Proceedings of the Sixth International Conference on Trends in Electronics and Informatics (ICOEI 2022)
IEEE Xplore Part Number: CFP22J32-ART; ISBN: 978-1-6654-8328-5

Design and Implementation of E-commerce Website using Automata Theory

M. M. V Sai Nikhil
Department of Computer Science and Engineering
Amrita School of Engineering,
Bengaluru India
sainikhilmmv@gmail.com

Aniket Sarrin
Department of Computer Science and Engineering
Amrita School of Engineering,
Bengaluru India ,
aniketsarrin@gmail.com

Ghanshyam S. Nair
Department of Computer Science and Engineering
Amrita School of Engineering,
Bengaluru India ,
mail2ghanshyam2001@gmail.com,

Supriya M.
Department of Computer Science and Engineering
Amrita School of Engineering,
Bengaluru India ,
, m_supriya@blr.amrita.edu

2022 6th International Conference on Trends in Electronics and Informatics (ICOEI) | 978-1-6654-8328-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICOEI53556.2022.9777191

***Abstract*— The current growth of technology has made a tremendous increase in the e-commerce industry. Internet-based e-commerce becomes the sole choice of a media to promote their industry as it offers most exciting business opportunities. This paper involves designing the backend structure of an e-commerce web-application. The method incorporated in this paper is a development of a prototype which is structured with flow chart tools, context diagram and data flow diagram. In the given structure, we take the user through the routes of creating a new user or login using automata theory concepts (push down automata to check if the user is already present or the password is wrong), choosing items via favorite categories, adding them to cart and placing orders. The design also includes the payment gateway structure as well as the return policy design (in which a pushdown automaton is used to work in reverse of the 'order' structure). A database could be incorporated which can maintain the input and output of the user and achieve uniqueness for the user. The challenges faced were brainstorming the simplest yet feasible and optimized structure of the backend of the architecture which is easy to understand for both the developers and users. We finally figured out that the best method is to have multiple databases for various tasks.**

Keywords*— *e-commerce web application, push down automata, database ,

I. INTRODUCTION

In the recent decade, there is a rapid change in the business environment. Hence, we must be more efficient, effective, and faster in listening and understanding the requirements of the customer to enable them to access the available products instantly. We can implement this by designing an E-commerce web application or an android app for shopping online [1]. Such platforms allow the customers to purchase a wide variety of fashions and goods either by paying instantly on the website for a seamless no-contact delivery which is more preferred by considering the spread of Covid-19 or we can also opt for cash on delivery so that the customer is satisfied as they pay after the item has been delivered. With the technology growing at this pace, most of the commercial transactions are taking place using websites not just in cities but in towns as well. Due to all these reasons, it makes shopping online a process which is familiar and makes the E-commerce an accepted paradigm.

To implement such an architecture a virtual store on the web will be required which will permit customers to search for their desired product or item and select from an exclusive catalog [2]. The purpose of this paper is to design and implement a web shopping to buy items like toys, gifts etc. This website has been developed using client/server techniques, relational database SQL, non-relational database MongoDB and using languages like HTML, CSS, JavaScript, Java, and PHP.

An automaton is a formal machine which can work like an acceptor or a transducer. Formally, an automaton is composed of various transitions and states. Every state in the structure represents as to where the automaton is in, while processing the input. In the study of abstract machines, which is a subject of discrete computer science, there exists well-known classification of machines and formal languages that could recognize each automaton. This method of classification is known as Chomsky hierarchy as can be seen in Fig. 1 [3]. This paper presents the e-commerce website creation as a set of transitions which has been implemented and tested using a java-based automata tool named JFLAP [4].

The main advantage of state machines above traditional bug testing is that they aid in the design process. Drawing out the state machine on paper makes it much easier to figure out all of the possible edge circumstances. This will ensure that your program has fewer problems and bugs. It also specifies which aspects of your object's internal state are exposed as external API. A finite-state machine is a tried-and-true mechanism for determining an application's logical structure. While fsm outperforms unit testing in terms of test-ability and debugging ease, having access to the state graph opens up various possibilities for the test-ability of the code. Because this technique enables for simple and unambiguous coding.

