

B2B Service for Tracking User Behavior

Bui Ngoc Son[1], Vu Thu Diep[2], and Phan Duy Hung[1(✉)]

[1] FPT University, Hanoi, Vietnam
{sonbn7,hungpd2}@fe.edu.vn
[2] HaNoi University of Science and Technology, Hanoi, Vietnam
diep.vuthu@hust.edu.vn

Abstract. This paper presents a solution for collecting user action data for the B2B model. Collecting user data plays an important role in personalized experience, product improvement and business strategy optimization. Data helps Internet platforms publish personalized content and products, and helps businesses optimize subsequent campaigns and improve advertising performance. In addition, data also helps improve customer service, detect errors and prioritize user interfaces. In the field of AI and Machine Learning, data is an important foundation for training models, making the system smarter and more automated. By effectively exploiting user data, businesses can make accurate decisions, optimize operating processes and anticipate market trends, thereby improving performance and quality services. Collecting user data not only helps individuals experience and optimize business strategies, but also opens up opportunities for businesses to provide data collection and analysis services in a B2B (Business to Business) model. Many companies do not have the ability to automatically collect and process big data, so they turn to third-party data aggregation services from various sources. These businesses use the data to optimize subsequent campaigns, anticipate consumer trends, and improve products. The approach and results of this discussion can be fully applied to other types of user data.

Keywords: B2B Service · Data Collection · User Tracking · User Behavior

1 Introduction

1.1 Problem and Motivation

Currently, the demand for B2B (business-to-business) is growing strongly in the era of technology 4.0, especially in many fields when businesses increasingly focus on digital transformation and optimizing work processes.

In software services and digital platforms (SaaS), businesses are looking for technology solutions to help manage data, automate processes using platforms such as customer relationship management (CRM), supply chain management (SCM), data analysis tools, etc. In particular, the analysis tools of these platforms often collect information about user behavior such as activity history, usage habits to optimize the experience as well as predict needs.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15891, pp. 3–16, 2026.
https://doi.org/10.1007/978-3-031-97617-9_1

With digital marketing for B2B, not only stopping at traditional surveys, many businesses have applied digital marketing strategies such as SEO, digital content, and advertising on online platforms. And on these platforms, collecting user data is also very popular, however, the collected data is somewhat different from management platforms such as: collecting emails, clicks, website behavior, personal information, etc. to be able to deploy personalized campaigns, display ads suitable for each audience. Not only that, more and more businesses also customize according to customer needs according to industry, product or business model characteristics. The data collected from these models is quite diverse so that businesses can personalize experiences or create unique products and services.

One type of user website data collection is gathering user location data. When collecting the location of many users in a certain number, analysts can determine the number of customers interested in which products, thereby developing reasonable strategies with stores at the counter as well as suggesting reasonable products. Another example of data collection is collecting personal information of users when they register to use website features, valuable information such as age, address or transaction history can help analysts have a deeper insight into the level of interest of users in products depending on age or geography. Not only that, some websites and apps such as TikTok, Reddit also create surveys about interests, things that users are interested in, thereby displaying content, products or creating features with the lowest risk while still optimizing profits. In addition, some large e-commerce websites such as Amazon and eBay also collect users' product clicks for analysis, and when the number of clicks on a product reaches the threshold where the AI knows what product the user is interested in, the website display that product along with countless related products to retain users as well as increase the conversion rate much higher than displaying products on the website randomly and illogically.

Collecting e-commerce web user behavioral data through computer mouse information is part of analyzing user habits. Computer mouse information not only tells analysts about user behavior and habits on the website, but also helps them understand user emotions as mentioned [1]. Not only that, this is an easy-to-collect type of data that also has many meanings. In addition to expressing emotions, computer mouse information also shows analysts all user activities on the website based on the path of the mouse, products of interest based on clicks, click density or movement on the website.

