

A QuaggaJS Computer Vision Authentication Scheme for Electronic Payments

Koffka Khan

Department of Computing and Information Technology
The University of the West Indies,
Trinidad and Tobago, W.I

Sidone Reyes

Department of Computing and Information Technology
The University of the West Indies,
Trinidad and Tobago, W.I

Adesh Maraj

Department of Computing and Information Technology
The University of the West Indies,
Trinidad and Tobago, W.I

Andre Persad

Department of Computing and Information Technology
The University of the West Indies,
Trinidad and Tobago, W.I

Abstract— Trinidad and Tobago is home to one of the campuses of the leading University within the Caribbean and a place where innovation thrives. This is where ideas for new or emerging technologies are fostered and carried out, hence our group setting out to create and set the University of the West Indies (UWI) as the first Campus within Trinidad to become cashless. What does this mean? Well this means that we aim to move the campus from being dependent on carrying around physical cash to using the main identifier the UWI ID card as a centralized debit card, getting persons on campus to stop carrying around physical cash and using this as their main payment method for all things UWI related. We implemented and tested two QuaggaJS and Barcode to PC computer visions authentication schemes to perform the barcode scanning task. It was found that QuaggaJS outperformed Barcode to PC by 23%.

Keywords—UWI; cashless; debit card; barcode; computer vision; authentication

I. INTRODUCTION

A problem we face on University Campuses across the world is financial security [23], in the sense that having to carry around physical money, as well as paying for various items without a debit or credit card system in place as per the situation with most small vendors that students frequent, is unsafe and can leave individuals open to various security risks.

Therefore, the objective of our project “UwiPay” is to establish a secure, cheap and operational cashless means for students and vendors to carry out transactions throughout the campus [4]. We decided to do this by building a new local credit system from scratch that would feature the use of the University of the West Indies’ student identification card functioning as a debit card for each student to use. For this project, we would have had to figure out how to securely set up this system by finding the right hardware and software for the best development of both the frontend and backend aspects of the system, which would include setting up the databases and query functions that would allow for students to carry out the basic functions. Our system would also need to have a user-friendly GUI that persons can use for various functions such as viewing the system or updating their balance through an online method.

Tasks performed during the duration of our time working on the project were the design of a functioning web application built to perform the functionalities of the system and the queries of the databases [26]. We also performed database designs to hold and store information on various elements that could be scaled up when fully implementing the system.

UWIPay require various resources to be functional, some of these resources can be broken up into physical resources and technological resources, such as a smartphone to run the barcode application, a physical laptop or desktop to use the web application, databases to store information and carry out queries, a local server to host the application and others [20]. Using these resources there were various risks that could have risen due to the nature of the project, such as incompatible software or libraries and that would put a halt to the project or change the direction of the project.

Major Objectives for our project would have been at 3 main points these being:

- The development of the Web Application (Front End)
- The development of the Functionalities for the Web Application (Back End)
- Finalizing the barcode application, we used and then getting the scanner to send the barcode scanned to the web application.

The various stakeholders that would be involved in “UwiPay” would be, the users such as University students, lecturers, UWI based vendors and the University itself as the system will be built for the University itself. Other stakeholders would be the bank associated with the University which would be used to complete transactions, this bank being Republic Bank. “UwiPay” would be a full debit card system using the barcode technology on student identification cards as a means to carry out various monetary transactions on the University of the West Indies campus to simulate a cashless campus.

The cost modelling technique used was the COCOMO 2 Model [7]. As the COCOMO 2 Model is an algorithmic cost modelling it generates repeatable estimations meaning different values can be entered into the software at different times and the estimation can be generated. The expert

judgement model does not offer this as the expert judgement model can only give what they have experienced which may not always be accurate. The information in the COCOMO 2 model can easily be altered and modified, whereas, with the expert judgement model, the expert is a part of the team and can give biased estimations if their input on the values are not considered. The COCOMO 2 Model's equations are based on the research and historical data and use inputs such as source lines of code, the number of functions to perform and the other cost drivers such as language, design, methodology, skill-levels, risk assessments and more to generate the cost estimation, this means that the cost estimation, this means that the cost estimation was made using quantifiable information. Using the expert judgement model that the cost model would have meant that the cost estimation would have been based on the expert's experience, which is highly risky as the expert on the team may not have had the experience for this particular project, therefore the cost estimation would have been inaccurate as the information would have been quantifiable.

