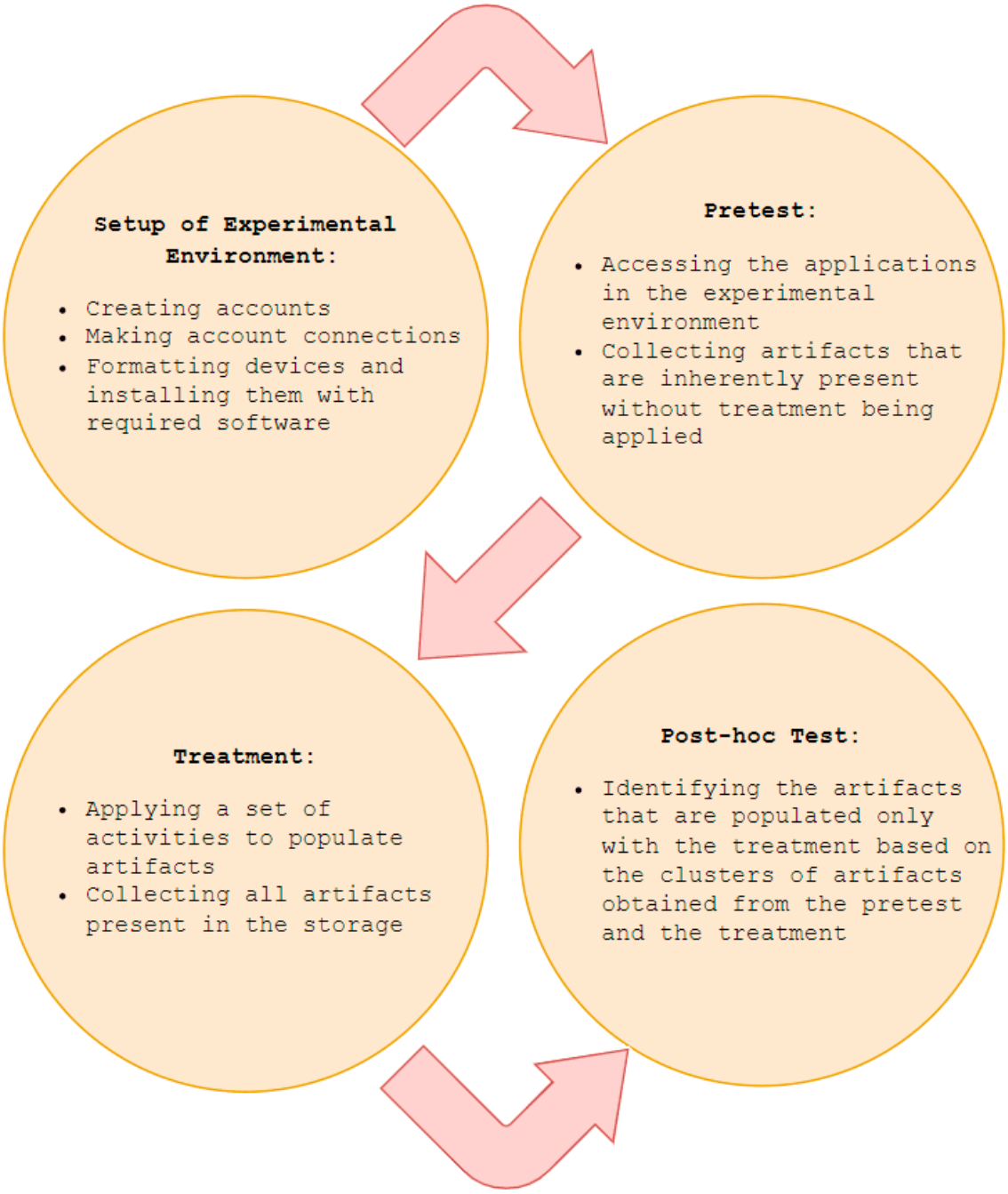


Figure 1. Experimental Process.

The subject provided in this research is the Instagram web application. The experimental environment is an HP Laptop that operates a Windows 10 Home Operating System with Mozilla Firefox 89.0.2 (64-bit) and Google Chrome v91.0.4472.124 (64-bit) browsers installed. Additionally, a Samsung s20 FE Android device was utilized to create the Instagram accounts as part of the environmental setup procedure. The experiment was applied twice, once on Mozilla Firefox and once on Google Chrome, for the verification of the data obtained from Mozilla Firefox SQLite storage. In order to set the experimental environment prior to the quasi-experiment, the following steps were carried out.

- Three different Instagram Accounts were created with Samsung Android device (Phone1). Table 3 displays the personal information details of the accounts, which were later used to determine their availability in the IndexedDB storage.

Table 3. User Accounts Utilized in Experiments.

Personal Information	Account1	Account2	Account3
Name	Forensic Researcher 1	Forensics Researcher 2	Forensic Researcher 3
Username	forensicreasearchaccount1	forensicresearchaccount2	forensicresearchaccount3
Website	x	http://forensicresearcher2.com accessed on 22 May 2022 (Dummy info)	http://forensicresearcher3.com accessed on 22 May 2022 (Dummy info)
Bio	x	Bio of Forensic Researcher 2	Bio of Forensic Researcher 3
Email	forensicresearchaccount1@protonmail.com	forensicresearchaccount2@protonmail.com	forensicresearchaccount3@protonmail.com
Email Confirmation	Not Confirmed	Confirmed	Confirmed
Phone Number	x	+33-4-64-03-67-89 (Randomly generated)	x
Phone Number Confirmation	x	Not Confirmed	x
Gender	x	Male	Female

- Account1 and Account2 were added as followed connections through the Android application.
- Account2 and Account3 were added as followed connections through the Android application. (No connection was created between Account1 and Account3.)
- In Account1 and Account2, a public account (Account4) was added as a followed connection to increase the scope and diversity of the available data. The idea is that the experimental accounts can overlook some data that exist in an operational account. Account4: awesome.photographers [41].
- Windows 10 computer (PC1) was formatted and installed with Mozilla Firefox (Browser1) and Google Chrome (Browser2) browsers.

4.2. Pretest

The artifacts inherently found in Mozilla Firefox and Google Chrome browsers were tested with the following procedures.

- Instagram Web Application was accessed through Browser1 and Browser2 without logging in to accounts.
- The connection was left idle for a time of fifteen minutes.
- The artifacts were collected from IndexedDB storage locations of Browser1 and Browser2 in PC1.

4.3. Treatment

A set of activities were designed according to the observation of common user behavior on Instagram. These activities constitute the treatment of the research. The observations were made based on the stored data of activity from five Instagram accounts of volunteers. Three of the volunteers were students in the computer science department, whereas two were faculty, again in the computer science department, of a university in Stafford, Texas. The listed activities constitute over ninety-eight percent of all activities of the volunteers in the subsequent nine days of the collection.

- Instant private messaging
- Sending messages with video and picture contents

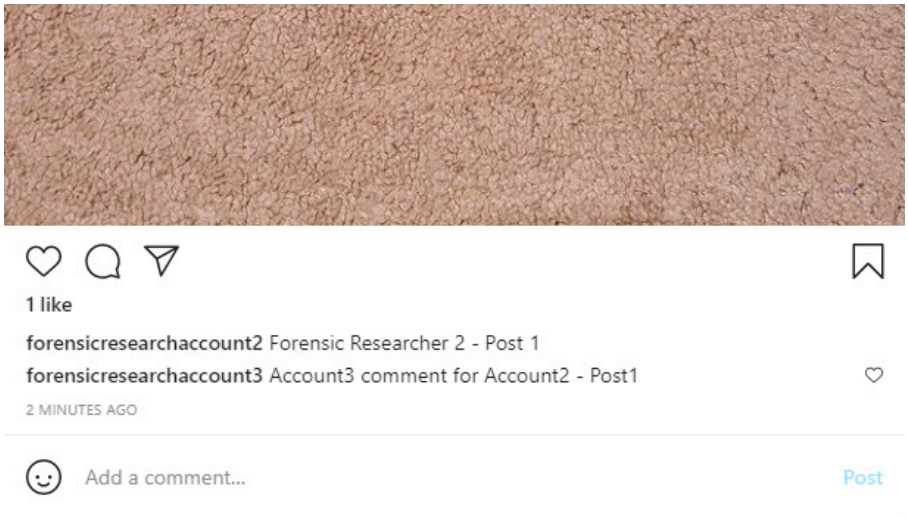
- Displaying messages with video and pictures received from other users
- Adding stories
- Displaying stories
- Visiting profiles
- Displaying recently added posts of followed connections on the home page
- Commenting on posts of followed connections
- Liking posts of followed connections
- Discovering new accounts through the Explore page
- Searching an account with its name
- Adding a post with graphic content to personal account.

Proceeding from the observation-based user activities, the treatment procedure of the experiment was established with the following steps:

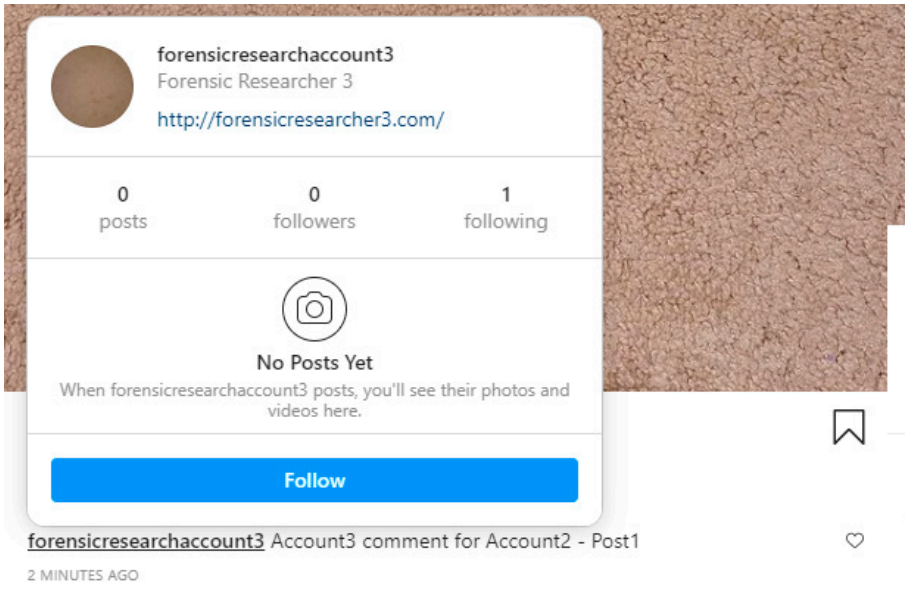
- Account2 was logged in from Phone1
- A random picture of a carpet was added to Account2 with Phone1, including the description "Forensic Researcher 2—Post 1"
- A random picture of a ceiling was added to Account2 as a story
- A message with the content "Message1—Account2 to Account1" was sent from Account2 to Account1 from messages page through Phone1
- Account2 was logged out from Phone1
- Account3 was logged in from Phone1
- Post1 of Account2 was liked with Account3 on Phone1
- Post1 of Account2 was commented on with the following content: "Account3 comment for Account2–Post1" by Account3 on Phone1
- Account3 was logged out of Phone1
- Account1 was logged in from Browser1 in PC1
- Account1 home page was displayed while scrolling down to display the posts and comments made by Account2 and Account3 on Browser1
- Story1 of Account2 was displayed on the home page
- An emoji reaction was added to Story1 of Account2
- The messages page was accessed and the message from Account2 was displayed with Account1 on Browser1
- A message with the content "Message2—Account1 to Account2 with emoji content: ☹ ⊖ ㊙" was sent as a reply to Message1
- Account1 home page was accessed and a recent story from Account4 was displayed through Account1 on Browser1
- The explore page was accessed through Account1 on Browser1
- The words "Forensics Researcher 2" were entered on the search-box of Account1 on Browser1
- The profile page of Account2 was accessed through Account1 on Browser1
- The procedure followed with Account1 on Browser1 was repeated on Browser2
- The procedure, where Account1, Account2, and Account3 were set as public accounts, was repeated with Account3 as a private account Figure 2.

