Journal of Applied Data Sciences
Vol. 3, No. 1, January 2022, pp. 24-32

ISSN 2723-6471

Application of Hash Sha-256 Algorithm in Website-Based Sales Software Engineering

Fauziah Nikmatul Khasanah [1,*]

Universitas Amikom Purwokerto, Purwokerto, Indonesia
[1] fanyfauziah17@gmail.com*
* corresponding author

(Received: November 14, 2021; Revised: December 23, 2021; Accepted: January 2, 2022; Available online: January 25, 2022)

Abstract

Rapid technological developments can spur changes in the cycle of human activities, one of which is software engineering activities that continue to develop in accordance with technological developments, the development of software engineering activities is also developing methods of data security to withstand attacks from irresponsible parties. answer. This research was conducted to analyze the performance and robustness of the sha-256 data security method with ciphertext customization. The steps that the researchers took in conducting the analysis were collecting theory and case examples, designing programs, implementing programs, testing and saving the results. Based on this process, it can be concluded that ciphertext customization on sha-256 is needed to strengthen security and resistance to attacks from irresponsible parties, besides that the performance of sha-256 calculations with customization on ciphertext and without ciphertext is not too much different where only 65 ms difference based on the results of performance testing that researchers did.

Keywords: Technology Web-Based Sales; Hash Sha-256; Data Mining; Machine Learning

1. Introduction

Rapid technological developments can spur a change in the cycle of human activity, one of which is software engineering activities that continue to develop in accordance with technological developments. Software engineering is the application of a quantitative, disciplined and systematic approach to software development, operation and maintenance [1]. Another definition of software engineering is the creation and use of engineering principles to obtain economical, reliable and efficient software on real machines [2,3]. Another definition of engineering According to IEEE 610.12 a systematic study of approaches and applications, operations development discipline and software maintenance, all of which are engineering applications related to software [3-5]. The result of software engineering is an application system, whether based on mobile, desktop or website that has functions according to user needs, in this study researchers discuss the results of website-based software engineering or commonly known as website-based information systems [6].

This website-based information system can be operated in a computer network that is connected to the internet, therefore this information system is easy to access anytime and anywhere [7]. Security in a system is the most important part in system development so it requires securing the data stored in it [8]. This is because there are many threats from outside parties for the system being built, one of which is the theft and burglary of system data [9]. Efforts to secure data on websites can be done in various ways, one of which is to apply data encryption algorithms, one of which is a hash algorithm [10]. The hash algorithm is an algorithm that converts information data in the form of letters, numbers or other characters into encrypted characters with a fixed size; data that has been encrypted through a hash function cannot be returned or decrypted. Based on the explanation above, it can be concluded that the

hash algorithm is an algorithm used to secure data on a system by encrypting or changing data in a random way with a fixed size using a hash function [11].

The hash function used in this study is the SHA-256 hash function. Secure Hash Algorithm (SHA) 256 is a commonly used hash function, so far no one has been able to solve the SHA-256 hash function algorithm [12,13]. SHA-256 is a one-way hash function designed by The National Institute of Standards and Technology (NIST) in 2002 [14]. In SHA-256 there are three additional new functions from the previous SHA type, namely SHA-1. This hash function functions in the integrity of a message: any changes to the message, with a high probability will result in a different message digest [15]. So SHA-256 is called safe. The large number of uses of the SHA-256 hash function and research that discusses the analysis of the security level of the SHA-256 hash function makes researchers interested in conducting research entitled Website-Based Software Engineering as a Sales Media With Hash Algorithm which aims to determine the level of data security on the website [16].

2. Research Methods

1) Collect theory and case examples

In this study, researchers collect theories and case examples from previous studies as well as e-books that discuss the application of hash algorithms.

2) Program design

At this stage the researcher designs the program starting from the design of the program flow to the design of the implementation of the hash algorithm to the program.