The mathematical equation used for the basic COCOMO Model [7] was $MAN-MONTHS = K1 * (Thousands\ of\ Delivered\ Source\ Instructions)^{K2}$, where K1 and K2 are two parameters dependent on the application and development environment. Some of the factors that helped with the values for the estimation for the basic COCOMO 2 model was taking into consideration the qualifications and experience of the development team members and the software development team members. The COCOMO 2 model assumes that factors such as the required reliability of the team, the size of the database, memory and execution time, the analyst and the programmer capabilities, experience of the team in the application area, experience of the team with the programming language and computer, and the usage of tools and software engineering practices are defined and stable. For these reasons, the COCOMO 2 model was the best choice made to generate cost and budget estimations for this proposed project.

This paper consists of five sections. In Section II we provide details for the proposed UwiPay system. Section II discusses how computer vision is used in machine authentication. Section IV gives the methodology used to develop UwiPay. Section V gives the results and in Section VI we entail the conclusion.

II. UWIPAY SYSTEM FEATURES

A. Functional Requirements

When certain criteria are fulfilled, a functional requirement will specify a certain behavior of the system's function. UwiPay functional requirements are:

- Barcode Scanners must be installed at each outlet on campus.
- A UWI Credit System account must be created for the user in order for the user to utilize the service.
- The bursary's financial system shall access the user's account, deposit the credit to the respective account in real time and a receipt is given to the user.
- The online top up system via the university's website shall deposit the credit to the student/staff's account within 2-3 hours and a confirmation email shall be sent to the user.
- The transaction system shall access the user's account upon scanning of the ID card using the barcode and

send a verification message confirming if the transaction is successful or unsuccessful. The transaction shall then either be cancelled or completed.

- When a user is no longer part of the University, the U.W.I system will issue a reimbursement upon request to the user and then their account is deleted from the database. A request must be made within a space of 3 months upon departure of the university.

- If a user's U.W.I ID is reported stolen or lost, the System shall freeze the user's account until a replacement is given or the card is retrieved.

- Purchases should be allowed as long as the user has sufficient credit.

- When an error occurs, the system shall alert the user to abort the transaction or re-input the data necessary.

B. Non-Functional Requirements

A non-functional requirement specifies how a system should behave and what its functional limitations are. UwiPay non-functional requirements are:

- The system must be secure since sensitive and personal data is being accessed.

- The system response-time must be quick. The deposit and transaction process must be done in a timely manner.

- The system must be reliable. The transaction process should not fail at any time.

- The system should be available from 7am to 10pm Monday to Saturday. The System should be available to the users for credit deposits at the bursary during school hours and online at any time as long as updates aren't being done on the system.

- Updates on the system must be done after working hours if needed. The fundamental programming must not be affected by these.

C. User Stories

A user narrative or story is an informal, natural language description of software system characteristics used in software development and product management. They're written from the perspective of a system's end user or user, and they're kept on index cards, Post-it notes, or in project management software.

We now give two user stories (A and B) for UwiPay in the following description.

User Story A:

- Billy is a Final Year student that is currently enrolled in a Computer Science degree, and is currently on the search for something to eat within his general area, sadly he has no physical cash on him just his new UWIPay ID card and his debit card and unfortunately all the vendors nearby do not use card transactions only cash but luckily for Billy these vendors recently signed up for the UwiPay system as well, so Billy can pay for his meal/ snack with just scanning his ID card and entering his security code.