Computer mouse information can include many factors such as mouse coordinates, click coordinates, mouse scrolling, etc. and when these data are large enough, they can let analysts know whether the website's UI is good or not. For example, the rate of missing a button click or areas on the website that are less interested. From there, the user experience can be optimized. In addition, the data also shows which products users are interested in based on how long their mouse lingers on that product. So that businesses can optimize advertising, display products that meet user needs, thereby increasing conversion rates as well as traffic. However, collecting computer mouse information is still very difficult to implement for many small and medium-sized businesses. The first problem is the huge amount of data because the mouse can move very quickly as well as perform countless other actions in a moment, thereby creating a lot of data in a short period of time. Second, the storage capacity and understanding of big data technologies

of many businesses or digital transformation companies are still limited when they only collect simple information such as button clicks, visits, and that data is not good enough to be used to truly analyze user behavior. Third, many businesses that want to collect and analyze this data need the support of third parties at a high cost and can be blocked by the user's own extensions. And finally, compliance with ethics and privacy regulations such as the General Data Protection Regulation (GDPR) in Europe or the California Consumer Privacy Act (CCPA) in the US must also be a top priority.

1.2 Related Work

The results of some studies have been found, for example: In [2], the author presented an advanced website analysis support tool that focuses on tracking mouse behavior and processing real-time data to optimize the efficiency of data collection on the website. This tool uses the mouse path vectorization technique, which significantly reduces the amount of data transmitted through the JavaScript library but still ensures high accuracy. The system includes a vectorizer to optimize and compress data, a log tracker to track, and also supports session storage and playback of user sessions. The tests have shown amazing results when it can reduce up to 85% of the transmitted data size and 84% of the number of requests. And the tool still works well under limited network conditions. The tool not only supports user experience analysis but also has the ability to expand when integrated with large data analysis systems such as Google Analytics. In [3], a method combining user modeling, mouse behavior tracking and recommendation system is presented to enhance user experience on e-commerce platforms. The study has built a recommendation algorithm based on user interaction data, including click events, mouse movements and mouse hover time. The results show that the system achieves 87.50% accuracy in recommending suitable products and received positive reviews from 40 survey participants. AIMT-UXT mentioned in [4] is an open tool that supports the collection, storage and analysis of computer mouse information on the web, integrating many analytical modules such as single-view, heatmap and fuzzy logic to evaluate UX and t-SNE clustering algorithm to group users according to behavioral characteristics. The goal is to introduce a method for evaluating user experience through mouse tracking data and artificial intelligence (AI). The system with high reliability when compared with the traditional method based on questionnaire (SUS), UX classification assessment makes it a potential tool for low-cost UX assessment without depending on any commercial solutions.

During the research, real-time processing systems all apply Socket.IO[1] so that the Client can communicate with the Backend in the most efficient and optimal way. Some systems proposed in [5–7] all apply it and achieve high efficiency. Therefore, this study also apply it in this system.

In paper [8] it is concluded that HDFS is a powerful distributed file system that plays a core role in big data processing applications. It is designed to provide efficient and reliable data storage and processing, meeting the needs of large-scale systems. From there, this paper decided to implement it in this paper.

[1] https://socket.io/.

Spark Streaming[2] processes data continuously, meaningful data is processed and analyzed as it is generated. This is useful for applications that require real-time feedback or continuous data analysis such as financial transactions, monitoring systems, and social media stream processing. Spark Streaming is applied in many fields such as [9–12]. Therefore, it also be implemented in this research. In [13–15], very promising results have been achieved using Apache Kafka[3], which is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications. On the other hand, in paper [16], combining Kafka and Spark Streaming to build a system that delivers high performance. So Kafka also be used here along with Spark Streaming.

This paper takes a different approach, this work focuses on collecting user computer mouse information and transmitting this data accurately and minimizing loss as much as possible with the big data processing technologies mentioned above but still ensuring simplicity in deployment and integration into the system without changing the current structure of the website.