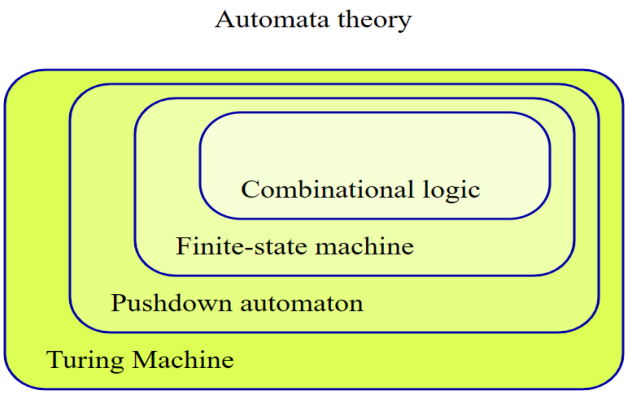

Fig. 1- Diagrammatic representation of the machines in Automata Theory

II. Literature Survey

Software systems have taken a tremendous change for the past two decades and at times they seem to be unreliable and difficult to construct [1] [5]. However, this rapid growth has always insisted to develop the systems using model driven techniques [6]. Automata based or state-based programming is one among the model driven approach that becomes essential for software modelling. A state machine is used to model the program which has a sequence of control instructions that helps in recognizing a set of inputs that is fed into the system [7] [8]. The dynamism in the system's resources and the number of users', expects the model to adapt to the environment at a faster rate which is addressed by the automata programming. The use of this method holds a higher complexity but makes the program's behavior and structure clear. In this paper, the series of tasks behind the website creation and purchase of orders is designed using the automata-based programming approach for an e-commerce web application of the Innovation and Entrepreneurship unit [9]. The proposed work is an extension of the automata model presented in [9].

The proposed application is modelled using finite state machines and push down automata. Though the use of automata concepts is found in many applications like activity representation, gaming, language processing etc. [10-15], this paper experiments it for an e-commerce application that is implemented using the tools identified.

The next section explains the methodology used and the implementation of the proposed model.

The use of spark jobs for the implementation of recommendation system and the utilization of multiple databases makes our model more efficient compared to existing models. We have observed that the users make more orders through the website as they get recommendations based on the browsing history of similar users and also based on the users searches.

III. Methodology and Implementation

A. System Architecture:

On visiting the website, the first page visible is the Welcome page which asks you to either login or register. If you are a new user, you must register using a unique email-id and password or register using google. Encrypted data is stored in an SQL Database. Login happens only if the username and password match the data in the database (SQL) else will redirect back to the welcome page. If you are an existing user but do not remember the password, you can opt for forgot password where the username will be asked. An email will be sent for confirmation. You can then change the password. On the server side the database updates the password. As default, a few items will be displayed on the home page based on some learning done by the model like orders made in your area or items that you prefer (based on searches and previous purchases). For finding specific items that you require there is a search module available to ease the task of finding required products for the user. During special festival or discount weeks we can have Cache memory where most of the people search for similar items like lights for Diwali or colors for Holi etc. to reduce the latency and also give a raw glance of favorable items available on the website. If the item is not in cache memory, then it will directly search through the Supply Chain Database which has the list of all items (NO SQL Database like Mongo DB). Items can be manually added and removed by the supplier depending upon the demand and stock. We can have an elastic search that has already performed a few SQL queries and stored the data so that we can save time for retrieving the data from the database (Example – electronics, watches, clothes etc.) [16].

All searches will be added to a distributed queue and spark jobs can be performed on them on a daily basis to recommend items better. Not everyone has the same taste and budget so we can filter the search depending upon our desire (Example – rate, date of manufacture, color etc.). After finding the required item, the site asks you to enter the pin code for delivery and adds it to cart only if the item is deliverable. The item can then be ordered through the buy now page. Processing the payment is the next step. If the payment is successful, then the item will be sent to procurement and the delivery can be tracked by the customer for which a SMS and email will be sent for the same. The user can check for updates via the app in the orders section as to how much more time it may take for the item to be delivered. The user can also track the present location of the item in the orders section. Spark jobs can be processed frequently to improve recommendations for the user for next order. The process is completed once the product is delivered to your doorstep. If people find any problems with our software or have any issues with the delivery or item, they can always approach the customer care through the home page. All the ordered items will be stored in a SQL Database. If the user wishes to return the product ordered, he can request the same from the home page and once the item is delivered, the refund will be processed. The item will be then sent to the supply chain database and the stock of the item will be increased by one if the reason for return is not related to quality of the item.