User Story B:

- It's current day is Sunday and Joe is a student who recently registered for UwiPay but has not updated his balance as yet. He knows that the bursary is closed on a Sunday and that he would be unable to update his balance or on the following day, when he suddenly remembers that he can

update his account online using the UwiPay website. He then proceeds to the website, logs in using his credentials given to him upon signing up, after this he navigates to online payments and enters his banking information and the amount he would like to add to his account and within a few hours his account is then updated along with an updated transaction alert on his account.

D. User Case Diagram

The needs of a system, including internal and external factors, are gathered via use case diagrams. The majority of these criteria are design-related. As a result, use cases are generated and actors are identified when a system is studied to gather its functionality. Figure 1 below gives the use case diagram for the UwiPay system.



Fig. 1. UwiPay Use Case Diagram.

E. Entity Relationship Diagram

An entity relationship diagram (ERD), often called an entity relationship model, is a graphical depiction of relationships between people, things, locations, concepts, and events in an information technology (IT) system, see Figure 2.

F. System Diagram

A system diagram is a visual representation of a system's components and interactions. It may capture all of the relevant information about a system's design with the help of supporting documentation. Figure 3 and 4 below shows a level 0 and 1 system diagrams for UwiPay.

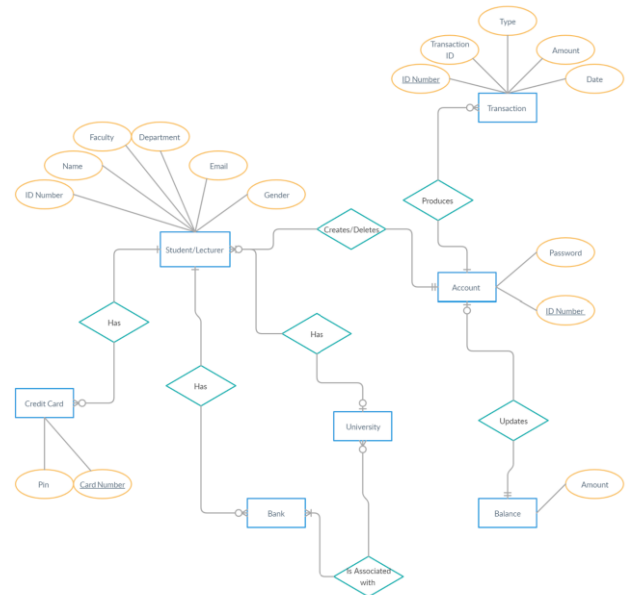


Fig. 2. UwiPay Entity Relationship Diagram (ERD).

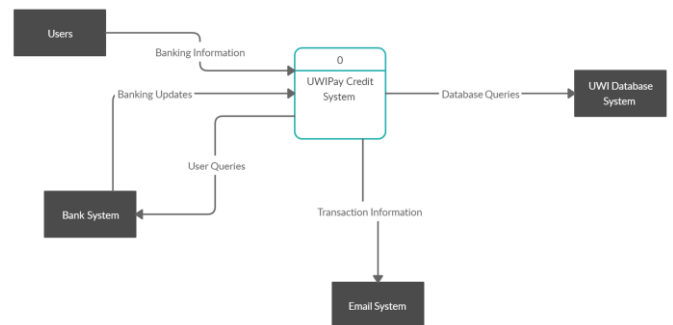


Fig. 3. UwiPay Level 0 System Diagram.