2 Contribution

The main contribution of this paper is to propose a service to collect user behavior through computer mouse information. Each step has been studied and evaluated to find a suitable processing method for automatic data collection. The service is implemented with 3 simple steps: Collect data, send data, stream data and store. In which, data collection includes data preprocessing. This work also tested the results and performance when deploying the system on the test environment which is mentioned in the results section of this thesis. The rest of the thesis is organized as follows. Section 2 describes the system architecture, the implementation method is analyzed in Sect. 3. Finally, the implementation results and conclusions are discussed in Sect. 4.

3 System Architecture

3.1 Architecture Overview

In general, the model this paper propose is deployed for businesses that provide data collection services (Service Provider) to other businesses in need (Business Partner). Through the Internet, businesses in need of data collection will connect with the service provider. And businesses can collect user data, analyze website data after deploying additional functions and functions according to the provider's API description. The basic system architecture is shown in Fig. 1.

Going into Business Partner, the website of businesses that want to track user computer mouse information, JavaScript is the best choice because of the language's ability to access the Document Object Model (DOM). Then, the data collected from JavaScript can be sent to the server in many different ways. However, sending a huge amount

[2] https://spark.apache.org/.
[3] https://kafka.apache.org/.

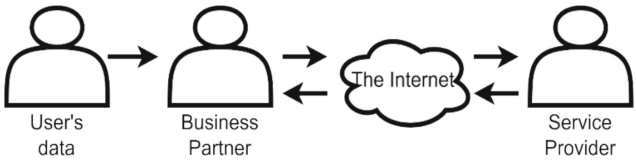

Fig. 1. Architecture overview of B2B Service

of data from the user's mouse to the web server does not have many options. Therefore, Socket.IO will be deployed into the website system to bridge the communication between the layers. As for the web server, the website will have to add some functions according to the API description to be able to encrypt and send the collected data to the Service Provider. The data sent from Business Partner must ensure security standards such as HTTPS or TLS to ensure that the data is not attacked. The system architecture of Business Partner is shown in Fig. 2.

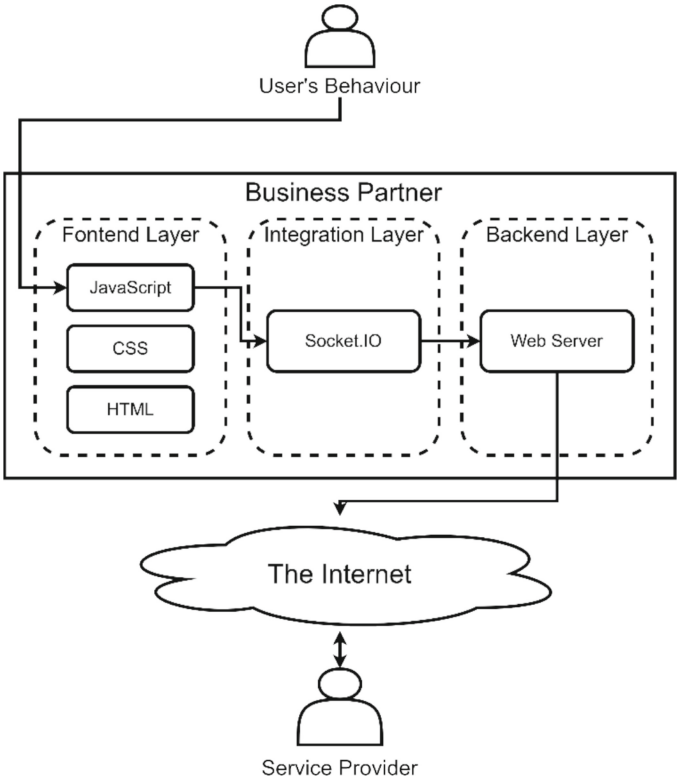

Fig. 2. Architecture of Business Partner

As for the Service Provider, the system has an API gateway that ensures the use of necessary security to ensure that it is not attacked, as well as authenticating and authorize

requests from Business Partners using security methods such as OAuth, API Key, etc. And the API Gateway also coordinates requests to other components in the system. Data after authentication via the API gateway is uploaded to Kafka and then stored in HDFS through Spark. Data in HDFS is processed by Spark and sent to Business Partner or displayed directly on the system dashboard depending on the needs. The system architecture of the Service Provider is shown in Fig. 3 and the service of the proposed system includes the following subcomponents are listed in detail in Table 1.