The user can browse into his profile from the home page where he can view his name, email id, saved cards, saved addresses etc. And finally, when he completes his work, he can logout or sign out safely. The pictorial representation of the above discussions is presented in Fig. 4.

Implementing the flow diagram as state transitions will make it easy to understand the scenario better. A sample implementation of the login page using push down automata is presented in Fig. 3 and its transition table is presented in Fig. 5.

So While we talk about building and E-commerce Website or to be more general, any web application. It can be broken down to states. Our basic purpose of using finite state machine is to get a clear development process as we can clearly see what can be done by the user on each state (in the form of actions and transitions). An ideal web application is one that can handle events in any order and respond in a meaningful way even if they happen in a different order than intended.

In a domain development point of view, these actions can be translated as function which would be triggered after a validation from the controller function (which cross-checks state changes). The action which has to be performed in response to an event depends mainly on the event and on the state of the machine, as well as determining and updating the machine's status.

The explanation of the automata presented in Fig. 3 has 13 states the description of which is presented below:

Q0 -> The starting point of the data while we enter an e-commerce platform.

Q1 -> transition takes place if the user wants to enter the store

Q2 -> Checking whether the username is unique or not.

Q3 -> If the user is already having an account in the website, he can login through this node.

Q4 -> If the user doesn't have an account in the website

He/she can register through this node.

Q5 -> After the user enters a username in the Login Phase.

Q6, Q8, Q11 are depictions of databases and how data is sent and received from it.

Q7 -> A node after Full authentication of the user or the home page of the e-commerce platform.

Q9 -> Checking the password if it follows the described criterion.

Q10 -> Through this node the user provides his information for the register.

Q13 -> Cross checks the login input with its corresponding username and authenticates the user.

Other nodes are user's input transition nodes.