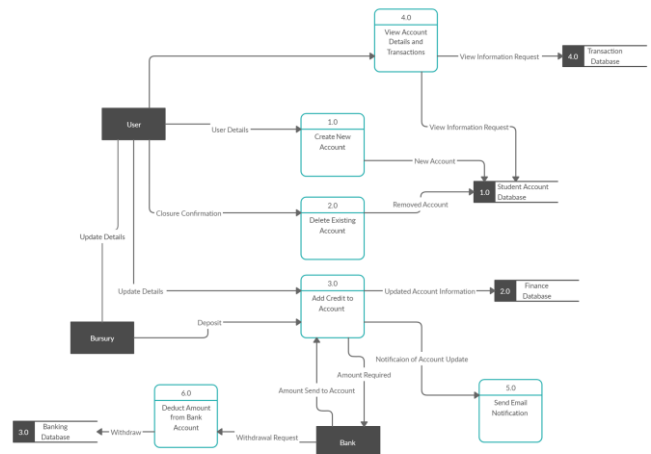


Fig. 4. UwiPay Level 1 System Diagram.

III. COMPUTER VISION AND MACHINE AUTHENTICATION

Computer vision [19], [28], [5] is an interdisciplinary scientific topic that studies how computers can perceive digital pictures or movies at a high level. From an engineering standpoint, it aims to comprehend and automate operations that the human visual system is capable of. On the other hand, machine authentication [12], [29] is the

verification of a digital certificate or digital credentials to authorize automated human-to-machine (H2M) or machine-to-machine (M2M) communication. We explore both concepts in this paper to develop the cashless system.

A barcode is a picture (see Figure 5) made up of parallel black and white bars that can be read by a barcode scanner. Products are labeled with barcodes so that they may be identified immediately. Barcodes are commonly used in retail businesses as part of the purchase process, in warehouses to track and manage inventory, and on invoices to assist with accounting, among other things.



Fig. 5. Barcode scan

The barcode [18] is input into the computer as an image taken by a camera. The eliciting of information from the input data comprises the computer vision task. The goal of the task is to identify a given pattern from the picture or image of the barcode. There are many algorithms that can do this identification task including neural networks especially deep learning. [5]

Once the pattern is taken from the barcode image it has to be compared to an existing pattern. If it matches, then you are authenticated to the system and allowed access. You must first scan a barcode on an ID card using a barcode reader before using it. The data is fed into a computer system, which retrieves the data associated with that code. The barcode itself does not hold much information; instead, it simply refers to data saved in database software.

Your neighborhood UWI shop, for example, most likely has a reward on your student membership card. Every time you make a purchase, the barcode on your rewards card is scanned, allowing you access to exclusive member-only UWI bonuses or UWI specials. However, the barcode does not carry all of your membership information; instead, it only contains your UWI student membership ID number due to a lack of storage capacity. When the barcode is scanned, However, because the barcode is linked to your student member number, the program will call up your account when you scan it.

QuaggaJS [13] is a scanner with real-time localization techniques for finding barcodes. The pipeline can be divided into the following three steps:

1. Reading the image and converting it into a binary representation
2. Determining the location and rotation of the barcode
3. Decoding the barcode based on the type EAN, Code128

The first phase requires a webcam stream or an image file to be used as the source, which is then converted to grayscale and saved in a 1D array. The picture data is then sent on to the locator, which is in charge of looking for a barcode-like

pattern in the image. Finally, if a pattern is discovered, the decoder attempts to read the barcode and returns the outcome.

PC to Barcode [16] turns your smartphone into a barcode scanner by connecting it to your computer. Access the compatible model's hardware and firmware settings and program them to identify and analyze barcode data. The findings of a scanning session can be saved as a CSV file.

IV. METHODOLOGY

For our project we decided on using and implementing a software architecture design to complete the goal we set out to accomplish. We decided to branch off into using a web application architecture, using both client-side code and server-side code that was done mainly in PHP, and other web application languages. As far as discussions went for alternative designs, we did not discuss any, we knew that this was the design we wanted to use from the beginning of the project.