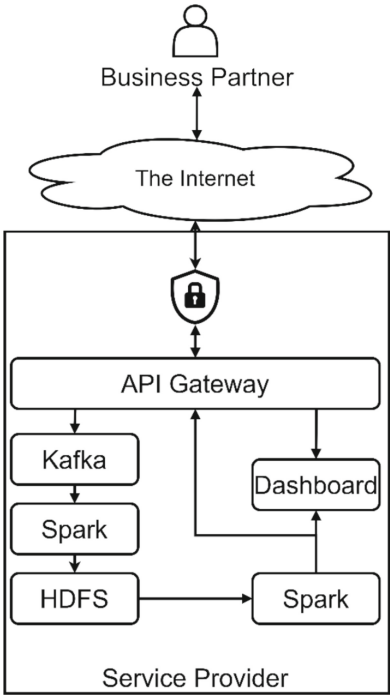

Fig. 3. Architecture of Service Provider

Table 1. Subcomponents included in proposed system

Component	Description
Kafka	Kafka creates topics, each topic can represent a user, and with the strength of receiving and distributing data, Kafka takes on the role of continuously receiving data from the server into the specified topic and distributing them efficiently into HDFS through Spark
Spark Streaming	Spark with the strength of processing and analyzing big data, will be used to analyze continuously updated data in HDFS to visualize data for analysts, as well as taking on the task of broadcasting data from Kafka and saving to HDFS
HDFS	Data received from Kafka will be distributed into blocks, along with creating and distributing data copies on Data nodes to increase retrieval speed, ensure availability and reliability, load balancing as well as better scalability

3.2 System Requirements

To meet the needs of the experiment, a website in Python with Flask Framework is created. This website is deployed on the test platform. The test platform here includes a physical machine with Intel I5-13600KF 3.50 GHz chip, 8GB RAM, Windows 11 64 bit, and a virtualization platform Oracle VirtualBox Version 7.1.0 r164728, 4GB RAM, 2 CPUs, PIIX3 chipset, Ubuntu 22.04 LTS operating system. And the software's versions are listed in detail in Table 2.

Table 2. Version of software and systems be used

Name	Version
Python	3.10.12
Flask	3.0.3
Socket.IO	4.1.2
Hadoop	3.4.0
Kafka	3.8.0
Scala	2.13
Zookeeper	3.8.4
Spark	3.5.3

Python, Flask, and Socket.io are used with the most popular versions to create a Business Partner test environment. The same applies to Hadoop, Spark, and Kafka as their versions are well compatible with each other. Zookeeper and Scala are installed alongside the Kafka installation.

4 System Design and Implementation

4.1 Business Partner Layer

Client

In this system, Python and Flask are used to create a sample website to serve as a testing environment. First, the Frontend is embedded with a JavaScript code consisting of the following components listed in Table 3:

Table 3. Components included in the Frontend of the Client

Component	Description
socket	Connects to the server to receive data from the frontend
port	The port of the server
Mouse_Coordinate	The name of the event that receives mouse coordinate
Clicked_Coordinate	The name of the event that receives mouse's clicked coordinate

(*continued*)

Table 3. (*continued*)

Component	Description
document	An object that represents the entire HTML in the browser, allowing JavaScript to interact and manipulate the content of the website
X-coordinate	The horizontal coordinate of the mouse
Y-coordinate	The vertical coordinate of the mouse
Button clicked	Parameter that determines whether the mouse is being clicked
Mouse_Event	Name of channel on socket

Process 1 presents the process of collecting data using script and sending collected data to socket's channel. Line 1 initializes socket on specified port. Line 2–9 presents the process of handling "Mouse_Coordinate" event. When document detects the changing of mouse movement, retrieves X-coordinate and Y Coordinate from web page and create an object to store it (Line 3–7). Then send it's coordinate to channel on socket (Line 8). In addition, the process of handling "Clicked_Coordinate" event (Line 10–19) has the same structure as "Mouse_Coordinate" but with the addition of retrieving button clicked.