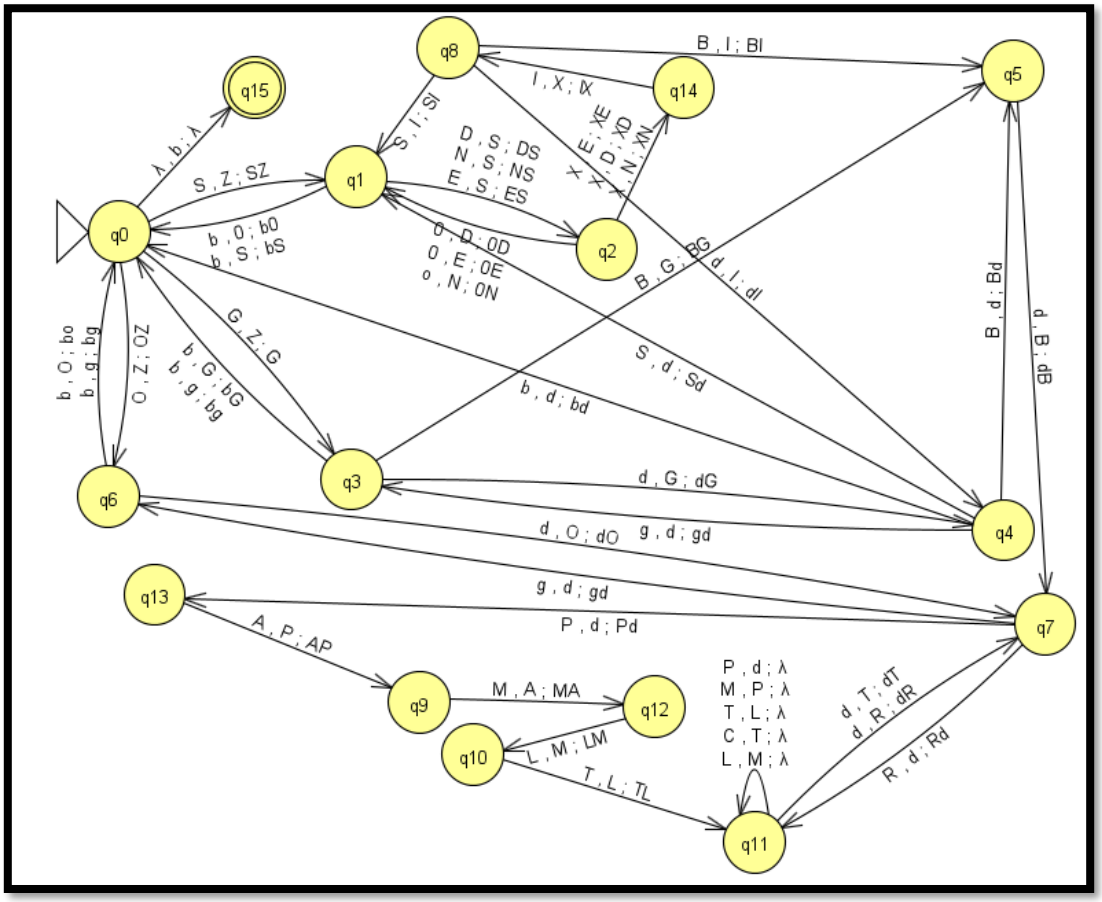

Fig. 2 - Jflap Implementation of the whole data flow in an E-commerce platform.