The system interface breakdown is as follows, see Figure 6:

- Phone Scanner
- Barcode to PC Server
- Backend Server
- Web Application
- Student(User) Database
- Financial Database
- Transaction Database

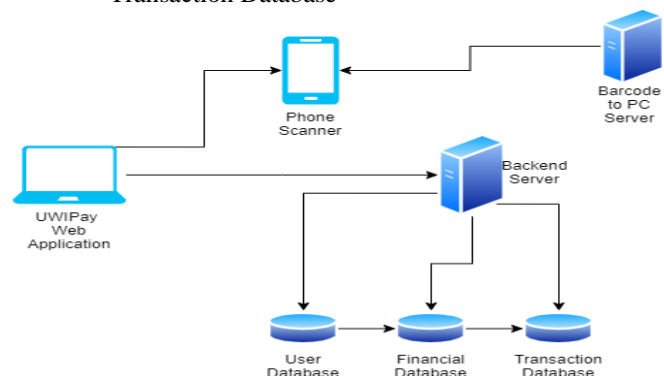


Fig. 6. UwiPay Components/Subsystems.

The following components for UwiPay are used:

- Phone Scanner: This Component will be the main application to retrieve the barcode from an identification card. It is expected that UWI vendors use this application to scan ID cards to carry out transactions across campus.

- Web Application
- Barcode to PC Server
- Backend Server
- Student(User) Database
- Financial Database
- Transaction Database
- User Interface Design

The user interface is made up of a main web page that serves as a homepage for users where they can carry out various functions. These functions are all nested within multiple web pages that contain forms that take user input and carry out queries to the various databases.

Technologies/libraries and languages used were Laravel [22], XAMPP [10], PHP [11], JavaScript [25], Bootstrap [6], jQuery [2] [15], CSS [3], Barcode To PC (Open Source Server and Application for barcode Scanning), HTML [17], IcoMoon [9] (Font and Icons Set), Ionicons [9] (Open Source Icon Set) and Owl Carousel (Touch enabled jQuery plugin). [30]

For our project we had various technical constraints throughout our timeline such as database constraints where we had to switch from MySQL to a more dynamic database platform that would work with our web based application, such as Laravel (Laravel is a free, open-source PHP web framework. It is meant for the building of online applications that follow the model-view-controller architectural pattern [1]) and XAMPP (XAMPP is a free and open-source cross-platform web server solution stack bundle built by Apache Friends, which consists mostly of the Apache HTTP Server [21], MariaDB database [27], and interpreters for PHP and Perl scripts [24]). Another constraint we faced was the use of the barcode technology to get the main functionality of the system (retrieving the identification number from the UWI ID card). We used a laptop webcam and the QuaggaJS library to scan the barcode for input. Additionally, we wanted a portable program to conduct the barcode scanning, and we found Barcode to PC, a server and client side barcode scanning tool to do this task. We later compared the QuaggaJS output to the Barcode to PC technique.

V. RESULTS

Our testing was done locally on a desktop PC. This includes testing of the backend and barcode scanner. The main testing throughout the project timeline would have been for the queries for the databases and the forms to ensure that they were updating the databases when an event occurred. These functions were then tested when the barcode features were included, and the system have been tested with live input.

During our project we decided that the best approach to the testing would be following a proactive approach, in which we implement a methodical approach in which we test based on what did not work at each stage of implementing the project, and then improving and fixing accordingly.

For the overall system most features were implemented and tested these being the Account Creation, Delete Account, Login and Adding Credit Features, Transaction Summary and the View Account Feature. For UWIPay the features that we were additionally implemented and tested were the Balance Per Year, Mini Statement and Balance per month features.

The Screenshots of our user interface can be found on the following Figures 7-13.

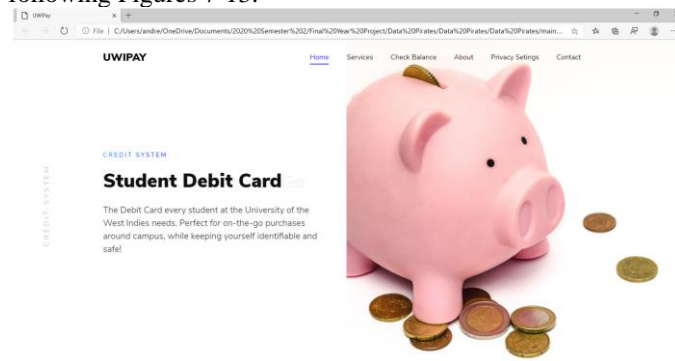


Fig. 7. UWIPay Homepage.