Process 1: Collecting data using script and sending collected data to socket's channel

```
1. INITIALIZE socketConnection TO CONNECT TO specifiedPort
2. WHEN document DETECTS 'Mouse_Coordinate' event:
3.         SET mouseX TO RETRIEVE X-coordinate FROM web page
4.         SET mouseY TO RETRIEVE Y-coordinate FROM web page
5.         CREATE moveData OBJECT WITH:
6.                 PROPERTY x = mouseX
7.                 PROPERTY y = mouseY
8.         SEND moveData TO 'Mouse_Event'
9. END WHEN
10. WHEN document DETECTS 'Clicked_Coordinate' event:
11.         SET mouseX TO RETRIEVE X-coordinate FROM web page
12.         SET mouseY TO RETRIEVE Y-coordinate FROM web page
13.         SET mouse_clicked TO RETRIEVE button clicked FROM web page
14.         CREATE clickedData OBJECT WITH:
15.                 PROPERTY x = mouseX
16.                 PROPERTY y = mouseY
17.                 PROPERTY clicked = mouse_clicked
18.         SEND clickedData TO 'Mouse_Event'
19. END WHEN
```

And from there, both the mouse path and the location where the user's mouse clicks are collected. Each user activity is collected through the script and saved on the socket. From there, the web server will collect and process for the next steps.

Web Server
In addition to operating the website, the Web Server also takes on two other tasks: processing data from the client and pushing processed data to Kafka. In this part, the components for operation include in Table 4:

Table 4. Subcomponents included in the Web Server

Components	Description
producer	Used to send data to the specified topics of Kafka
Kafka producer	Is a class provided by the Kafka library
Bootstrap servers	Parameter used to assign the address of the Kafka broker
Value serializer	Parameter used to serialize data before sending it to Kafka

Process 2 presents Kafka's producer, which is used to send data from business partner to service provider. The producer configuration has 2 parts. First is setting the bootstrap server with the address that is provided by the service provider (Line 2). And the second is the data sent by the producer is guaranteed to be serialized before sending (Line 3).

Process 2: Kafka producer Configuration

1. **INITIALIZE** Kafka producer **WITH CONFIGURATIONS**:
2. **SET** bootstrap servers **TO** Kafka broker address
3. **SET** value serializer **TO** Json Serializer

The process 3 below extract data when received it on socket's channel (Line 1). After extracting X value, Y value and clicked from data to x, y, clicked (Line 3–5), create structureData with extracted value (Line 6), then use initialized producer in process 2 to send it to service provider's Kafka's topic (Line 7).

Process 3: Extracting data on Socket's channel and using Producer to send data

```
1. WHEN 'Mouse_Event' on socket IS RECEIVED:
2.     FUNCTION handleMouseEvent(data):
3.         SET x TO EXTRACT 'X value' FROM data
4.         SET y TO EXTRACT 'Y value' FROM data
5.         SET clicked TO EXTRACT 'clicked' FROM data
6.         CREATE structuredData WITH x, y, clicked
7.         USE producer TO SEND structuredData TO Kafka's topic IN Kafka Broker
8.         FLUSH ALL messages TO ENSURE IMMEDIATE SENDING
9.   END FUNCTION
10. END WHEN
```

After performing the above processes, the data from here is transferred to the Service Provider for processing for the next steps. This issue is presented in the section below.

4.2 Service Provider Layer

Kafka, Spark and HDFS

Producer from Business Partner after sending data to the pre-designated topic on Kafka is backed up to ensure reliability as well as prevent data loss. Then ready for Consumers to subscribe and read data from the partition of that topic.