3) Program implementation

At this stage the results of the design will be executed into a program using the PHP Codeignither programming language, HTML, and CSS.

4) Test

Based on [17] hash performance testing can be done through testing the time performance required by the system in carrying out the encryption and decryption process on data 15 times to get travel time, testing all time results and then taking the average in milliseconds.

The data to be encrypted on this website comes from user input in the form of account data, sales and so on [18]. In addition, customization of the ciphertext results is carried out to avoid attacks from the decryptor by inserting a few letters or numbers into the ciphertext results to make it difficult for the decryptor to know the data encryption method used [19].

3. Discussion

1) Collect theories and cases

At this stage the researcher collects theories from various sources with the same theme to be used as a reference in implementing the hash-SHA 256 algorithm in this study as an identification of problem solving and to determine the functional requirements of the hash algorithm and non-functional requirements on the system to be built [20]. This stage produces functional and non-functional hash requirements on the system that will be built by the researcher.

2) Designing program

The application design carried out by the next researcher is designing context diagrams, implementing and customizing the SHA-256 hash algorithm, and designing user interface designs. The following are the results of the sha-256 hash algorithm customization design that the researchers made:

ISSN 2723-6471

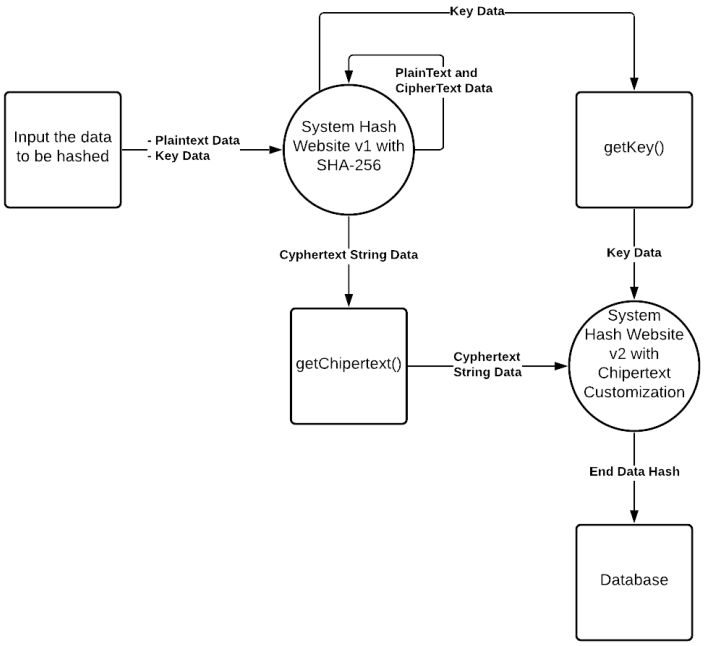


Figure. 1. Hash Sha-256 customization flow design

In the picture above, it can be concluded that customization is carried out through two processes, namely the first hash process with SHA-256 and custom ciphertext with the final result being stored in the database.

First hash with SHA-256

This first process will receive plaintext string data input and keywords that will be used for the first hashing process with the ciphertext returned data from the hashing and key that will be used for the last customization.

Second hash with ciphertext customization

In this process, the created function will receive ciphertext string data and a key in the form of a username or other data from the previous function's return data for customization. The following is the flow of ciphertext data customization carried out by researchers:

(1) Retrieve key data in the form of a username or other key.
(2) Doing split or breaking the index-index key or username that has been taken.
(3) Combine key data into ciphertext.
(4) Re-encrypt the data that has been combined above using SHA-256 and save the mature data into the database.