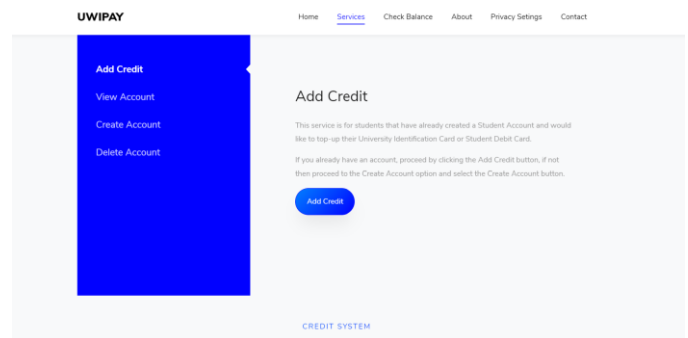


Fig. 8. UwiPay Add Credit webpage.

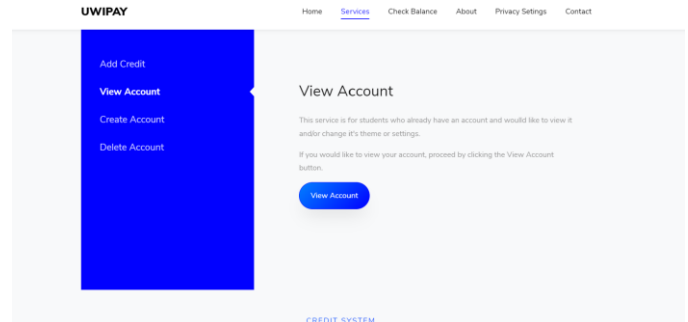


Fig. 9. UwiPay View Account webpage.



Fig. 10. UwiPay View Account Details webpage.

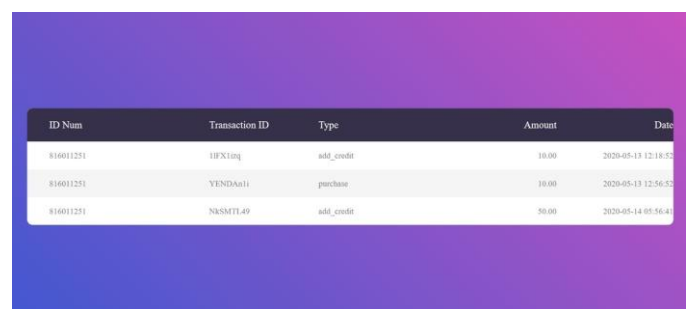


Fig. 11. UwiPay Transaction View webpage.

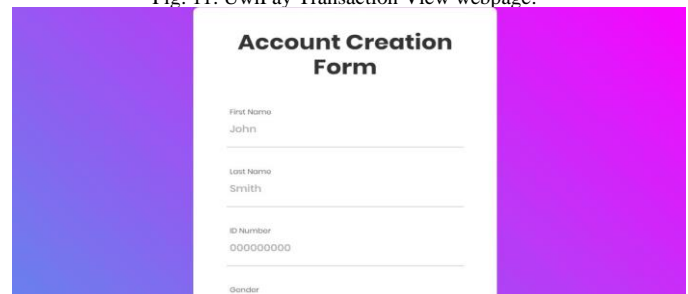


Fig. 12. UwiPay Create Account webpage.