Additionally, observations from the first extractions of the experiment showed that more steps in the treatment are needed to obtain additional artifacts. Firstly, it was determined that there were changes in the stored information from Account3 when its comment on the posts was displayed with hovering the mouse and when the profile page was visited. Therefore, two additional extractions were performed. In total, three cases where an extraction was performed are as follows:

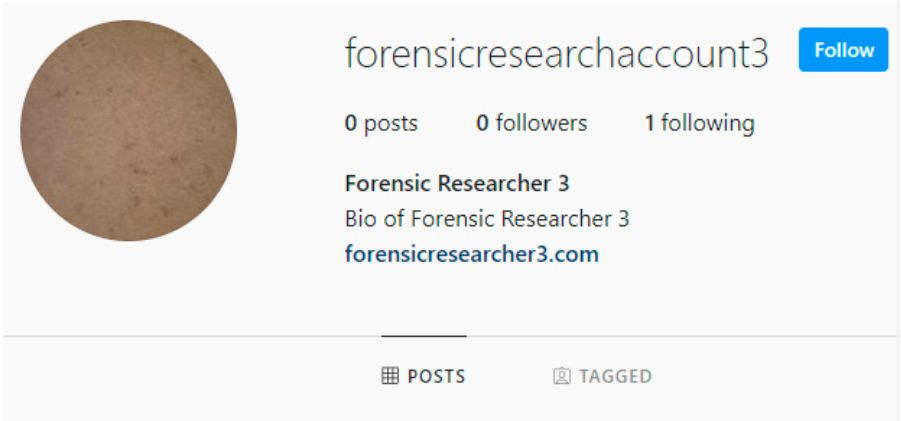
- When only the comment of Account3 was displayed on the home page but no interactions with the Account3 profile was taken
- When the brief profile information of Account3 was displayed with hovering the mouse over the profile section of the comment
- When Account3 profile page was visited



(**a**)



(**b**)



(**c**)

Figure 2. (**a**) Comment is Displayed without Profile Interactions, (**b**) Comment is Displayed with Profile Displayed Through Mouse Hovering, (**c**) Commenting Profile is Visited.

Secondly, the first observations on the record of the relationships showed an attribute called "stable". In order to discover the actions that would result in changes to this attribute, the following steps were added to the treatment with an extra extraction to test the changes to the values at each step. These additional steps can be listed as follows:

- Account3 was blocked, and Post1 of Account2 was displayed on the home page of Account1
- Account3 was unblocked, and Post1 of Account2 was displayed on the home page of Account1
- Account3 was restricted, and Post1 of Account2 was displayed on the home page of Account1
- Account3 was unrestricted, and Post1 of Account2 was displayed on the home page of Account1

4.4. Post-Hoc Test

In accordance with the procedures of pretest–posttest quasi-experiments, an independent test was conducted without the treatment following the experiment's environmental set-up. Consequently, the artifacts created in PC1 for IndexedDB were tested after the treatment was applied. The comparison of the results from before and after the treatment allowed the authors to isolate artifacts that can be tied exclusively to the treatment. Therefore, the results section presents the observation of the artifacts created with the treatment and only with the treatment in the IndexedDB storage of PC1.

5. Results

In order to test the hypotheses of the paper, the artifacts created by the treatment of the quasi-experiment were evaluated. The evaluations targeted artifact significance to the forensic investigations and suitability to the formation of a timeline.

5.1. Instagram IndexedDB Artifacts Location and Storage Files for Mozilla Firefox and Google Chrome Browsers

The treatment of the experiment was performed with Browser1 and Browser2, which, in order, were Mozilla Firefox and Google Chrome. In Windows 10 Home Operating System, the LevelDB files for Google Chrome are stored in "C:\Users\<user>\AppData\Local\Google\Chrome\User Data\Default\IndexedDB". There are multiple folders in the location containing the LevelDB files of different web origins. Adding the particular folder containing Instagram IndexedDB artifacts, the default storage location can be updated as "C:\Users\<user>\AppData\Local\Google\Chrome\UserData\Default\IndexedDB\https_www.instagram.com_0.indexeddb.leveldb". For LevelDB-based IndexedDB technologies, multiple files with extensions of .ldb and .log store the artifacts of their web origins. SQLite files for Mozilla Firefox IndexedDB storage are located in the folder "C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\<Random-Profile-Id>.default\storage\default". SQLite files can be easily accessed with SQLite browsers [42]. Furthermore, the access is not blocked by any security mechanisms. However, the presentation of the data is not organized and creates difficulties for the formation of time frame analysis.

5.2. Instagram IndexedDB Artifacts

The treatment of the experiment produced identical artifacts in the IndexedDB storage of Mozilla Firefox and Google Chrome browsers. The preponderance of artifacts obtained from the experiments were the results of posts or comments of users being displayed on the home page. Consequently, they were scrutinized for their potential value to the forensic examinations. The encountered artifacts almost exclusively belonged to users and their interactions on the application. A summary of the created artifacts is seen in Table 4 with their corresponding treatment action. Treatments that did not create meaningful artifacts are not listed in Table 4. The obtained artifacts are presented in the following

subsections. The value and peculiar characteristics of the artifacts are discussed in detail in the Discussions section.

Table 4. Treatment Actions and Artifacts Created [1].

Treatment	Artifacts Encountered
Displaying shared posts	• Posting time • Statistics of the interactions for the post such as like count • Owner identification artifacts • Artifacts indicating the relationship and permissions between the posting and viewing accounts • Direct link to displayed profile picture
Displaying a shared post's comments	• Entire content of the comment • Posting time of the comment • Statistics of the interactions for the comment such as like count • Owner identification artifacts • Artifacts indicating the relationship and permissions between the commenting and viewing accounts • Direct link to displayed profile picture
Displaying account profiles by hovering the mouse over posts and comments	• As addition to only displaying posts and comments: • Full name, biography, and other detailed information from the posting account, e.g., its saved website • Status information of the posting account, e.g., account being new or private
Visiting user profile pages after displaying their posts and comments	• As addition to only displaying posts and comments: • Direct link to high-definition profile picture • Artifacts indicating whether it is possible to display account contacts • Account categories and classification
Displaying shared posts and comments from blocked accounts	• Artifacts indicating the blocking status between the posting and displaying accounts
Displaying shared posts and comments from restricted accounts	• Artifacts indicating the restriction status between the posting and displaying accounts
Sending direct messages with emoji content	• Emojis and the number of times they were used
Displaying shared stories	• Time artifacts for the story such as expiry date, posting date, last posting date • Owner identification artifacts • Artifacts indicating the relationship and permissions between the posting and viewing accounts • Artifacts carrying information about the display of the story, e.g., whether it is muted or not

[1] Treatments that did not create meaningful artifacts are not listed.

5.2.1. users.users

When the home page is accessed, users.users record is populated with artifacts from the connections that have recent posts on display. These artifacts contain profile information of the connections and their mutual followers with the suspect account. Surprisingly, the artifacts are not limited to the followed connections but include all the accounts that made a comment on the displayed posts. At initial display, generated artifacts are limited with

several fields. However, as the suspect interacts with the posting account, the number of collected artifacts increases drastically. Additionally, when a post or comment is not displayed on the home page, visiting the posting account's profile page does not populate any records in IndexedDB storage. Table 5 gives details on the artifacts contained in users.users for three cases of suspect interactions with the posting accounts. The 'x' entry is inserted for artifacts that were not present for their corresponding case.

Table 5. Artifacts Found in users.users Record with User Profile Interactions.

Attribute	Displaying Posts/Comments without Interaction	Displaying Posts/Comments and Hovering over the Posting Account with Mouse	Displaying Posts/Comments and Visiting the Posting Account's Profile Page
bio:	x	"Bio of Forensic Researcher 3"	"Bio of Forensic Researcher 3"
followedBy:	x	0	0
follows:	x	1	1
fbid:	x	x	"17841448515262719"
fullName:	x	"Forensic Researcher 3"	"Forensic Researcher 3"
id:	"48581753175"	"48581753175"	"48581753175"
isNew:	x	FALSE	TRUE
isPrivate:	x	FALSE	FALSE
mutualfollowers:	x	An empty list	An empty list
profilePictureUrl:	A (lengthy) link to profile picture is obtained	A (lengthy) link to profile picture is obtained	A (lengthy) link to profile picture is obtained
username:	"forensicresearchaccount3"	"forensicresearchaccount3"	"forensicresearchaccount3"
website:	x	"http://forensicresearcher3.com/ accessed on 22 May 2022"	"http://forensicresearcher3.com/ accessed on 22 May 2022"

'x' refers to data that are not present in IndexedDB storage.

The users.users record contains more fields. However, most of the fields did not contain any meaning information for this experiment. Therefore, they were omitted from Table 5. A full list of artifacts and their fields for users.users record is shared in Appendix A Table A1.