3) Program Implementation

In this study, the implementation of the SHA-256 hash algorithm is carried out using the hash() function that already exists in PHP. The customization model can be seen in the table below.

keyChipertext	Chipertext Murni	Date **(for code token)**

Figure. 2. Model formation of ciphertext customization

ISSN 2723-6471

KeyChipertext is obtained from substring processing with the substr function to retrieve several digits of characters that exist in certain text. After the keyChipertext was obtained, the researcher combined the keyChipertext, pure ciphertext and date for the token code requirement with Susan as above. For example, the researcher took the request reset password function that the researcher made on a website that was built as shown below.

```
public function requestreset(){
    $this->db->select('email');
    $this->db->from('acount');
    $queryAcount = $this->db->get()->result();
    foreach( $queryAcount as $row )
    {
         $email = $row->email;
         $chipPass = hash("sha256", $email);
         $keyPass = substr(str_shuffle($email), 2, 7);
         $data=array(
           'code'=> hash("sha256", $keyPass.$chipPass.date("Y.m.d")),
           'email'=> $email
           );
         $updatingCode = $this->db->set('code', $data['code'])
                                  ->where('email', $email)
                                  ->update('acount');
         $this->mailer->send($data);
    }
  }
```

Figure. 3. The function of request reset password

The data used is email data "fani@gmail.com" and password data "12345678abcde". The email data "fani@gmail.com" will be split using the substr function and scrambled using the shuffle string function with reference indexes 2 and 7 so that the ".aiamgo" key is obtained which will be used as keyChipertext. The password data "12345678abcde" will be hashed first using the sha256 function and combined with the keyCipertext and the current date according to the key formation model above. So the data becomes = ".aiamgo70ec8f0c7c3f334817297fe4e28879d2a7adcc0506ae9f20f16f14c3121284432021.01.01". Then the above data is encrypted using SHA-256 where in the process of forming the SHA-256 ciphertext it goes through stages where the data will be encrypted and converted into binary form to then get padding. Padding is done by adding bits '1' and the remaining bits '0' until the message length is congruent with 448 modulo 512. Furthermore, the message that has been padded is divided into N 512 bit blocks. Each 512-bit block obtained is then split into 16 32-bit words which are usually in the form of variables M0(i), M1(1), …, M15(i) which will later be expanded to 64 words containing the label W0, W1, …., W63 with the rules that have been set by the standard algorithm sha-2. As in the image below.

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16} & 16 \le t \le 63 \end{cases}$$

Figure. 4. Standard calculation formula sha-2

With the functions $\sigma 0$ and $\sigma 0$ formulated as follows:

$$\sigma 0(x) = ROTR7\ (X) \oplus ROTR18\ (X)\ \theta SHR3\ (X) \quad (1)$$

$$\sigma |(x) = ROTR17(X) \oplus ROTR19\ (X)\ \theta SHR10\ (X) \quad (2)$$

ISSN 2723-6471

The following is the rotation and shift calculation formula that researchers can explain:

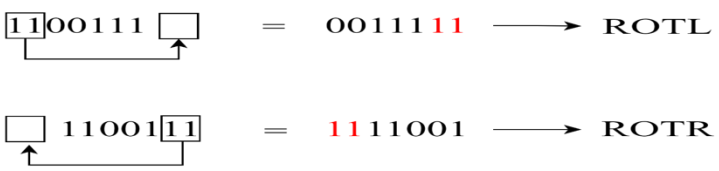


Figure. 5. Calculation of rotation

The picture above explains how to calculate Rotation Left and Rotation Right in the calculation of ciphertext sha-2 where it can be concluded that the hexadecimal data contained in the constant sha-2 will be converted into binary numbers first for rotation according to the standard formula of the hash algorithm sha- 2, ROTR (Rotation Righ) works by rotating from right to left while ROTL (Rotation Left) works by rotating from left to right as shown above.