Process 4 shows the process of initializing Spark Consumer to subscribing the topic on Kafka Broker to DataFrame. From the beginning, data source format set as Kafka's format (Line 2), then define the address of Kafka Broker and the topic which is the target of subscribing (Line 3 – 4). In addition, the message should be set to read from earliest to ensure the correctness (Line 5). After those setup steps, the data will be streamed.

Process 4: Initialize Consumer and read data from Kafka using Spark to DataFrame

```
1. INITIALIZE DataFrame TO READ and LOAD FROM Kafka USING Spark WITH
   CONFIGURATIONS:
2.     WITH FORMAT 'kafka'
3.     WITH OPTION 'bootstrap.servers' SET TO 'Kafka broker address'
4.     WITH OPTION 'subscribe' SET TO 'Kafka's topic'
5.     WITH OPTION 'startingOffsets' SET TO 'earliest'
```

In process 5, DataFrame must pre-process before writing into HDFS. First, define schema for DataFrame with "x", "y", "clicked" as integer, integer, string (Line 1 – 4).

Then parse DataFrame with steps: Convert “value” column to string and set to json (Line 6). After setting json, parse json use defined schema and set to parsedData. Then extract and select all fields from parsedData (Line 7 - 8).

Process 5: Pre-Processing data

1. **DEFINE** schema **FOR** DataFrame **WITH FIELDS**:
2. 'x' **AS INTEGER**
3. 'y' **AS INTEGER**
4. 'clicked' **AS STRING**
5. **PARSE** DataFrame:
6. **SET** json **TO CONVERT** 'value' **COLUMN TO STRING**
7. **SET** parsedData **TO PARSE** json **USING** schema
8. **EXTRACT** and **SELECT** all fields **FROM** parsedData

Process 6 presents the process of writing prased DataFrame into HDFS. Strating write data from parsed DataFrame into HDFS with configurations. First, set “outputMode” to “append” (Line 2), which defines how it interacts with HDFS. Second is setting format when appending (Line 3), “json” here is selected format. Finally, set the save location and checkpoint location in HDFS (Line 4 – 5).

Process 6: Write parsed DataFrame to HDFS

1. **START WRITING** data **FROM** parsed DataFrame **WITH CONFIGURATIONS**:
2. **SET** outputMode **TO** 'append'
3. **SET** format **TO** 'json'
4. **SET** outputLocation **TO** 'HDFS folder location'
5. **SET** checkpointLocation **TO** 'HDFS checkpoint location'
6. **WAIT INDEFINITELY FOR** streaming query **TO TERMINATE**

Visualize with Spark and HDFS

User mouse data stored in HDFS is continuously uploaded by Spark via Kafka and is also continuously streamed by another Spark session for analysis and visualization purposes. Here, Spark data fetched from HDFS is divided into two separate sets called dataMove and dataClicked. With dataMove, the data only has the mouse coordinates, used to get the user’s movement data on the website. Meanwhile, dataClicked has an additional mouse click attribute to determine the location on the page where the user has clicked. Both data types are shown in Table 5:

Table 5. Structure of data's mouse information

	X coordinate	X coordinate	Clicked
dataMove	O	O	–
dataClicked	O	O	O

Both data sets, when displayed together on a graph on the dashboard, provide the clearest view for analysts when studying user behavior. This data can also be exported in Json or csv format if desired.

4.3 Implementation

A simulated E-commerce website using Flask has been deployed. The front end of the website has been embedded with JavaScript code to collect mouse information as mentioned in Process 1, and the backend has also been added with a function to send data to the Service Provider as shown in Process 2 and 3. The sample web page is shown in Fig. 4.