It can be seen in Table 5 that a digital forensic investigator can obtain information about the connections of the suspect, including their mutual followers, profile pictures, and account characteristics. More on how these fields can be useful is given in the Discussion section.

5.2.2. Relationships

The relationships record contains information about the following and blocking status between accounts. Artifacts obtained from two different cases are displayed in Table 6. For Account1 and Account3, there is no connection. Account3 is blocked and restricted by Account1, as described in the Materials and Methods section. It can be seen in Table 6 that the information of the suspect following, blocking, and restricting another account can be obtained from the relationships record. However, the blocking status of the accounts for the suspect account was obtained as null.

Table 6. Artifacts Found in relationships Record with User Profile Interactions.

Attribute	Account1 and Account3	Account1 and Account2
blockedByViewer:	"BLOCK_STATUS_UNBLOCKED"	"BLOCK_STATUS_BLOCKED"
followedByViewer:	"FOLLOW_STATUS_NOT_ FOLLOWING"	"FOLLOW_STATUS_NOT_ FOLLOWING"
followsViewer:	"FOLLOW_STATUS_NOT_ FOLLOWING"	"FOLLOW_STATUS_NOT_ FOLLOWING"
hasBlockedViewer:	null	null
restrictedByViewer:	"RESTRICT_STATUS_ UNRESTRICTED"	"RESTRICT_STATUS_ RESTRICTED"

5.2.3. comments.byId and comments.byPostId

When comments are displayed on the home page of an account, comments.byId records are created. As users.users records contain information about the owners of the comments, supplemental information specific to the comments is stored in comments.byId records. The entire text content of comments can be found in this record. Additionally, important time information indicating when the comment was posted can be obtained in epoch time with the postedAt field of the record. An instance of this record can be seen in Table 7, in the comment of Account3 on the post of Account2.

Table 7. Artifacts Found in comments.byId Record.

Attribute	Value
deleted:	FALSE
didReportAsSpam:	FALSE
id:	"17884757831263207"
isAuthorVerified:	FALSE
isRestrictedPending:	FALSE
likeCount:	0
likedByViewer:	FALSE
postedAt:	1625222296
text:	"Account3 comment for Account2-Post1"
userId:	"48581753175"

comments.byPostId record targets not a specific comment but a summary of all the comments made for a post. It displays the count and display information, e.g., how many of the comments are visible on the post. In addition to an overall look into the comments, an array list of all the account IDs is also stored in this record. Table 8 shows an instance of comments.byPostId record obtained for Post1 by Account2. As can be seen from the commentsIds attribute value, the comment that is seen in detail in Table 7 is the only comment on this post.

Table 8. Artifacts Found in comments.byPostId Record.

Attribute	Value
commentIds:	0 -> "17884757831263207"
length:	1
count:	1
hasNextPage:	FALSE

Table 8. *Cont.*

Attribute	Value
hasPreviousPage:	undefined
isFetching:	FALSE
loadedCount:	1
visibleCount:	1

5.2.4. posts.byId

Similar to comments, information about every post displayed on the home page is also recorded in IndexedDB storage of Instagram. posts.byId record contains detailed information for posts including their location and posting time. More noteworthy information for connecting suspects and the evidence is in the viewerInPhotoOfYou and owner->fullName fields. The owner of the post and whether the suspect is involved in the post can be obtained from these fields. Table 9 shows the posts.byId record for Post1 of Account2. Since this record contains over fifty fields, only the fields deemed significant are listed in Table 9. A full list of artifacts and their fields for posts.byId record is shared in Appendix B.

Table 9. Artifacts Found in posts.byId Record.

Attribute	Value
accessibilityCaption:	"Photo by Forensics Researcher 2 in Missouri City, Texas."
caption:	"Forensic Researcher 2-Post 1"
commentsDisabled:	FALSE
followHashtagInfo:	null
hasAudio:	TRUE
isVideo:	FALSE
likedByViewer:	FALSE
likers:	An empty list
location->id:	"228672033"
lat:	undefined
lng:	undefined
location->name:	"Missouri City, Texas"
slug:	"missouri-city-texas"
numComments:	1
numPreviewLikes:	1
owner->fullName:	"Forensics Researcher 2"
owner->id:	"16009265888"
isNew:	FALSE
isPrivate:	FALSE
username:	"forensicresearchaccount2"
postedAt:	1625222164
previewCommentIds:	0 -> "17884757831263207"
savedByViewer:	FALSE
usertags:	An empty list
viewerCanReshare:	TRUE
viewerInPhotoOfYou:	FALSE

5.2.5. users.usernameToId and users.viewerId

The stored data, which are associated with posts and comments of Instagram users, are linked to their owner accounts through user IDs. users.usernameToId record provides the connection between the usernames and the user IDs. Table 10 displays information from users.usernameToId record for the usernames and user IDs of the accounts utilized in the experiments.

Table 10. Artifacts Found in users.usernameToId Record.

Attribute	Value
awesome.photographers	"1077125"
forensicreasearchaccount1	"46912168943"
forensicresearchaccount2	"16009265888"
forensicresearchaccount3	"48581753175"

users.viewerId record is a single-field record that indicates the ID of the user for whose account the IndexedDB storage is populated. This can be matched to the username of the suspect in the users.usersnameToId record. For Account1, users.vieverId was recorded as "46912168943".

5.2.6. direct.emojis

Emojis are a relatively new and primitive language for describing emotions quickly. [43]. direct.emojis record stores the emojis used by the user whose account is under investigation. The actions that populate the direct.emojis record include direct messages, comments, and posts. The record stores the emoji and the number of times it is used. However, the direct message or account for which the emoji is used is not specified. This record is not populated when an emoji is used, sent, or displayed from another account's entries or direct messages. It is also noteworthy to remark that when a reaction is given to a story or a direct message with emojis, no data are inserted in direct.emojis record. Table 11 gives the contents of this record from the experiments.

Table 11. Artifacts Found in direct.emojis Record.

Emoji	Number of Times It Is Used
☹	1
⊖	1
㊙	1

5.2.7. stories.feedTray and stories.reels

The story items displayed on the home page yield artifacts in stories.feedTray and stories.reels records. The stories.feedTray lists the user IDs of all the accounts for which a story is displayed on the home page. stories.reels contains details of the stories. Considerable fields in the stories.reels record emphasize times of interactions. The "seen" attribute gives the epoch time of when the user displayed the story for the first time. latestReelMedia takes it one step further and gives the time of post for the latest story of the account. expiringAt attribute represents the time when the story will be out of the display. Furthermore, the position and order in which the story was seen are recorded in seenRankedPosition attribute. stories.reels record also contains attributes similar to attributes of post records, including location information and the abilities of the account viewing the story. Table 12 displays the stories.feedTray record for Account1, where there is only one story from Account2. Table 13 displays stories.reels record for Story1 posted by Account2.

Table 12. Artifacts Found in stories.feedTray Record.

Attribute	Value
id:	"16009265888"
length:	1

Table 13. Artifacts Found in stories.reels Record.

Attribute	Value
canReply:	TRUE
canReshare:	TRUE
expiringAt:	1625683666
id:	"16009265888"
isCloseFriends:	FALSE
latestReelMedia:	1625597266
locationId:	undefined
muted:	FALSE
rankedPosition:	1
seen:	1625597266
seenRankedPosition:	1
tagName:	undefined
title:	undefined
userId:	"16009265888"

6. Discussion

The artifacts encountered in the IndexedDB storage of Instagram Web are primarily created from the interactions on the home page. Posts, comments, and stories displayed on the home page populate the IndexedDB storage with data such as owner identification, account relationships, permissions, and direct links to post resources, e.g., direct links of profile pictures. The entire contents of comments and post descriptions, post locations, and user tags were able to be extracted from the storage. The number of artifacts belonging to an account also increases when more interactions with their profile page are provided. The additional artifacts that can be obtained with supplemental profile interactions exhibit full name, biography, saved website, status, and category of the accounts. It is also possible to obtain statistical information from the account profiles, such as the number of followers and the number of accounts followed. Furthermore, significant information pertinent to the accounts, e.g., whether they are new, private, professional, unpublished, and verified accounts, can be obtained through the profile page interactions in the case that their posts or comments are displayed on the home page.

With scrutinization of IndexedDB storage for Instagram application, it is possible to create connections between the account owners. A dedicated record called relationships contains valuable information that can be utilized, along with the information from the users.users record, for the construction of maps that can indicate the place of suspects in their social networks. Furthermore, the level of the relationships can be estimated based on attributes such as viewerInPhotoOfYou attribute from posts.byId records and isCloseFriends attribute from stories.reels records. These connections and their strength are valuable to forensic investigations, as social media applications are often utilized for criminal activity of drug transactions, organized crime coordination, and cyberbullying [44]. It creates opportunities to detect the accomplices of a crime and to collect information on the posts, although comments of the accounts that are private to the public would require additional warrants.

Artifacts created by the usage of user stories contained an additional value for the establishment of user connections. Mainly, they contain an attribute called isCloseFriends. Additionally, it is possible to obtain information on when a story was displayed and in what order it was seen. Utilizing the artifacts extracted from users.users, users.usernameToId, stories.reels, and relationships records, we were able to create connection analysis between the accounts employed in the experiments. The techniques for establishing connections between Instagram accounts are discussed in the proof-of-concept tool section.

By examining the IndexedDB artifacts of Instagram, examiners can also detect a suspect account's focus and interests on social networks. If the contents of a post or comment on the home page of the account contain an indication of a criminal activity or tendency to criminal behavior, likedByViewver, savedByViewer, and savedByViewerToCollection attributes from the posts.byId record can be utilized to detect any interest in these posts or comments.

Another potential subject of interest for behavioral analysis is the direct.emojis records. Researchers have been utilizing emoji analysis to detect the personality and mental states of users since it gained popularity [45]. It is particularly important for criminal behavior analysis [44]. The direct.emojis records provide the emojis used by the suspect account and the frequency of usage for each emoji. This frequency can be practical, e.g., in cases where a sad face is used extensively, or in cases where a religious symbol is used with a similar frequency to emojis that indicate negative feelings [46].

Artifacts obtained from postedAt attribute of posts.byId, expiringAt and "seen" attributes of stories.reels, and postedAt attribute of comments.byId display time data in UNIX epoch time format. This is a frequently encountered format, since it eliminates the need for time conversions from different time zones. Epoch time is calculated by calculating the number of seconds since 1 January 1970, midnight, in UTC/GMT [47]. The epoch time artifacts are useful in formation of time-frame analysis, as they can be sorted based on their numerical value. Additionally, the post captions contain information on the share time of the posts. These attributes can be utilized in the time frame analysis of forensic investigations.