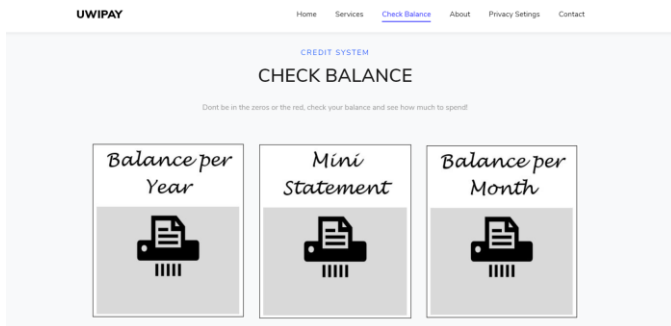


Fig. 13. UwiPay Check Balance webpage.

From our experiments QuaggaJS had a 96% accuracy in identifying real-time barcodes while Barcode to PC had a 73% accuracy measure. In the future we plan to incorporate different scanning algorithms to test their performance.

VI. CONCLUSION

Our product UWIPay and the services it offers which we tailored to be suited for the University of the West Indies is an innovative method to help the university maintain a safer environment for their students and lecturers as they carry out their daily activities. UWIPay also offers UWI the opportunity to take a step forward to create the first cashless campus in Trinidad and Tobago. Our system's main features are: (1) The ability to convert a student's identification card into a debit card. (2) The ability to carry out "cashless" monetary transactions across the campus. (3) The ability to scale to provide additional services. (4) The ability to carry out functions such as adding credit, and deducting credit.

The emergence of cashless campuses came up within recent years in Countries such as America and China where schools such as National University of Singapore (NUS) have either completely transferred or partly transferred over to a cashless system using one card to fully many purposes such as making purchases both on and off campus.

Our product looks to continue on this trend and help UWI become the first University in Trinidad to implement these features starting off the innovative competition to create cashless and safe campuses within Trinidad. Due to how similar the concept of a "Cashless campus" is, our system tends to only differ slightly from what is "Standard" so we are taking advantage of the simpleness, overall effectiveness and cost effectiveness of Barcode Technology as compared to using RFID ID technology within smart cards. Our research is looking to provide a service for the University of the West Indies that looks at the best interest of the students, lecturers, UWI Staff, administration and Republic Bank.