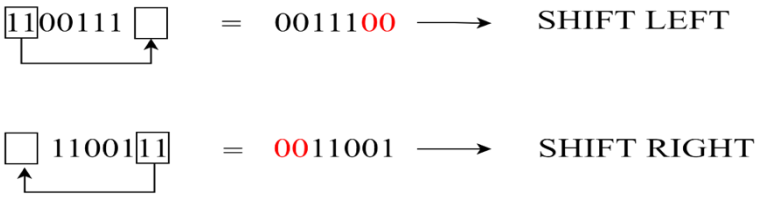


Figure. 6. Calculation of shift

The picture above explains how SHIFT RIGHT and SHIFT LEFT work in the calculation of ciphertext sha-2 where the compressed binary numbers from the hexadecimal in each constant are shifted both left and right shifts by changing bits with a value of 0 accordingly. with the number of bits shifted to the opposite side as described in the figure above. Each of the 64 words that have the label W0, …Wn is then processed with the SHA-256 hash function algorithm as described above. In this process, the SHA-256 algorithm creates 8 variables which are assigned values for the initial values of H0(0) - H7(0) at the beginning of each hash function. The SHA-256 algorithm performs 64 rounds of calculations for each block calculation. The final result obtained from the above process is: "80988f2ecd78337cc5ef25f95668884620c86324253c285fbc4e542682630410". An example of the implementation of hashing results can be seen in the image below.

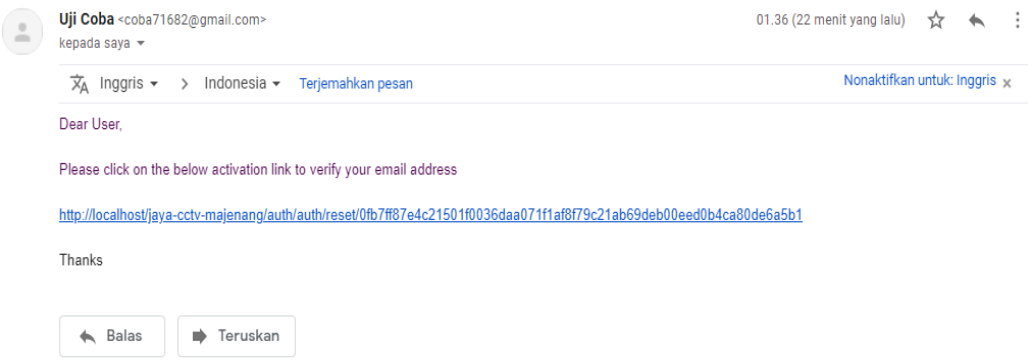


Figure. 7. Email password reset request with ciphertext customization sha-256

The final results obtained are then stored in a database and will be tested for security with software attacks and performance.

4) Testing

In this study, the hashing results will be tested using several methods as follows:

Endurance testing

The hashing resistance test is carried out to determine the resilience of the hashing results from brute force attacks using attacker software. The attacker software used by researchers to test the hashing results is Crackstation.