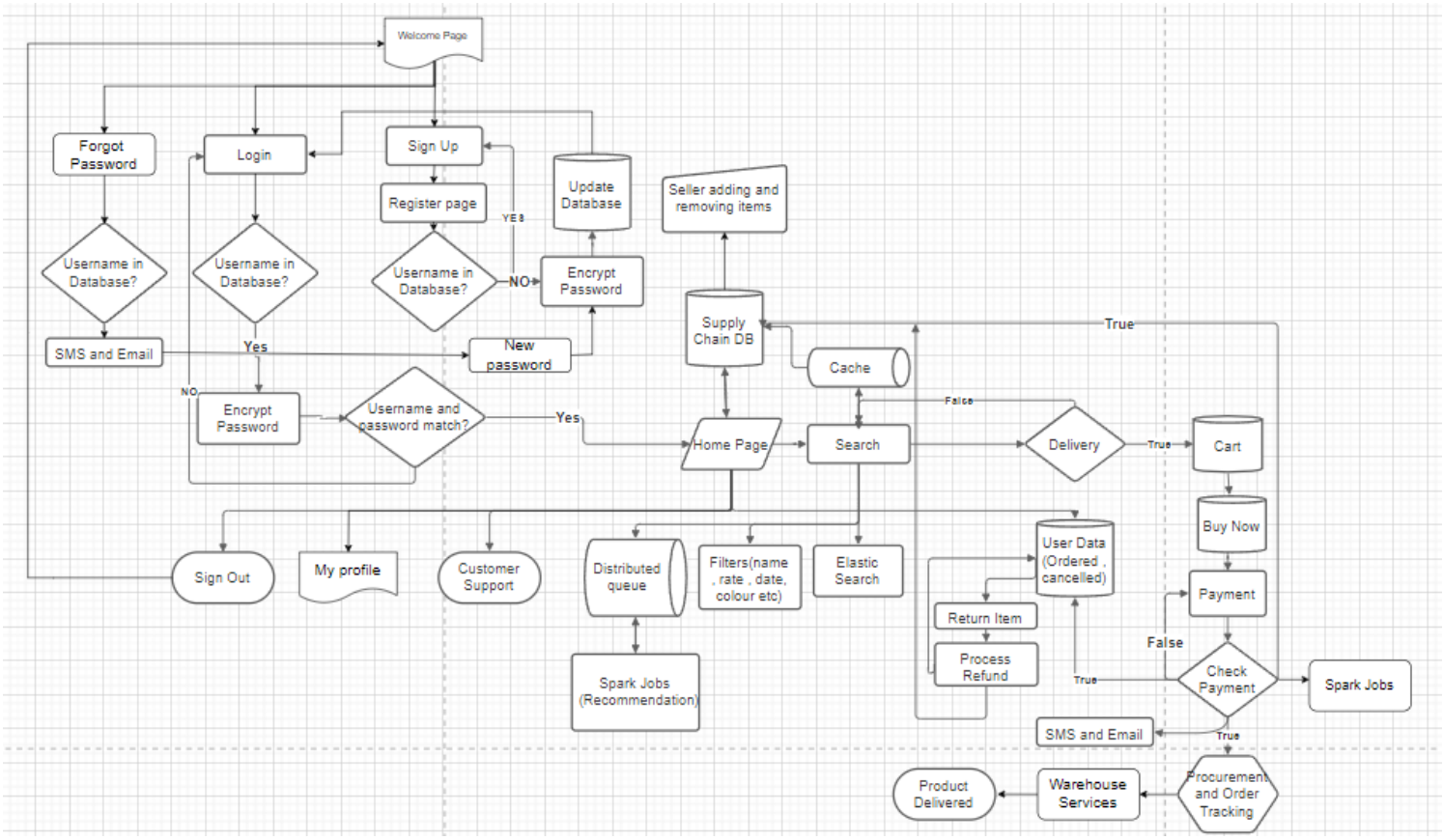

Fig. 3- System Architecture of proposed E-Commerce Platform (Transitions from the home page -- the payment gateway)

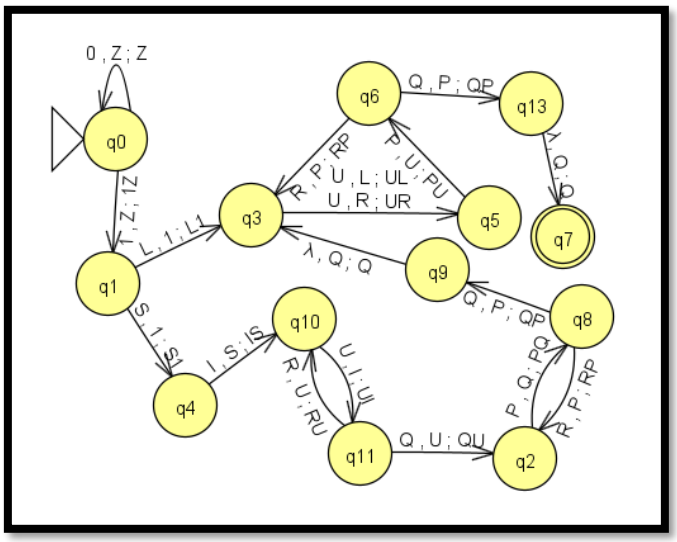

Fig. 4 - Jflap Implementation of the login page of the E-commerce Platform

Fig. 2 describes the Jflap implementation of the proposed e-commerce application the details of which are explained below. The corresponding transition table is presented in Fig. 6.

Q0 -> the starting page or home page

Q1 -> is a depiction of the search engine present in the website.it takes in three types of searches from the home page.

Q2, Q7, Q4 are depiction of database in this page.

Q5 -> To buy the items taken in by the user.

Q6 -> When the user wants to check on his orders.

Q6, Q8, Q11 are depictions of databases and how data is sent and received from it.

Q8-> The user can choose to either buy the item at that instant or add it to the cart to (Q4).

Q10 -> Provides the user access to track the order

INPUT NODE	USER INPUT ACTION	OUTPUT NODE
Q0	Z-(User enters the site)	Q1
Q1	L-(USER LOGIN)	Q3
Q1	S-(USER SINGUP)	Q4
Q6	R-(Retry on incorrect Credentials For Login)	Q3
Q3	U-(Username for login)	Q5
Q5	P-(User Entering password)	Q6
Q6	Q-(Correct Credentials and Approved Authorization)	Q13
Q13	Transition into Home page	Q7
Q4	I-(Append User Information on Registration)	Q10
Q10	U-(Check for Existence of Username in DB)	Q11
Q11	R-(Re-enter Username because of redundancy)	Q10
Q11	Q-(Approved Username)	Q2
Q2	P-(Password Guideline Check)	Q8
Q8	R-(Re-enter incase of violating guideline)	Q2
Q8	Q-(Approved User for registration)	Q9
Q9	Re-directs User to Login Page	Q3