The extraction of artifacts providing the account, time, and content details of posts, stories, and comments proves the hypothesis that actions of the users on Instagram can be detected. These actions can also be structured based on their epoch time information to form time-frame analysis.

7. Proof-of-Concept Tool

The SQLite file used for IndexedDB storage in Mozilla Firefox resides in "C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\<Random-Profile-Id>default\storage\-default\" with the naming convention of "<Random-Numbers>rxeud.sqlite", where rxeud is the name of the database. Therefore, the first step before processing was to extract the <Random-Numbers>rxeud.sqlite file to the operation folder. IndexedDB saves blob data together with text data in the browser storage. When the data were received in their raw format, they needed to be processed to a manageable format to perform seven tasks:

- Identification of record names
- Extraction and identification of account IDs
- Detection of the suspect account
- Establishment of relationships with other accounts
- Extraction of time and location information
- Emoji breakdown utilizing data in hex format
- Extraction and processing to present data from records in a keyword-searchable formation

Each of these tasks is presented in the following sections, with the addition of integration, verification, and restrictions sections.

7.1. Identification of Record Names

The returning strings from $query = "Select key From object_data" returned different names of records. In order to ensure the right names could be obtained even after Instagram changes to these names in the future, a logical transformation was performed. Therefore, table names were first obtained in the raw format and translated to user display format with the following principles:

- The records start with the character '0'and are divided by a '/' character (separator).
- Except for the relationships record, all records have a separator.
- There are three sets of user records.
- users.users record has the keyword "users" before and after its separator.
- Identifying the rest of the user records by length yields the keyword "byId" after the separator.
- comments.byId can be identified with "byId" keyword, which yields "comments" as well.
- Nonrepeating record name after the previous steps is direct.emojis.

The translation of the table names resulted in the creation of a translation table. This table is printed in the tool with the following format:

1	relationships	0sfmbujpotijqt
2	users.users	0vtfst/vtfst
3	users.usernameToId	0vtfst/vtfsobnfUpJe
4	users.viewerId	0vtfst/wjfxfsJe
5	comments.byId	0dpnnfout/czJe
6	posts.byId	0qptut/czJe
7	comments.byPostId	0dpnnfout/czQptuJe
8	feed.visibleCount	0gffe/wjtjcmfDpvou
9	feed.items	0gffe/jufnt
10	stories.feedTray	0tupsjft/gffeUsbz
11	stories.reels	0tupsjft/sffmt
12	direct.emojis	0ejsfdu/fnpkjt

7.2. Extraction and Identification of Account IDs

Account IDs were initially obtained from users.usernameToId record. A function called getAccountTranslation was utilized on the raw data to subtract numbers over the length of 5 characters. This strategy works on users.usernameToId record, because the only data contained here are the account names and their corresponding IDs. The IDs were then matched to the account names that were listed below their ID strings.

7.3. Detection of the Suspect Account

The suspect account is the account for which the .sqlite file is obtained. In Instagram IndexedDB storage, a record called users.viewerId contains the account ID of the account for which the data are stored. This ID was matched to one of the account IDs extracted

from users.usernameToId. The strings obtained from the tables occasionally missed the beginning or the ending characters. Therefore, the matching operation was performed by checking the most similar ID in the users.usernameToId record. The shorter ID was subsequently updated to its complete format.

7.4. Extraction and Identification of Account IDs

The basic relationship of the suspect account with other accounts was obtained from the relationships record. However, this table contained the most blob data, and the field names were only partially obtained. The key field names integrated into the blob data were separated occasionally with the number 8 (eight) at the beginning and the number 6 (six) at the end. Therefore, the raw data were cleared by extraction of the blob characters. Subsequently, they were presented in separate lines to form meaningful representations. In addition, occasional values were extracted from their partial strings where the data were clearly represented.

The tool first detects the account IDs in the relationship records and then matches them to the account IDs that were previously discovered. It separates them into line breaks, obtains the text in between account IDs, and constructs the relationship from obtained keywords. The relationship information is then presented to the user with the following prints.

The account IDs discovered in the relationships record:

```
-------------------------------------------------------------
1 | 16009265888 | forensicresearchaccount2
-------------------------------------------------------------
2 | 46912168943 | suspectaccount
-------------------------------------------------------------
```

Positions of account IDs in the record:

```
---------------------------------------
1 | 16009265888 | 2
---------------------------------------
2 | 46912168943 | 18
---------------------------------------
```

The relationship between suspect and 16009265888 (forensicresearchaccount2):

```
-----------------------------------------------
1 | BlockedBySuspect     | 0
-----------------------------------------------
2 | FollowedBySuspect    | 1
-----------------------------------------------
3 | RestrictedBySuspect  | 0
-----------------------------------------------
```

7.5. Extraction of Time and Location Information

There are a couple of sources that contain time and location information, which is discussed in the discussion section. The tool obtains the time data given in these sources by keyword detection and extraction of the data from the next or the previous word of the string data. Additionally, to obtain the time and location information, the strings in particular structures were extracted. The getTimeTranslation and getLocationTranslation functions utilized the following regular expressions for obtaining these structures.

```
preg_match_all('/(on)[a-zA-Z]{4,9}[0–9]{6}/', $body[$i], $matches
preg_match_all('/(in)[a-zA-Z]{6,20}(with)/', $body[$i], $matches
```

These expressions return the data from the captions and prints, displaying the following prints.

The account IDs discovered in the posts record:

```
-------------------------------------------------------------
1 | 16009265888 | forensicresearchaccount2
-------------------------------------------------------------

Times detected in the record:
---------------------------------------
1 | onJuly212021 | 75
---------------------------------------

Locations detected in the record:
-----------------------------------------------
1 | inMissouriCityTexaswith | 3
-----------------------------------------------
```

7.6. Emoji Breakdown Utilizing Data in Hex Format

The hex values of the data are utilized to detect emojis and their frequency of use. A set of experiments were run on the data obtained from the direct.emojis record to obtain the positions of data, which would change with the emoji and the frequency of use. Figure 3 displays the hex value positions that changed with the usage of different types of emojis and their usage frequency. The marked values are changed in accordance with the position of the emoji in the selection screen and the frequency of use. It was observed that when multiple emojis are utilized, a different instance of the 5th line from Figure 3 was inserted after the first one, and the rest of the values were not changed. The tool represents this information with the following print.

```
Found emojis and number of times they were used:
-----------------------------------------------------------------
1 | 1st emoji in Instagram selection list | 2
-----------------------------------------------------------------
```

```
Starting emoji construction for the file: 3932499597rxeud.sqlite
1 30 04 03 00 01 01 04 F1 FF 01 06 3C 08
2 00 FF FF
3 02 00 00 00 04
4 00 FF FF
5 3D D8 00 DE 01 14 01 10 2C 03
6 00 FF FF
7 00 00 00 00 13
8 00 FF FF
```

Figure 3. Emoji Usage Effect on the Hex Data of the direct.emojis Record.

7.7. Extraction and Processing to Present Data from Records in a Keyword-Searchable Formation

Heavy data processing brings with it a potent complication. In simple terms, processed, filtered, and removed data can be useful in future instances of storage. Therefore, a feature was introduced to the tool to display the artifacts such that data can be searched in a raw, filtered, and processed form directly from the .sqlite file. Figure 4 shows this feature of a graphical user interface in the raw and numbered tab. The raw and numbered tab does not translate or remove any characters from the original blob data. The same data after processing are displayed in the filtered and numbered tab. At the beginning of the tabs, there is a list of the line positions where account IDs are detected. Therefore, the investigator can choose to locate the account information manually without performing any search.

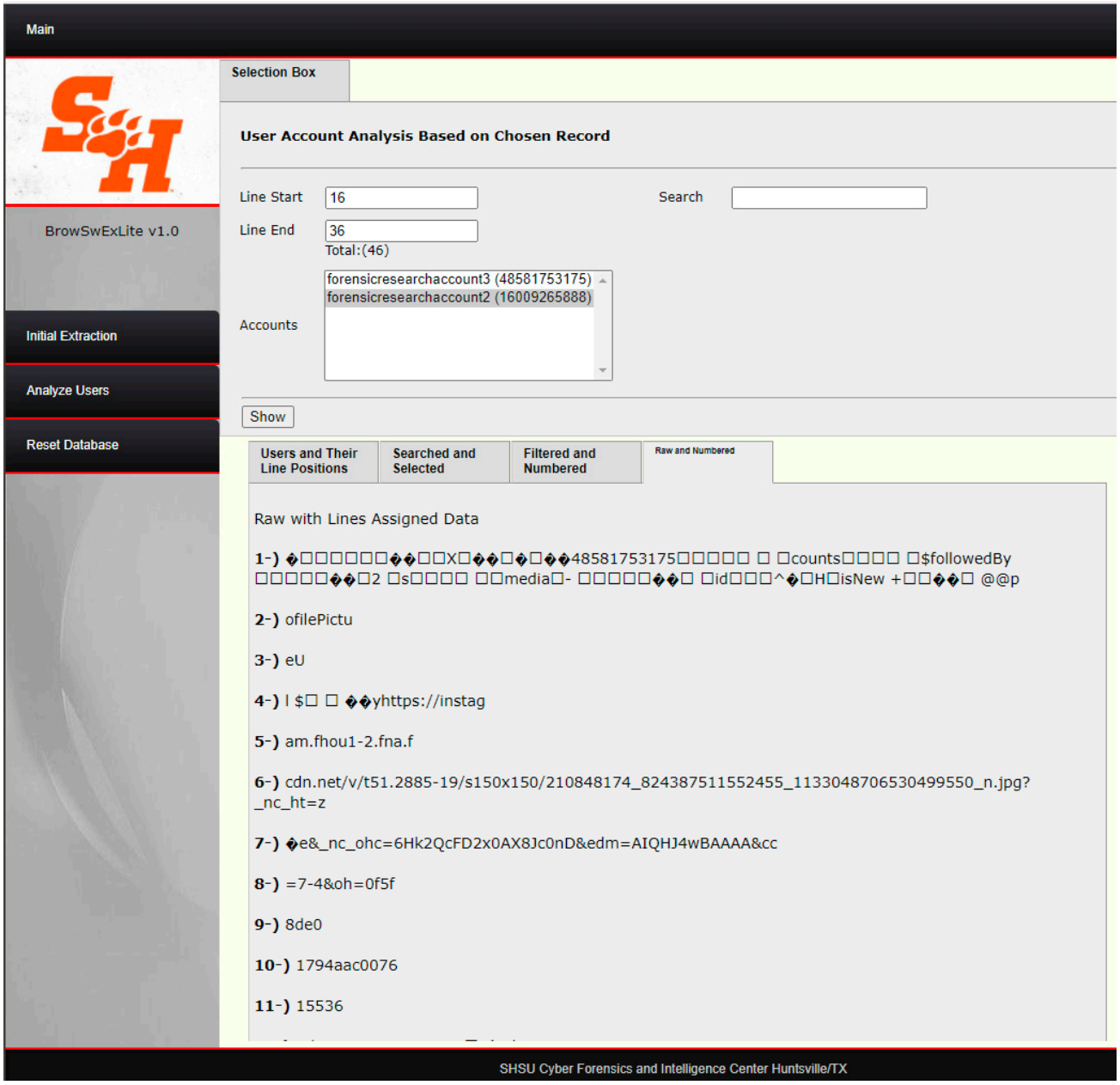


Figure 4. Displaying Records in Processed and Raw Format.