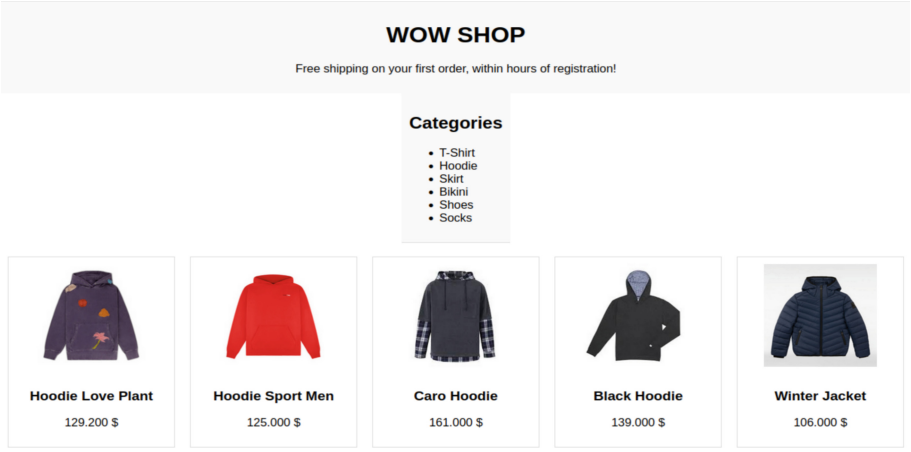

Fig. 4. Experimental Website

To simulate a user using and interacting with the website, the computer mouse has been used to interact with the website for a while. When moving the mouse, all mouse information is sent to the socket to communicate with the server. Through the processing steps from Process 4 to 6, Service Provider obtain the user's behavior when interacting with the mouse on the Web page. The results are presented in Fig. 5.

The test results show us which products are of interest even without any analysis tools based on click density as well as customer activity area in a short time. This information when extracted is in Json format as shown in Fig. 6.

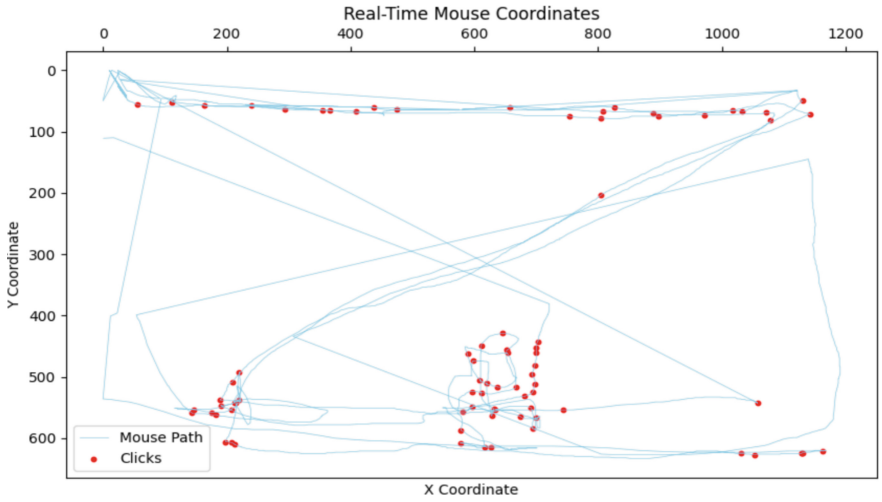

Fig. 5. Experiment Result


```
{"x":578,"y":557}
{"x":580,"y":557}
{"x":581,"y":557}
{"x":581,"y":557,"clicked":"0"}
{"x":582,"y":557}
{"x":583,"y":557}
{"x":584,"y":557}
{"x":586,"y":556}
{"x":587,"y":556}
{"x":588,"y":556}
```


Fig. 6. An example of the exported data in Json format

From the exported data, the results shown that the information above is simple and convenient enough to be used for other analytical tasks. Moreover, with the amount of information extracted, providing data for machine learning tasks is completely simple.

5 Conclusion and Perspective

In conclusion, this paper introduces a method to deploy a B2B service that collects user data. It is worth noting that this method is very flexible and effective when it is extremely easy to deploy for both Business Partners and Service Providers. In particular, Business Partners do not need to intervene too deeply in their systems when applying this method. In addition, the method is also very scalable when adding parameters to the code is also extremely simple, thereby helping Service Providers customize the code to create different services based on this study's method.