Fig. 5 - Transition Table for Fig 3

INPUT NODE	USER INPUT ACTION	OUTPUT NODE
Q0	NO User action /tends to leave the page	Q15
Q0	S-Serch For item	Q1
Q0	G-Go to Cart	Q3
Q0	O- User Orders	Q6
Q1	N-item name based Search,D-User cache based search, E-Elastic Search	Q2
Q2	X-Item of relevance exists	Q1
Q2	0-No item of relevance, So tends back to Search	Q14
Q14	I-Chose an item from the collections of choices	Q8
Q8	S-Go Back to searching	Q1
Q8	B-Buy item at that Instant	Q5
Q8	d-Add item to the Cart dataBase	Q4
Q5	d-Send Data to user's order Database	Q7
Q4	b-Back to Home Page	Q0
Q4	S-Back to Searching	Q1
Q4	g-Get data from cart Database on display	Q3
Q4	B-Buy That item from Cart	Q5
Q3	d-Get data cart Data	Q4
Q3	b-Back to home page	Q0
Q3	B-Buy the Whole Cart	Q5
Q7	g-Get order Data	Q6
Q7	R-Retract Order	Q11
Q7	P-transitions into payment gateway	Q13
Q6	d- Order Data to Database	Q7
Q6	b-back to Home Page	Q0
Q11	Retraction on payment ,Message ,tracking and order.	Q11
Q11	d-Updating Order data	Q7
Q13	A-Approval of Payment	Q9
Q9	User receives Mail, messages .Evidence on purchase from Application	Q12
Q12	Order Detals Sent to vendor or Warehouse	Q10
Q10	Tracking details provided to user	Q11

Fig. 6 – Transition table for Fig. 2

Q11-> Discards the previous access and protocols if the user decides to either return or cancel the item.

Q12 -> Providing the user with evident receipt of the order.

Q13 -> Validating the payment made by the user.

Q14 -> Through this node if Search results exist. After this phase user can pick and choose any item from his search results.

Q15 -> Transition in case the user discards shopping in this website.

B. Search Engine Implementation:

For efficient and quick use of the website we have implemented a search engine where the user can enter a keyword and appropriate results would be displayed.

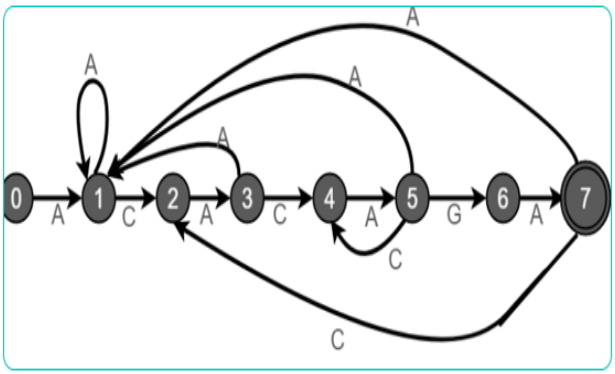

Fig. 7 - Finite State machine network for keyword searching

We implemented a finite state machine for keyword searching as shown in Fig. 7. This is coded to find the number of similar products available based on the search.

Basically, we take in all the items present in the database and make a single string out of it containing multiple(all) items.

We have implemented a pattern finding algorithm using Finite Automata (FA). Using this algorithm, we first try and preprocess the pattern. We then try to build a 2D array that represents a Finite Automata.

In the process of Search, our goal is to begin from the initial state of the automata and the beginning character of the string. In each step, we will check for the following character in the string, similarly look for the succeeding state in the FA that is constructed and move to a new state.

If we reach the terminating state, then that pattern is found in the text. The time complexity of this process is very compact(O(n)).

If we have a character 'x' and a state 'y', we will be able to reach the successive state by taking the string “pat [0....y-1]x”.[17]

So it is the concatenation of the pattern characters pat [0], pat [1] … pat[y-1] and the character x.

The basic goal is to get the length of the longest prefix of the given pattern so that the prefix is also a suffix of the pattern “pat [0..y-1]x”.

A conjoint string can emphasize on better optimization of the search engine as it would provide every output of the user entered keyword.

C. Delivery Route Optimization:

Finally, for the delivery of the package/order made by the user an optimized route has to be chosen by the delivery agency. The process of route optimization can be defined by which the delivery or logistics stakeholders can consistently improve a given route that connects disparate locations.

In simpler terms it just means discovering the most efficient route to move from point A to point B and optimize this route by considering factors which can be critical like traffic congestions and number of stops.
We can consider the resultant orders as an Adjacent graph and consider every node as a delivery location. The resultant shortest path we receive from it would be used to implement the FSM in order to track and get the priority of order delivery fixed [18].
When multiple orders go to the same destination or the destination of 1 product is a mid-way path to the other, then both the products are picked up from the departure city at the same town and the final delivery can happen with multiple stops while optimizing the time and cost of delivery.