Initially, the account IDs detected in the records, which match the account IDs recorded during the extraction phase, were located in the corresponding positions of the record. This position matching is performed with the line numbers where the account ID is present. When the investigator clicks on one of the listed accounts, an estimated position of starting and ending line numbers is given to lead the investigator to the position where related data can be observed. Additionally, when a search term is given, it is extracted within the obtained position interval and displayed accordingly in a separate tab. Searching with the keyword fullName within the given line positions 16 (sixteen) to 36 (thirty-six) is demonstrated in Figure 5. The Line Start and Line End fields are filled automatically when the investigator clicks on the account with ID number 16009265888.

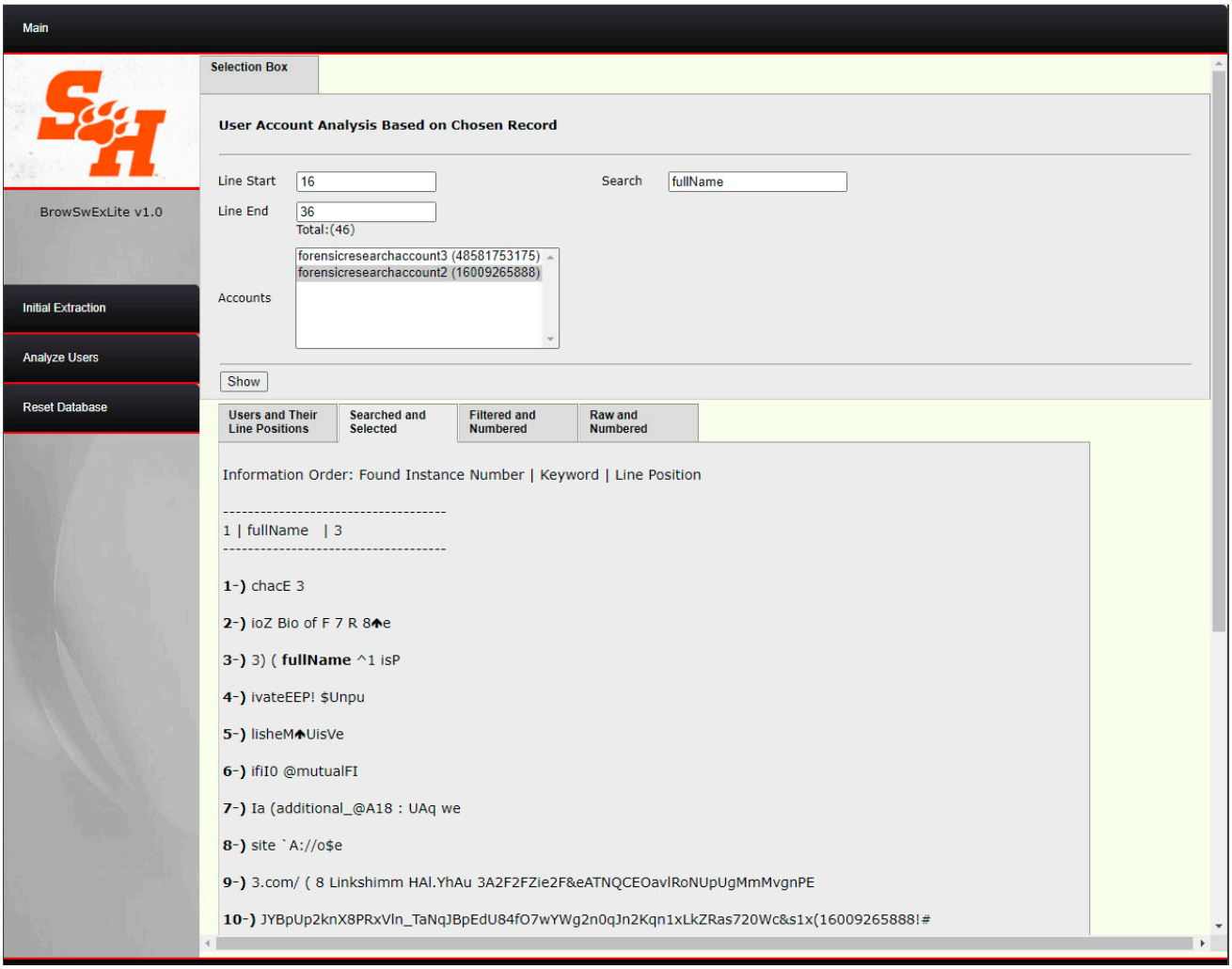


Figure 5. Searching Records in Processed and Raw Format.

The position interval is determined with plus ten and minus ten of the exact line location where the user ID is detected. If the ID is detected in the first nine lines, then the Line Start field is taken as 0 (zero). Similarly, if the ID is detected within the last nine lines, then the Line End is determined as the last line of the record. The searched and selected tab is displayed automatically after the search is performed. If the searched word is found, the word and the ID of the user are displayed in bold characters. If the search word is not found, it is stated at the beginning of the record. This can be seen in Figure 6 for the search word "nothing".

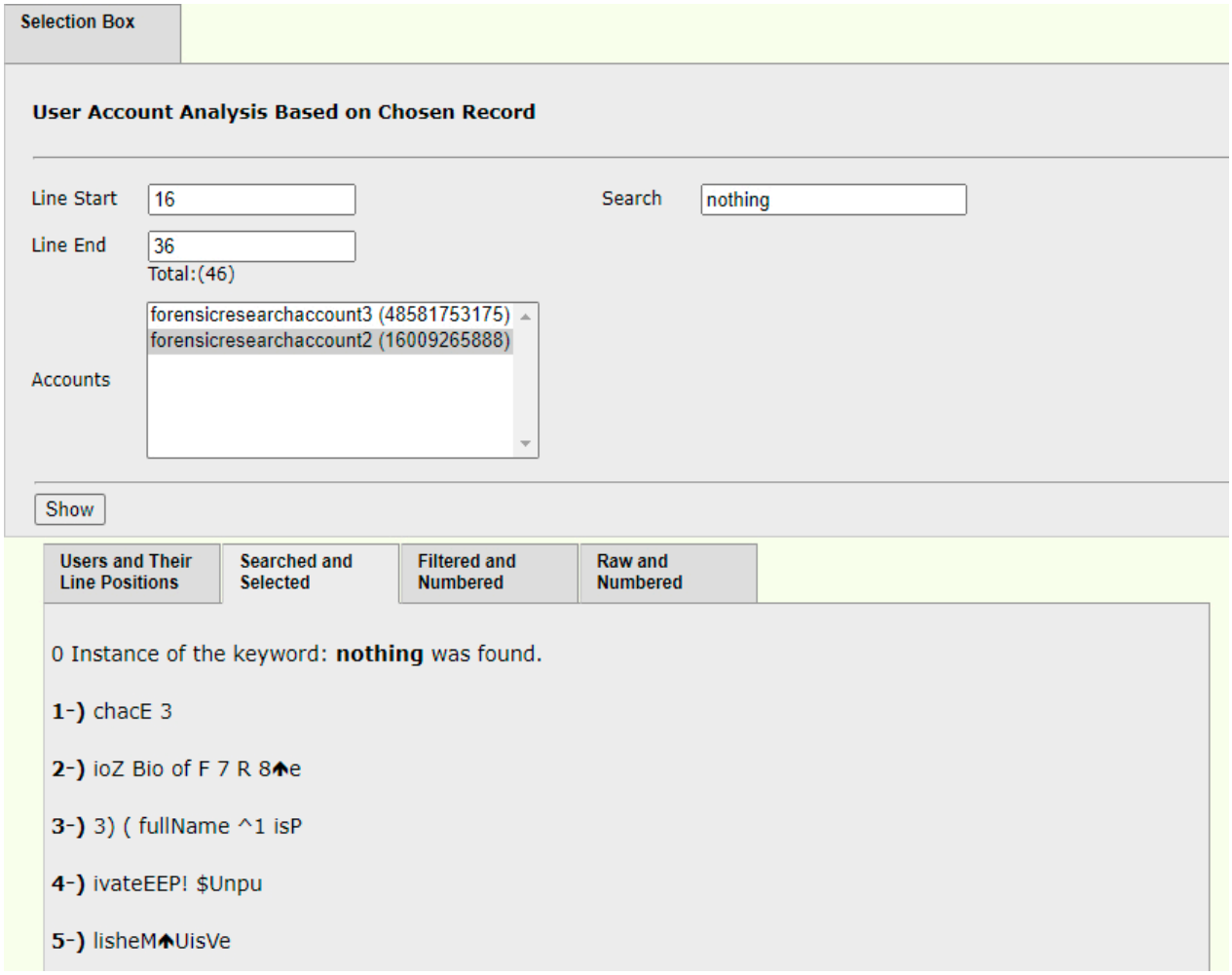


Figure 6. Searching Records for Non-existing Words.

7.8. Integration

Five main modules work on Instagram IndexedDB storage for extracting, parsing to an operatable format, storing, exploring, and presenting to the investigator. Figure 7 presents these modules with their sub-functionalities and interactions. Initially, the extraction module queries .sqlite file with PHP SQLite3 Class [48] in raw blob data, filtered text, and hex formats. The returned output is employed in the string parsing module to structure and index the data into lines with HTML arrangement to enable search and coherent presentation.