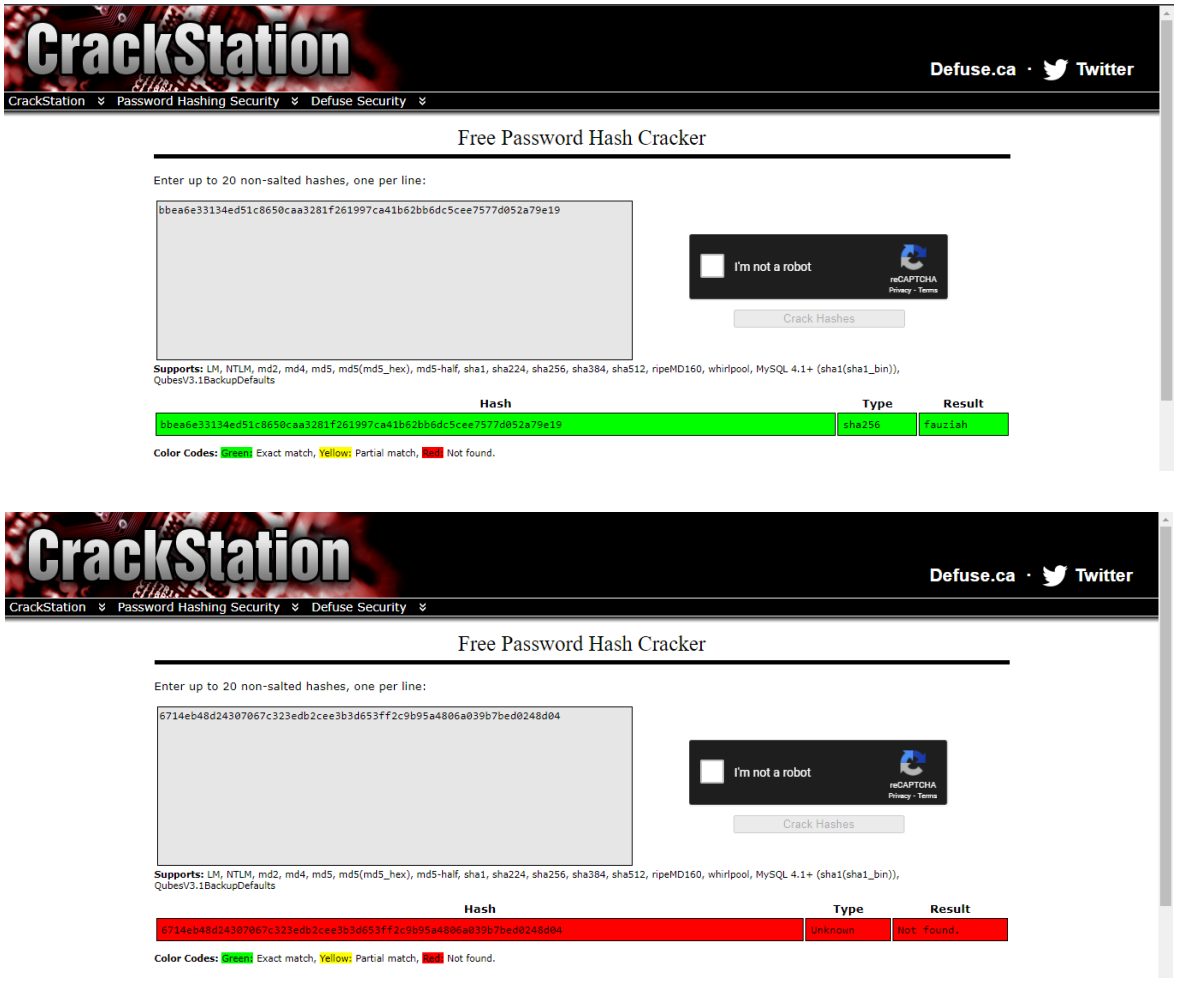


Figure. 8. Testing with ciphertext customization

Based on the results of the robustness test conducted using the crackstation media above, it can be concluded that the sha-256 hashing with ciphertext customization is more resistant to brute force attacks than without ciphertext customization.

Performance testing

In the performance test, the researcher conducted a test by measuring how long the website took hashing calculations using the performance test in the Google Chrome browser. Here are the results of testing the results of ciphertext customization sha-256 with a comparison of sha-256 without ciphertext customization.

ISSN 2723-6471

Table. 1. Table of performance test results

Test	Performance (ms)	
	SHA-256 (without ciphertext customization)	**SHA-256 (with ciphertext customization)**
1	1600	1700
2	1500	1400
3	1400	1900
4	1800	1600
5	1400	1400
6	1400	1400
7	1600	1900
8	1600	1800
9	1400	1400
10	1800	1600
11	1400	1800
12	1400	1400
13	1700	1700
14	1500	1400
15	1400	1400
16	1600	1800
17	1600	1600
18	1500	1800
19	1500	1400
20	1400	1400
Rata-rata	**1525**	**1590**

Table 1 shows that the average time required for the website to hash sha-256 with ciphertext customization is only slightly different from the time it takes to hash sha-256 without ciphertext customization, which is only 65(ms).