REFERENCES

- [1] Adam, S.I. and Andolo, S., 2019, August. A new PHP web application development framework based on MVC architectural pattern and ajax technology. In 2019 1st International Conference on Cybernetics and Intelligent System (ICORIS) (Vol. 1, pp. 45-50). IEEE.
- [2] Chaffer, J., 2013. Learning jQuery. Packt Publishing Ltd.
- [3] Geneves, P., Layaïda, N. and Quint, V., 2012, April. On the analysis of cascading style sheets. In Proceedings of the 21st international conference on World Wide Web (pp. 809-818).
- [4] Harasim, J., 2016. Europe: the shift from cash to non-cash transactions. In Transforming payment systems in Europe (pp. 28-69). Palgrave Macmillan, London.
- [5] Hassaballah, Mahmoud, and Ali Ismail Awad, eds. Deep learning in computer vision: principles and applications. CRC Press, 2020.
- [6] Hong, P., 2018. Practical web design: Learn the fundamentals of web design with HTML5, CSS3, bootstrap, jQuery, and vue. js. Packt Publishing Ltd.
- [7] Kaur, I., Narula, G.S., Wason, R., Jain, V. and Baliyan, A., 2018. Neuro fuzzy—COCOMO II model for software cost estimation. International Journal of Information Technology, 10(2), pp.181-187.
- [8] Khan, K. and Sahai, A., 2011. A levy-flight neuro-biosonar algorithm for improving the design of eCommerce systems. Journal of Artificial Intelligence, 4(4), pp.220-232.
- [9] Kotaru, V.K., 2016. Miscellaneous—Icons and ARIA. In Material Design implementation with AngularJS (pp. 149-157). Apress, Berkeley, CA.
- [10] Kumari, P. and Nandal, R., 2017. A Research Paper On Website Development Optimization Using Xampp/PHP. International Journal of Advanced Research in Computer Science, 8(5).
- [11] Lerdorf, R., Tatroe, K., Kaehms, B. and McGredy, R., 2002. Programming Php. " O'Reilly Media, Inc. ".
- [12] Loske, M., Rothe, L. and Gertler, D.G., 2019, April. Context-aware authentication: State-of-the-art evaluation and adaption to the IIoT. In 2019 IEEE 5th World Forum on Internet of Things (WF-IoT) (pp. 64-69). IEEE.
- [13] Lou, D., 2019. Two fast prototypes of web-based augmented reality enhancement for books. Library Hi Tech News.
- [14] Mehta, C., Bhavsar, A.K., Oza, H. and Shah, S., 2018. MySQL 8 administrator's guide: effective guide to administering high-performance MySQL 8 solutions. Packt Publishing Ltd.
- [15] Musciano, C. and Kennedy, B., 2002. HTML & XHTML: The Definitive Guide: The Definitive Guide. " O'Reilly Media, Inc. ".
- [16] Ngee, S., 2010. Data Transmission Between PDA and PC Using WiFi for Pocket Barcode Application (Doctoral dissertation, Thesis, University Teknologi Malaysia, May 2007, 126 pp. Found online at <http://eprints.utm.my/6421/1/SeahYeowNgeeMFKE20007TTT.pdf>).
- [17] Nielsen, J. and Nielsen, B., 1995. Multimedia and hypertext: The Internet and beyond. Morgan Kaufmann.
- [18] Nishitha, R., Naik, S.S., Raksha, V. and Kulsum, U., 2019, May. IoT based automatic billing system using barcode scanner by android device and monitoring unregistered barcode by RFID. In 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT) (pp. 15-20). IEEE.
- [19] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D. and Walsh, J., 2019, April. Deep learning vs. traditional computer vision. In Science and information conference (pp. 128-144). Springer, Cham.
- [20] Othuo, K.D., 2018. Improving customer shopping experience using an Android bar-code reader application (Doctoral dissertation, Strathmore University).
- [21] Piantadosi, V., Scalabrino, S. and Oliveto, R., 2019, April. Fixing of security vulnerabilities in open source projects: A case study of apache http server and apache tomcat. In 2019 12th IEEE Conference on software testing, validation and verification (ICST) (pp. 68-78). IEEE.
- [22] Stauffer, M., 2019. Laravel: Up & running: A framework for building modern php apps. O'Reilly Media.
- [23] Trinh, H.N., Tran, H.H. and Vuong, D.H.Q., 2020. Determinants of consumers' intention to use credit card: a perspective of multifaceted perceived risk. Asian Journal of Economics and Banking.
- [24] Valentine, T., 2021. Installing and Using the Perl Server. In Database-Driven Web Development (pp. 131-137). Apress, Berkeley, CA.
- [25] Wirfs-Brock, A. and Eich, B., 2020. JavaScript: the first 20 years. Proceedings of the ACM on Programming Languages, 4(HOPL), pp.1-189.
- [26] Withana, P.W.J.C., Uwanthika, G.A.I. and Wijethunge, I.A., 2021. A Novel Personalized Mobile Application for Systematically Monitoring Cash Transactions.
- [27] Wood, W., 2019. MariaDB solution. In Migrating to MariaDB (pp. 59-71). Apress, Berkeley, CA.
- [28] Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A.M. and Wang, X., 2021. Computer vision techniques in construction: a critical review. Archives of Computational Methods in Engineering, 28(5), pp.3383-3397.
- [29] Xun, Y., Liu, J., Kato, N., Fang, Y. and Zhang, Y., 2019. Automobile driver fingerprinting: A new machine learning based authentication scheme. IEEE Transactions on Industrial Informatics, 16(2), pp.1417-1426.
- [30] Zlatilov, Z., 2018. Design and development of a responsive website to enhance business potential—A case project for Penelope Art.