A limited amount of data are also stored in a MySQL database to enable cross comparisons of account IDs and record names. if it doesn't already exist, the database is automatically created at the time of operation. Code 3 displays the code used to initialize its database. Since data storage is utilized for functional objectives and not for the storage of data over time, it is not designed to support multiple cases. The database needs to be reset before starting another analysis. This can be done from the Reset Database selection of the side menu seen in Figures 4 and 5. The data saved in the database are not all the data that are presented to the investigator but the data that need to be used for further processing. Therefore, the search and printing modules utilize information both from the database conversion module and the string parsing module.

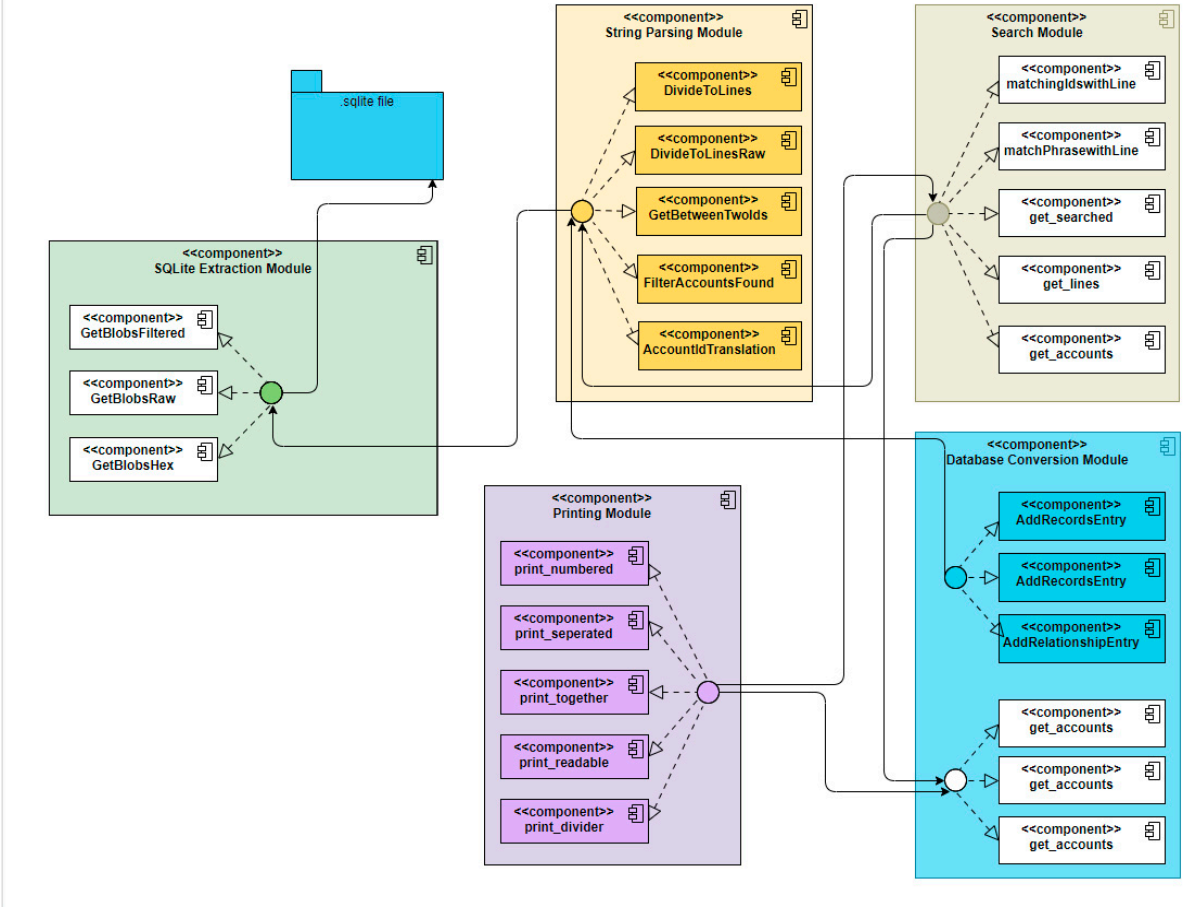


Figure 7. Tool Modules.

Code 3. Tool Database Initiation

```
CREATE TABLE IF NOT EXISTS Records(
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
                                    name VARCHAR(30),
                                    obtainedname VARCHAR(30) )
CREATE TABLE IF NOT EXISTS Accounts(
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
                                    name VARCHAR(30),
                                    instagramid VARCHAR(30),
                                    issuspect INT(1) )
CREATE TABLE IF NOT EXISTS Relationships(
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY
                                    instagramid VARCHAR(30),
                                    isblocked INT(1),
                                    isfollowed INT(1),
                                    isrestricted INT(1) )
CREATE TABLE IF NOT EXISTS Times(
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY
                                    action VARCHAR(50),
                                    time VARCHAR(30),
                                    account VARCHAR(40),
                                    details VARCHAR(250) )
```

7.9. Verification

The majority of the artifacts from the single-case pretest–posttest quasi-experiment were detected by the proof-of-concept tool. However, a limited number of artifacts were affected by the character-encoding problems due to the insertion of entire logical data in blob format. This restriction is discussed further in the restrictions section. For the retrieved artifacts, verification was performed through Mozilla Firefox Storage Inspector [49] and Google Chrome Developer Tools [50]. These browser-integrated tools are only able to be operated when the website is actively open on the suspect's computer. Therefore, their capabilities in forensic investigations are limited. The proof-of-concept tool distinguishes itself with its ability to work directly on the storage files without user credentials present for authentication. The verification is performed through a cross check of artifacts by the proof-of-concept tool, Google Chrome Developer Tools, and Mozilla Firefox Storage Inspector.

7.10. Time Frame Formation

For time frame analysis, the list of actions obtained from the records was saved to a table in the database with their epoch time and account information. The entries of the table were subsequently sorted according to the epoch time field and printed for time frame analysis. The time frame of the actions in the experiment can be seen in Figure 8.

TimeFrame Analysis

This section gives ***Time Frame*** of Actions

Action	Time	Account	Details
Post Made	1625222164	forensicresearchaccount2	inMissouriCityTexaswith
Comment Made	1625222296	forensicresearchaccount3	for forensicresearchaccount2
Story seen	1625597266	forensicresearchaccount2	story of forensicresearchaccount2

Figure 8. Time Frame Display in the Tool.

7.11. Restrictions

The location of the storage file is determined based on a random number pattern that is different for every user. Therefore, the investigator is required to provide the .sqlite file to the working directory manually during live acquisitions. The tool will go over every .sqlite file placed in its directory.

The investigator needs to have a PHP and MySQL server to get the tool working. The database is created and populated automatically when the database username and password are entered to MakeDBConnect.php under the factory directory. We utilized a WAMP (Windows, Apache, MySQL, and PHP) server [51] that contains PHP and MySQL together for the development of the tool. The blob data saved in object_data of Instagram IndexedDB storage contain characters that go over the limits of Unicode printable characters. Therefore, the characters of text data can be subsequently affected by encoding problems. During the experiments, similar issues were encountered with the initial and the last digits of account IDs being missing and significant indicator keywords being slightly different than expected. A major issue was encountered with emojis, where the entire data appeared in binary format. Therefore, it was necessary to perform a hex value experiment on the raw data. This approach appears more effective but requires substantially long experiments to arrive at the correct format. The tool is intended to operate fully on hex raw data in the future. However, in the current version, it is possible to experience glitches with characters.

The code for the Proof-of-Concept tool can be found in the Supplementary Material of the paper.

8. Conclusions and Future Work

In this study, the artifacts of social network site Instagram, which are stored in IndexedDB storage of web browsers, are put under scrutinization, with a focus on retrieval from SQLite storage of Mozilla Firefox browser. Two hypotheses were created for testing: IndexedDB artifacts can be utilized to determine the actions of a suspect on Instagram, and it is possible to present these actions with a time frame analysis. A single case pretest–posttest quasi-experiment was applied to populate the IndexedDB storage file. A proof-of-concept tool was introduced to methodize the extraction, processing, storage, and presentation of the discovered artifacts.

The experiment created a large number and variety of artifacts in the targeted storage file that indicate substantial value for the determination of user actions such as sharing posts, making comments, and displaying stories. Additionally, a time frame table was formed by the proof-of-concept tool utilizing epoch time artifacts as displayed in Figure 8. Therefore, the hypotheses put forth for the research are validated.

Furthermore, the artifacts exhibited suitability to constructing not only timeframe analysis but additional location and social connection analysis for the Instagram users. As many people share their life with their connections on social networking platforms, the effect of particular posts and content on the mental states of the users can be put under scrutinization [52]. Further research can be conducted on social circles of Instagram users by utilizing relationship records and the contents of the posts and comments shared by their connections.

Supplementary Materials: The following is available online at https://github.com/furkanpaligu/InstagramForensics.git (accessed on 22 May 2022), Proof-of-Concept tool software.

Author Contributions: Conceptualization, F.P. and C.V.; methodology, F.P. and C.V.; software, F.P.; validation, F.P. and C.V.; formal analysis, F.P. and C.V.; resources, C.V.; data curation, F.P.; writing—original draft preparation, F.P.; writing—review and editing, F.P. and C.V.; visualization, F.P.; supervision, C.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable, the study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Full List of Artifacts and Their Fields Found in users.users Record.