4. Conclusions and Suggestions

Based on the results of the research that has been done and described above, the researchers get the following conclusions: Based on the endurance test with "fauziah" and "qwerty" plaintexts, the results of sha-256 encryption by customizing the ciphertext cannot be decrypted at all by crackstation. Based on the endurance test with "fauziah" and "qwerty" plaintexts, the results of sha-256 encryption without ciphertext customization can be decrypted by crackstation. Customization of ciphertext on sha-256 can strengthen resistance to brute force attacks that can threaten data security. The average time required by the website to encrypt the 20 experiments that the researchers did was 1525 (ms) or 1.52 seconds without ciphertext customization. The average time required by the website to perform encryption with the customization described in chapter 4 is 1590 (ms) or 1.59 seconds. The performance of the time required to perform hashing does not differ much between whether or not the ciphertext customization is performed on the sha-256 algorithm which is only 65(ms) slower than the calculation with the sha-256 ciphertext customization.

In carrying out and completing this thesis, of course, it cannot be separated from various shortcomings and errors both in the design and manufacture of the system. Therefore, to correct deficiencies and improve other research that has the same theme as the researcher did, the researcher provides the following suggestions, ciphertext customization can be done more concisely for any hash algorithm by optimizing the functions and syntax that are made.

References

[1] J. Lee, Y. Lee, and C. Park, “The effect of consumer group breadth and depth on movie sales: the mediating effect of eWOM-to-viewing ratio,” Asia Pacific J. Mark. Logist., vol. ahead-of-print, no. ahead-of-print, Jan. 2021, doi: 10.1108/APJML-08-2020-0560.

[2] W.-C. Tsao, M.-T. Hsieh, and T. M. Y. Lin, “Intensifying online loyalty! The power of website quality and the perceived value of consumer/seller relationship,” Ind. Manag. Data Syst., vol. 116, no. 9, pp. 1987–2010, Jan. 2016, doi: 10.1108/IMDS-07-2015-0293.

[3] I. Khan and Z. Rahman, “E-tail brand experience’s influence on e-brand trust and e-brand loyalty,” Int. J. Retail Distrib. Manag., vol. 44, no. 6, pp. 588–606, Jan. 2016, doi: 10.1108/IJRDM-09-2015-0143.

[4] B. Berman and S. Thelen, “Planning and implementing an effective omnichannel marketing program,” Int. J. Retail Distrib. Manag., vol. 46, no. 7, pp. 598–614, Jan. 2018, doi: 10.1108/IJRDM-08-2016-0131.

[5] I. Khan, Z. Rahman, and M. Fatma, “The role of customer brand engagement and brand experience in online banking,” Int. J. Bank Mark., vol. 34, no. 7, pp. 1025–1041, Jan. 2016, doi: 10.1108/IJBM-07-2015-0110.

[6] N. Gudigantala, P. Bicen, and M. (Tae-in) Eom, “An examination of antecedents of conversion rates of e-commerce retailers,” Manag. Res. Rev., vol. 39, no. 1, pp. 82–114, Jan. 2016, doi: 10.1108/MRR-05-2014-0112.

[7] W.-Y. Wu, C.-L. Lee, C.-S. Fu, and H.-C. Wang, “How can online store layout design and atmosphere influence consumer shopping intention on a website?,” Int. J. Retail Distrib. Manag., vol. 42, no. 1, pp. 4–24, Jan. 2014, doi: 10.1108/IJRDM-01-2013-0035.

[8] I. Khan, Z. Rahman, and M. Fatma, “The concept of online corporate brand experience: an empirical assessment,” Mark. Intell. Plan., vol. 34, no. 5, pp. 711–730, Jan. 2016, doi: 10.1108/MIP-01-2016-0007.

[9] R. J. McQueen and Z. Yin, “Perceptions of entrepreneurs about their zero employee web enabled businesses,” J. Small Bus. Enterp. Dev., vol. 21, no. 1, pp. 26–48, Jan. 2014, doi: 10.1108/JSBED-10-2013-0144.