IV Results and Discussion

Input	Result
1LUPR	Reject
1LUPRUPQ	Accept

Fig. 8 – Test Case output for the Login Page FSM.

Input	Result
SNXIdBdPAMLTdgb	Accept
GdBdP	Reject

Fig. 9 – Test Case output for the E-commerce website traversal.

In Fig-8, We can observe that in the first case the user has been rejected authentication on invalid credentials and successfully authenticated successfully on retry. This manner of maintaining authentication based on states is predominantly efficient and easier way of reading up logs and data. Same as in Fig-9, We can observe that in the first case the user has bought an item from searching it and has left the page . that would be an optimal case for any e-commerce platform. FSM provides an alternate and creative manner of collecting data and logs of different users regarding their data flow and give opportunities for developing more comprehensive solutions to where the platform is lacking.

V CONCLUSION

This paper presents an automata-based approach to design the e-commerce website. The software has been modelled using various formal language automata machines like finite automata and push down automata. The route optimization for the delivery of products has been designed and implemented to optimize the path. The pattern matching algorithm using finite automata has been devised to seek the frequently searched products. In conclusion, a website has been constructed to reduce the latency of the complete process which is made user friendly as well using the web technology tools. This methodology is one of the experimentations of the use of finite state machines in web development application.

VI REFERENCES

[1] Selic, D. and Leo, M. 2003. Using Models in Real-time Software Design, IEEE Control Systems Mag., 23(3):31－42.

[2] Binz Astrachan C, & Botero I C 2018 “We are a family firm” An exploration of the motives for communicating the family business brand. Journal of Family Business Management 8(1) pp. 2-21.

[3] Clemes M D, Gan C, & Zhang J 2014 An empirical analysis of online shopping adoption in Beijing, China. Journal of Retailing and Consumer Services 21(3) pp. 364-375.

[4] Susan Rodger and Thomas Finley, JFLAP - An Interactive Formal Languages and Automata Package, ISBN 0763738344, Jones and Bartlett, 2006.

[5] Shalyto, A.A. 1991. Programmatic implementation of control automata, Marine industry: "Automation and remote control", issue 13, pp. 41, 42.

[6] Spalter, A. M., and Dam, A. V. 2003. Problems with using components in educational software. Computers & Graphics 27, pp. 329-337.

[7] Tukkel, N.I., and Shalyto, A.A. 2001. State-based programming”, PC World., vol. 8, pp.116-121 and vol. 9, pp.132-138.

[8] Gomaa, H. and Hussein, M. 2007. Model-Based Software Design and Adaptation. Proc. ACM/IEEE ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, Minneapolis.

[9] G. Beronius and S. Andrén, ‘E-Commerce Web design: The importance of a first impression’, Dissertation, 2017.

[10] Swathy Joseph and K.P. Jeevitha, “An Automata Based Approach for the Prevention of NoSQL Injections”, In security in Computing and Communications, Springer – Communications in Computer and Information Science (link is external), 2015, pp 538-546.

[11] Ramesh, G., Menen, A. “Automated dynamic approach for detecting ransomware using finite-state machine”, Decision Support Systems. 2020.

[12] K. P. Jevitha, Swaminathan J., Jayaraman, B., and M, S., “Finite-state model extraction and visualization from Java program execution”, Software: Practice and Experience, vol. 51, pp. 409-437, 2021.

[13] Kanjirangat, Vani & Gupta, Deepa. “Unmasking text plagiarism using syntactic-semantic based natural language processing techniques: Comparisons, analysis and challenges”, 2018.

[14] Tamizharasan, M., Shahana, R.S., Subathra, P. “Topic modeling-based approach for word prediction using automata”, Journal of Critical Reviews, PP 744-749, 2020.

[15] Kavya, J., & Geetha, M. "An FSM based methodology for interleaved and concurrent activity recognition". Paper presented at the 2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016, 994-999.

[16] Ali R & Beg M S 2017 Introduction. In Applications of Soft Computing for the Web Springer, Singapore pp. 1-7.

[17] Ma, Zongmin, Web-Based Intelligent E-Learning Systems.

[18] Technologies and Applications, 2005, Information Science Publishing M. Li, J. Zhang, and W. Wang, "Task selection and scheduling for food delivery: a game-theoretic approach," in *Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, Abu Dhabi, UAE, February 2018.