In this work, this paper focuses on implementing mouse data collection in an easy and efficient way. In the future, this study aims to extend this method and map it to other collection scenarios while still ensuring user data privacy laws. This work is designed with concurrency and scalability in mind and considers improving performance and data collection in more complex scenarios.

References

1. Lali, P., Naghizadeh, M., Nasrollahi, H., Moradi, H., Mirian, M.S.: Your mouse can tell about your emotions. In: 4th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 47–51. IEEE, Mashhad (2014)
2. Čegan, L., Filip, P.: Advanced web analytics tool for mouse tracking and real-time data processing. In: 14th International Scientific Conference on Informatics, pp. 431–435. IEEE, Poprad, Slovakia (2017)
3. Tanjim-Al-Akib, M., Ashik, L.K., Walid, H.-A., Chowdhury, K.: User-modeling and recommendation based on mouse-tracking for e-commerce websites. In: Computer and Information Technology (ICCIT), pp. 517–523. (2016)
4. Souza, K.E.S., Seruffo, M.C.R., De Mello, H.D. Souza, D.D.S., Vellasco, M.M.B.R.: User experience evaluation using mouse tracking and artificial intelligence. IEEE Access **7**, 96506–96515 (2019)
5. Josh, Godwin Sam. Asynchronous Wi-Fi Control Interface (AWCI) Using Socket IO Technology. arXiv preprint arXiv:1810.05502 (2018)
6. Singh, Y.V., Singh, H., Chauhan, J.K.: Online collaborative text editor using socket.IO. In: 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 1251–1253. Greater Noida, India (2021)
7. Rajarajeswari, P.L., et al.: Reach—a chat application. In: 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1961–1966. Coimbatore, India (2024)
8. Shvachko, K., Kuang, H., Radia, S., Chansler, R.: The hadoop distributed file system. In: 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1–10. Incline Village, NV, USA (2010)
9. Chung, N.N., Hung, P.D.: Logging and monitoring system for streaming data. In: Luo, Y. (eds.) Cooperative Design, Visualization, and Engineering. CDVE 2020. LNCS, vol. 12341. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60816-3_21
10. Hung, P.D., Hanh, T.D., Tung, T.D.: Term deposit subscription prediction using spark MLlib and ML packages. In Proceedings of the 2019 5th International Conference on E-Business and Applications (ICEBA 2019), pp. 88–93. Association for Computing Machinery, New York, NY, USA (2019)
11. Dat, D.Q., Hung, P.D.: Clustering of time-series balance history data streams using apache spark. In: Luo, Y. (eds.) Cooperative Design, Visualization, and Engineering. CDVE 2020. LNCS, vol. 12341. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60816-3_13
12. Nam, L.H., Hung, P.D.: Building a big data oriented architecture for enterprise integration. In: Luo, Y. (eds.) Cooperative Design, Visualization, and Engineering. CDVE 2021. LNCS, vol. 12983. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88207-5_17
13. Alothali, E., Alashwal, H., Salih, M., Hayawi, K.: Real time detection of social bots on twitter using machine learning and apache kafka. In: 2021 5th Cyber Security in Networking Conference (CSNet), pp. 98–102. Abu Dhabi, United Arab Emirates (2021)
14. Ilasariya, S., Patel, P., Patel, V., Gharat, S.: Image steganography using blowfish algorithm and transmission via apache kafka. In: 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 1320–1325. Tirunelveli, India (2022)
15. Phan, T.-C., Phan, A.-C., Trieu, T.-N.: Real-time opinion extraction and classification for vietnamese posts on social networks. In: 2020 12th International Conference on Knowledge and Systems Engineering (KSE), pp. 19–24. Can Tho, Vietnam (2020)
16. Le Noac'h, P., Costan, A., Bougé, L.: A performance evaluation of apache kafka in support of big data streaming applications. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 4803–4806. Boston, MA, USA (2017)