[10] G. Ö. Türker, “Website Designing and Its Impact on Tourism Destinations,” in The Emerald Handbook of ICT in Tourism and Hospitality, A. Hassan and A. Sharma, Eds. Emerald Publishing Limited, 2020, pp. 195–211.

[11] S. Yu, L. Hudders, and V. Cauberghe, “Targeting the luxury consumer,” J. Fash. Mark. Manag., vol. 21, no. 2, pp. 187–205, Jan. 2017, doi: 10.1108/JFMM-07-2016-0058.

[12] J. Murphy, P. Ho, and C. Chan, “Competitive Analyses for Marketing Electronic Wine Tourism,” Int. J. Wine Mark., vol. 17, no. 3, pp. 39–54, Jan. 2005, doi: 10.1108/eb008794.

[13] M. Ramezani Nia and S. Shokouhyar, "Analyzing the effects of visual aesthetic of Web pages on users' responses in online retailing using the VisAWI method," J. Res. Interact. Mark., vol. 14, no. 4, pp. 357–389, Jan. 2020, doi: 10.1108/JRIM-11-2018-0147.

[14] N. Soltani-Nejad, S. Z. Mirezati, and M. K. Saberi, "Predicting intention to share information on commercial websites based on personality traits," Bottom Line, vol. 33, no. 3, pp. 251–261, Jan. 2020, doi: 10.1108/BL-02-2020-0018.

[15] T. Ahn, Y. Ik Suh, J. K. Lee, and P. M. Pedersen, "Understanding purchasing intentions in secondary sports ticket websites," Int. J. Sport. Mark. Spons., vol. 16, no. 1, pp. 35–49, Jan. 2014, doi: 10.1108/IJSMS-16-01-2014-B004.

[16] H. Xu, K. Z. K. Zhang, and S. J. Zhao, "A dual systems model of online impulse buying," Ind. Manag. Data Syst., vol. 120, no. 5, pp. 845–861, Jan. 2020, doi: 10.1108/IMDS-04-2019-0214.

[17] R. Kozielski, G. Mazurek, A. Miotk, and A. Maciorowski, "E-Commerce and Social Media Indicators," in Mastering Market Analytics, R. Kozielski, Ed. Emerald Publishing Limited, 2017, pp. 313–406.

[18] S. Ferri, R. Fiorentino, A. Parmentola, and A. Sapio, "Patenting or not? The dilemma of academic spin-off founders," Bus. Process Manag. J., vol. 25, no. 1, pp. 84–103, Jan. 2019, doi: 10.1108/BPMJ-06-2017-0163.

[19] S. Bag, G. Srivastava, M. M. Al Bashir, S. Kumari, M. Giannakis, and A. H. Chowdhury, "Journey of customers in this digital era: Understanding the role of artificial intelligence technologies in user engagement and conversion," Benchmarking An Int. J., vol. ahead-of-print, no. ahead-of-print, Jan. 2021, doi: 10.1108/BIJ-07-2021-0415.

[20] M. A. Gardini, "A Study on the Online Sales Efficiency of Upscale and Luxury Hotels in Germany, Switzerland and Austria," in Advances in Hospitality and Leisure, vol. 3, J. S. Chen, Ed. Emerald Group Publishing Limited, 2007, pp. 173–192.

[21] T. J. Gerpott, S. E. Thomas, and A. P. Hoffmann, "Intangible asset disclosure in the telecommunications industry," J. Intellect. Cap., vol. 9, no. 1, pp. 37–61, Jan. 2008, doi: 10.1108/14691930810845795.

[22] H. Nayebpour and M. N. Bokaei, "Customers satisfaction by fuzzy synthetic evaluation and genetic algorithm (case study)," EuroMed J. Bus., vol. 14, no. 1, pp. 31–46, Jan. 2019, doi: 10.1108/EMJB-11-2017-0041.