Attribute	Displaying Posts/Comments without Interaction	Displaying Posts/Comments and Hovering over the Posting Account with Mouse	Displaying Posts/Comments and Visiting the Posting Account's Profile Page
bio:	x	"Bio of Forensic Researcher 3"	"Bio of Forensic Researcher 3"
businessAddress city_id:	x	x	x
businessAddress city_name:	x	x	x
businessAddress latitude:	x	x	x
businessAddress longitude:	x	x	x
businessAddress street_address:	x	x	x

Table A1. *Cont.*

Attribute	Displaying Posts/Comments without Interaction	Displaying Posts/Comments and Hovering over the Posting Account with Mouse	Displaying Posts/Comments and Visiting the Posting Account's Profile Page
businessAddress zip_code:	x	x	x
businessAddress businessEmail:	x	x	x
businessAddress categoryEnum:	x	x	null
businessAddress categoryName:	x	x	null
counts followedBy:	x	"http://forensicresearcher3.com/ accessed on 22 May 2022"	"http://forensicresearcher3.com/ accessed on 22 May 2022"
counts follows:	x	0	0
counts media:	x	1	1
fbid:	x	0	0
fullName:	x	x	"17841448515262719"
hasAREffects:	x	"Forensic Researcher 3"	"Forensic Researcher 3"
hasClips:	x	x	FALSE
hasGuides:	x	x	FALSE
hideLikeAndView Counts:	x	x	FALSE
highlightReelCount:	x	x	0
id:	"48581753175"	"48581753175"	"48581753175"
isNew:	x	FALSE	TRUE
isPrivate:	x	FALSE	FALSE
isProfessionalAccount:	x	x	FALSE
isUnpublished:	x	undefined	undefined
isVerified:	x	FALSE	FALSE
mutualFollowers additional_count:	x	0	0
mutualFollowers usernames:.	x	Array(0)	Array(0)
overallCategoryName:	x	x	""
profilePictureUrl:	"https://instagram.fhou1-2 .fna.fbcdn.net/v/t51.2885-19/ s150x150/210848174_824387511 552455_1133048706530499550 _n.jpg?tp=1&_nc_ht=instagram. fhou1-2.fna.fbcdn.net&_nc_ ohc=rtciv5BoApEAX_MBAS5 &edm=AIQHJ4wBAAAA& ccb=7-4&oh=9f6e0779d54488c6 a8fe69bff01fbe8e&oe=60E63D2 B&_nc_sid=7b02f1 accessed on 22 May 2022"	"https://instagram.fhou1-2.fna. fbcdn.net/v/t51.2885-19/s150x1 50/210848174_824387511552455 _1133048706530499550_n.jpg?tp= 1&_nc_ht=instagram.fhou1-2.fna. fbcdn.net&_nc_ohc=rtciv5 BoApEAX_MBAS5&edm=AEF8 tYYBAAAA&ccb=7-4&oh=ab270 768ccd5a11f9d4fd872cf21bd37& oe=60E63D2B&_nc_sid=a9513d accessed on 22 May 2022"	"https://instagram.fhou1-2.fna. fbcdn.net/v/t51.2885-19/s150x1 50/210848174_824387511552455 _1133048706530499550_n.jpg?tp= 1&_nc_ht=instagram.fhou1-2.fna. fbcdn.net&_nc_ohc=rtciv5 BoApEAX_MBAS5&edm=ABfd0 MgBAAAA&ccb=7-4&oh=060b0 f4fd8772e0d68be39523e109dfa& oe=60E63D2B&_nc_sid=7bff83 accessed on 22 May 2022"

Table A1. *Cont.*

Attribute	Displaying Posts/Comments without Interaction	Displaying Posts/Comments and Hovering over the Posting Account with Mouse	Displaying Posts/Comments and Visiting the Posting Account's Profile Page
profilePictureUrlHd:	x	x	"https://instagram.fhou1-2.fna.fbcdn.net/v/t51.2885-19/s320x320/210848174_824387511552455_1133048706530499550_n.jpg?tp=1&_nc_ht=instagram.fhou1-2.fna.fbcdn.net&_nc_ohc=rtciv5BoApEAX_MBAS5&edm=ABfd0MgBAAAA&ccb=7-4&oh=d753326d583d2063f52be79dfcbf679b&oe=60E58DD3&_nc_sid=7bff83 accessed on 22 May 2022"
shouldShowPublic Contacts:	x	x	FALSE
username:	x	x	FALSE
website:	"forensicresearchaccount3"	"forensicresearchaccount3"	"forensicresearchaccount3"
websiteLinkshimmed:	x	"https://l.instagram.com/?u=http%3A%2F%2Fforensicresearcher3.com%2F&e=ATNQH9UUbaaBoX-fm5PkykJPbj399AqQeWaysa1WXMANeSr9NYwpP3LDSrAAfGe7yVqLpEP_BneR-ZNgMw57e0o&s=1 accessed on 22 May 2022"	"https://l.instagram.com/?u=http%3A%2F%2Fforensicresearcher3.com%2F&e=ATNQH9UUbaaBoX-fm5PkykJPbj399AqQeWaysa1WXMANeSr9NYwpP3LDSrAAfGe7yVqLpEP_BneR-ZNgMw57e0o&s=1 accessed on 22 May 2022"

'x' refers to data that are not present in IndexedDB storage.

Appendix B

Table A2. Full List of Artifacts and Their Fields Found in posts.byId Record.

Attribute	Value
accessibilityCaption:	"Photo by Forensics Researcher 2 in Missouri City, Texas."
attribution:	null
caption:	"Forensic Researcher 2 - Post 1"
coauthorProducers:	Array(0)
length:	0
code:	"CQ0lgsltI2-"
commentsDisabled:	FALSE
dimensions height:	1080
dimensions width:	1080
dimensions displayResources:	Array(3)
dimensions 0->configHeight:	640
dimensions 0->configWidth:	640

Table A2. *Cont.*

Attribute	Value
dimensions 0->src:	"https//instagram.fhou1-2.fna.fbcdn.net/v/t51.2885-15/sh0.08/e35/s640x640/209842360_101 9472398787878_2699888029284717661_n.jpg?tp=1&_nc_ht=instagram.fhou1-2.fna.fbcdn.net& _nc_cat=102&_nc_ohc=MKgGzMJfhMwAX-qbD9Y&edm=AIQHJ4wBAAAA&ccb=7-4&oh= ce096f38e23cad2f932c3db0a8e09c7c&oe=60E50FAB&_nc_sid=7b02f1 accessed on 22 May 2022"
dimensions 1->configHeight:	750
dimensions 1->configWidth:	750
dimensions 1->src:	"https//instagram.fhou1-2.fna.fbcdn.net/v/t51.2885-15/sh0.08/e35/s750x750/209842360_101 9472398787878_2699888029284717661_n.jpg?tp=1&_nc_ht=instagram.fhou1-2.fna.fbcdn.net& _nc_cat=102&_nc_ohc=MKgGzMJfhMwAX-qbD9Y&edm=AIQHJ4wBAAAA&ccb=7-4&oh=08 58f548445cccaed0f08f8fcd05008f&oe=60E62154&_nc_sid=7b02f1 accessed on 22 May 2022"
dimensions 2->configHeight:	1080
dimensions 2->configWidth:	1080
dimensions 2->src:	"https//instagram.fhou1-2.fna.fbcdn.net/v/t51.2885-15/e35/s1080x1080/209842360_10194723 98787878_2699888029284717661_n.jpg?tp=1&_nc_ht=instagram.fhou1-2.fna.fbcdn.net&_nc_ cat=102&_nc_ohc=MKgGzMJfhMwAX-qbD9Y&edm=AIQHJ4wBAAAA&ccb=7-4&oh=dcca5 3384890d8a47044594ec752fcd4&oe=60E4F184&_nc_sid=7b02f1 accessed on 22 May 2022"
length:	3
followHashtagInfo:	null
hasAudio:	TRUE
hasRankedComments:	FALSE
id:	"2608875058775756222"
isAffiliate:	FALSE
isPaidPartnership:	FALSE
isSidecar:	FALSE
isVideo:	FALSE
likedByViewer:	FALSE
likers:	Array(0)
length:	0
Location hasPublicPage:	TRUE
Location hasPublicStory:	undefined
Location id:	"228672033"
Location lat:	undefined
Location lng:	undefined
Location name:	"Missouri City, Texas"
Location profilePictureUrl:	undefined
Locationslug:	"missouri-city-texas"

Table A2. *Cont.*

Attribute	Value
mediaOverlayInfo:	null
mediaPreview:	"ACoqmooxS1maCUlOpuaACjFGKKAHAU6kFLQAlJinUEUANxSc07FJQAoooFKaACg0Gmk9KAFpKWkpgf/Z"
numComments:	1
numPreviewLikes:	1
overlayImageSrc:	null
owner counts:	No properties
owner fullName:	"Forensics Researcher 2"
owner id:	"16009265888"
owner isNew:	FALSE
owner isPrivate:	FALSE
owner profilePictureUrl:	"https//instagram.fhou1-2.fna.fbcdn.net/v/t51.2885-19/s150x150/209605462_607957357281103_474552616487604682_n.jpg?tp=1&_nc_ht=instagram.fhou1-2.fna.fbcdn.net&_nc_ohc=wEUFvjMJy0YAX9y62rP&edm=AIQHJ4wBAAAA&ccb=7-4&oh=b3d065caac08d16698b4b304e1da0854&oe=60E5948A&_nc_sid=7b02f1 accessed on 22 May 2022"
owner username:	"forensicresearchaccount2"
postedAt:	1625222164
previewCommentIds:	Array(1)
0:	"17884757831263207"
length:	1
relatedMedia:	Array(0)
length:	0
savedByViewer:	FALSE
savedByViewerToCollection:	FALSE
bloksAppUrl:	null
shouldHaveSharingFriction:	FALSE
sponsors:	Array(0)
length:	0
src:	"https//instagram.fhou1-2.fna.fbcdn.net/v/t51.2885-15/e35/s1080x1080/209842360_1019472398787878_2699888029284717661_n.jpg?tp=1&_nc_ht=instagram.fhou1-2.fna.fbcdn.net&_nc_cat=102&_nc_ohc=MKgGzMJfhMwAX-qbD9Y&edm=AIQHJ4wBAAAA&ccb=7-4&oh=dcca53384890d8a47044594ec752fcd4&oe=60E4F184&_nc_sid=7b02f1 accessed on 22 May 2022"
trackingToken:	"eyJ2ZXJzaW9uIjo1LCJwYXlsb2FkIjp7ImlzX2FuYWx5dGljc190cmFja2VkIjpmYWxzZSwidXVpZCI6ImQ5ZjNlZmMxMTEwMDQ4YzI4YzM4YzA1MjBiNDM1NmE4MjYwODg3NTA1ODc3NTc1NjIyMiIsInNlcnZlcl90b2tlbiI6IjE2MjUyMjIzMzM5MjJ8MjYwODg3NTA1ODc3NTc1NjIyMnw0NjkxMjE2ODk0M3xkYTNkZWY0OTA3NjdkZjcyNzZmZTEwODAyNTlkZmY5NDgzODdhNmFkN2RjYzMzNWY2MDdmMmNkMGQxN2NjZDA0In0sInNpZ25hdHVyZSI6IiJ9"
upcomingEvent:	null

Table A2. *Cont.*

Attribute	Value
usertags:	Array(0)
length:	0
viewerCanReshare:	TRUE
viewerInPhotoOfYou:	FALSE

References

1. Chew, A.M.K.; Gunasekeran, D.V. Social Media Big Data: The Good, The Bad, and the Ugly (Un)truths. *Front. Big Data* **2021**, *4*, 623794. [CrossRef] [PubMed]
2. Mann, M. The Max Schrems Litigation: A Personal Account. In *Institutionalisation Beyond the Nation State: Transatlantic Relations: Data, Privacy and Trade Law*; Fahey, E., Ed.; Springer International Publishing: New York, NY, USA, 2018; pp. 75–89. [CrossRef]
3. IndexedDB API. MDN Web Docs. Available online: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API (accessed on 20 May 2022).
4. MDN Web Docs. Browser Storage Limits and Eviction Criteria. Available online: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Browser_storage_limits_and_eviction_criteria (accessed on 20 May 2022).
5. Mendoza, A.; Kumar, A.; Midcap, D.; Cho, H.; Varol, C. BrowStEx: A tool to aggregate browser storage artifacts for forensic analysis. *Digit. Investig.* **2015**, *14*, 63–75. [CrossRef]
6. Kimak, S.; Ellman, J. The role of HTML5 IndexedDB, the past, present and future. In Proceedings of the 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 14–16 December 2015; pp. 379–383. [CrossRef]
7. Paligu, F.; Kumar, A.; Cho, H.; Varol, C. BrowStExPlus: A Tool to Aggregate Indexed DB Artifacts for Forensic Analysis. *J. Forensic Sci.* **2019**, *64*, 1370–1378. [CrossRef] [PubMed]
8. Mohsin, M.; Oberlo. 10 Instagram Stats Every Marketer Should Know in 2021. Available online: https://www.oberlo.com/blog/instagram-stats-every-marketer-should-know (accessed on 20 May 2022).
9. Ghafarian Ahmad and Keskin, D. Windows 10 Hibernation File Forensics. In *Intelligent Computing*; Springer International Publishing: New York, NY, USA, 2020; pp. 431–445.
10. Chang, M.S.; Yen, C.P. Forensic Analysis of Social Networks Based on Instagram. *Int. J. Netw. Secur.* **2019**, *21*, 850–860. [CrossRef]
11. Jadoon, A.K.; Iqbal, W.; Amjad, M.F.; Afzal, H.; Bangash, Y.A. Forensic Analysis of Tor Browser: A Case Study for Privacy and Anonymity on the Web. *Forensic Sci. Int.* **2019**, *299*, 59–73. [CrossRef] [PubMed]
12. Kimak, S.; Ellman, J.; Laing, C. Some Potential Issues with the Security of HTML5 IndexedDB. In Proceedings of the 9th IET International Conference on System Safety and Cyber Security, Manchester, UK, 15–16 October 2014. [CrossRef]
13. W3C. Indexed Database Specification API 2.0. Available online: https://www.w3.org/TR/IndexedDB-2 (accessed on 20 May 2022).
14. W3C. Indexed Database API 3.0. Available online: https://www.w3.org/TR/IndexedDB-3 (accessed on 20 May 2021).
15. Paligu, F.; Varol, C. Browser Forensic Investigations of WhatsApp Web Utilizing IndexedDB Persistent Storage. *Futur. Internet* **2020**, *12*, 184. [CrossRef]
16. Walnycky, D.; Baggili, I.; Marrington, A.; Moore, J.; Breitinger, F. Network and device forensic analysis of Android social-messaging applications. *Digit. Investig.* **2015**, *14*, S77–S84. [CrossRef]
17. Mushcab, R.; Gladyshev, P. Forensic analysis of instagram and path on an iPhone 5s mobile device. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 146–151. [CrossRef]
18. Pambayun, S.; Riadi, I. Investigation on Instagram Android-based using Digital Forensics Research Workshop Framework. *Int. J. Comput. Appl.* **2020**, *175*, 15–21. [CrossRef]
19. Seo, S.; Kim, Y.; Lee, C. Instagram Users Behavior Analysis in a Digital Forensic Perspective. *J. Korea Inst. Inf. Secur. Cryptol.* **2018**, *28*, 407–416.
20. Douglas, Z. *Digital Image Recompression Analysis of Instagram*; University of Denver at Colorado: Denver, CO, USA, 2015.
21. Zarei, K.; Farahbakhsh, R.; Crespi, N. Typification of Impersonated Accounts on Instagram. In Proceedings of the 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC), London, UK, 29–31 October 2019; pp. 1–6. [CrossRef]
22. Kumar, S.T.; Karabiyik, U. Instagram Forensic Analysis Revisited: Does anything really vanish? In Proceedings of the 2021 International Symposium on Networks, Computers and Communications (ISNCC), Dubai, United Arab Emirates, 31 October–2 November 2021; pp. 1–6. [CrossRef]
23. Quan, Y.; Lin, X.; Li, C.-T. Provenance Analysis for Instagram Photos. In *Data Mining*; Springer: Singapore, 2019; pp. 372–383. [CrossRef]

24. Dixon, M.W.; McGill, T.J.; Karlsson, J.M. Using a network simulation package to teach the client-server model. In Proceedings of the 2nd Conference on Integrating Technology into Computer Science Education—ITiCSE, Uppsala, Sweden, 1–5 June 1997; pp. 71–73. [CrossRef]
25. Al-Shaikh, A.; Sleit, A. Evaluating IndexedDB performance on web browsers. In Proceedings of the 2017 8th International Conference on Information Technology (ICIT), Amman, Jordan, 17–18 May 2017; pp. 488–494. [CrossRef]
26. Youn, T.-Y.; Chang, K.-Y.; Rhee, K.H.; Shin, S.U. Efficient Client-Side Deduplication of Encrypted Data with Public Auditing in Cloud Storage. *IEEE Access* **2018**, *6*, 26578–26587. [CrossRef]
27. Woods, D.; Snee, T.; Pekowsky, K. *Developer's Guide to the Java Web Server: Building Effective and Scalable Server-Side Applications with Cdrom*, 1st ed.; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1999.
28. Walker, J.D.; Chapra, S.C. A client-side web application for interactive environmental simulation modeling. *Environ. Model. Softw.* **2014**, *55*, 49–60. [CrossRef]
29. Millett, L.I.; Friedman, B.; Felten, E. Cookies and Web browser design. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems—CHI '01, Seattle WA, USA, 31 March–5 April 2021; pp. 46–52. [CrossRef]
30. Nalawade, A.; Bharne, S.; Mane, V. Forensic analysis and evidence collection for web browser activity. In Proceedings of the 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Pune, India, 9–10 September 2016; pp. 518–522. [CrossRef]
31. Ferragina, P.; Grossi, R. The string B-tree: A new data structure for string search in external memory and its applications. *J. ACM* **1999**, *46*, 236–280. [CrossRef]
32. W3C (World Wide Web Consortium). Available online: https://www.w3.org (accessed on 20 May 2022).
33. IndexedDB. Caniuse. Available online: https://caniuse.com/#search=indexedDB (accessed on 20 May 2022).
34. Browser Market Share. Available online: https://netmarketshare.com/browser-market-share.aspx (accessed on 20 May 2022).
35. Lin, J. Building a Self-Contained Search Engine in the Browser. In Proceedings of the 2015 International Conference on The Theory of Information Retrieval, Northampton, MA, USA, 27–30 September 2015; pp. 309–312. [CrossRef]
36. Liu, X.; Yu, X.; Ma, X.; Kuang, H. A Method to Improve the Fresh Data Query Efficiency of Blockchain. In Proceedings of the 2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Phuket, Thailand, 28–29 February 2020; pp. 823–827. [CrossRef]
37. Luo, H.; Jiang, H.; Yan, Z.; Yang, Y. Fast transaction logging for smartphones. In Proceedings of the 2016 32nd Symposium on Mass Storage Systems and Technologies (MSST), Santa Clara, CA, USA, 2–6 May 2016; pp. 1–5. [CrossRef]
38. Same Origin Policy. W3. Available online: https://www.w3.org/Security/wiki/Same_Origin_Policy (accessed on 20 May 2022).
39. MDN Web Docs. Using IndexedDB. Available online: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Using_IndexedDB (accessed on 20 May 2022).
40. Cook, T.D.; Campbell, D.T. The design and conduct of quasi-experiments and true experiments in field settings. In *Handbook of Industrial and Organizational Psychology*; 1976; pp. 223–326. Available online: https://www.scholars.northwestern.edu/en/publications/the-design-and-conduct-of-true-experiments-and-quasi-experiments-2 (accessed on 20 May 2022).
41. Awesome Photographers. Instagram. Available online: https://www.instagram.com/awesome.photographers (accessed on 20 May 2022).
42. Sqlitebrowser. DB Browser for SQLite. Available online: https://sqlitebrowser.org (accessed on 20 May 2022).
43. An, J.; Li, T.; Teng, Y.; Zhang, P. Factors Influencing Emoji Usage in Smartphone Mediated Communications. In *Transforming Digital Worlds*; Springer: Cham, Switzerland, 2018; Volume 10766, pp. 423–428. [CrossRef]
44. Pyrooz, D.C.; Moule, R.K., Jr. Gangs and Social Media. In *Oxford Research Encyclopedia of Criminology and Criminal Justice*; Oxford University Press: Oxford, UK, 2019. [CrossRef]
45. Marengo, D.; Giannotta, F.; Settanni, M. Assessing personality using emoji: An exploratory study. *Pers. Individ. Differ.* **2017**, *112*, 74–78. [CrossRef]
46. McMahon, M.; Kirley, E.A. When Cute Becomes Criminal: Emoji, Threats and Online Grooming. *Minn. JL Sci. Tech. Sci. Technol.* **2019**, *20*, 37–92.
47. Christidis, A.; Davies, R.; Moschoyiannis, S. Serving Machine Learning Workloads in Resource Constrained Environments: A Serverless Deployment Example. In Proceedings of the 2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA), Kaohsiung, Taiwan, 18–21 November 2019; pp. 55–63. [CrossRef]
48. PHP Manual. SQLite3. Available online: https://www.php.net/manual/en/book.sqlite3.php (accessed on 20 May 2022).
49. Storage Inspector. MDN Web Docs. 2021. Available online: https://developer.mozilla.org/en-US/docs/Tools/Storage_Inspector (accessed on 20 May 2022).
50. Developers Google. Chrome DevTools. Available online: https://developers.google.com/web/tools/chrome-devtools (accessed on 20 May 2022).
51. WAMP Server. Available online: https://www.wampserver.com/en/ (accessed on 20 May 2022).
52. Liu, S.; Zhu, M.; Yu, D.J.; Rasin, A.; Young, S.D. Using Real-Time Social Media Technologies to Monitor Levels of Perceived Stress and Emotional State in College Students: A Web-Based Questionnaire Study. *JMIR Ment. Health* **2017**, *4*, e2. [CrossRef] [PubMed]